J Sci Comput
DOI 10.1007/s10915-016-0351-1

Boundary Treatment and Multigrid Preconditioning for Semi-Lagrangian Schemes Applied to Hamilton–Jacobi–Bellman Equations

Christoph Reisinger[1] · **Julen Rotaetxe Arto**[1]

Received: 17 May 2016 / Revised: 6 November 2016 / Accepted: 26 December 2016

Abstract We analyse two practical aspects that arise in the numerical solution of Hamilton–Jacobi–Bellman equations by a particular class of monotone approximation schemes known as semi-Lagrangian schemes. These schemes make use of a wide stencil to achieve convergence and result in discretization matrices that are less sparse and less local than those coming from standard finite difference schemes. This leads to computational difficulties not encountered there. In particular, we consider the overstepping of the domain boundary and analyse the accuracy and stability of stencil truncation. This truncation imposes a stricter CFL condition for explicit schemes in the vicinity of boundaries than in the interior, such that implicit schemes become attractive. We then study the use of geometric, algebraic and aggregation-based multigrid preconditioners to solve the resulting discretised systems from implicit time stepping schemes efficiently. Finally, we illustrate the performance of these techniques numerically for benchmark test cases from the literature.

Keywords Fully non-linear PDEs · Monotone approximation schemes · Wide stencils · Semi-Lagrangian schemes · Multigrid

Mathematics Subject Classification 65M06 · 65M12 · 65M55 · 49L25 · 93E20

1 Introduction

We consider semi-Lagrangian schemes, as described in [5,9], for the numerical approximation of solutions to the Hamilton–Jacobi–Bellman (HJB) equation

✉ Julen Rotaetxe Arto
rotaetxe@maths.ox.ac.uk

Christoph Reisinger
reisinge@maths.ox.ac.uk

[1] Mathematical Institute, University of Oxford, Oxford OX2 6GG, UK

Published online: 13 January 2017

$$u_t - \inf_{\alpha \in \mathcal{A}} \left\{ L^\alpha[u](t,x) + c^\alpha(t,x)u(t,x) + f^\alpha(t,x) \right\} = 0, \quad (t,x) \in (0,T] \times \Omega, \tag{1.1}$$

$$u(0,x) = g(x), \quad x \in \bar{\Omega}, \tag{1.2}$$

$$u(t,x) = \psi(x), \quad (t,x) \in (0,T] \times \partial\Omega, \tag{1.3}$$

where Ω is a domain, $Q_T := (0,T] \times \bar{\Omega}$ with $\bar{\Omega} := \Omega \cup \partial\Omega \subseteq \mathbb{R}^d$, $\mathcal{A}$ is a compact set,

$$L^\alpha[u](t,x) = \mathrm{tr}[a^\alpha(t,x)D^2u(t,x)] + b^\alpha(t,x)Du(t,x) \tag{1.4}$$

is a second order differential operator, and ψ and g are the Dirichlet and initial conditions.

The coefficients $a^\alpha = \frac{1}{2}\sigma^\alpha\sigma^{\alpha,T}$, b^α, c^α, f^α, the initial data g and the boundary conditions ψ take their values, respectively, in $\mathbb{S}^d$, the space of $d \times d$ symmetric matrices, $\mathbb{R}^d$, $\mathbb{R}$, $\mathbb{R}$, $\mathbb{R}$, and $\mathbb{R}$, and $\sigma^\alpha \in \mathbb{R}^{d \times P}$ such that a^α is positive semi-definite. We also assume the usual well-posedness conditions on the PDE coefficients, i.e. Lipschitz continuous in x uniformly in α, Hölder continuous with exponent $\frac{1}{2}$ in time and continuous in α for each $(t,x) \in Q_T$ [18]. The relevant notion of solution for this type of non-linear equations is that of viscosity solutions [7] and the above conditions guarantee existence and uniqueness.

In general, the viscosity solution to (1.1)–(1.3) is unknown, thus it is necessary in practice to compute approximations numerically. Sufficient conditions for a numerical scheme to converge to the unique viscosity solution of (1.1)–(1.3) were proved by Barles and Souganidis [2] in terms of consistency, L^∞-stability and monotonicity. We restrict our attention to finite difference discretizations of the differential operator (1.4).

The requirement of monotonicity drastically affects the properties and construction of finite difference schemes. Theorem 4 in [27] proves that local monotone discretizations have at most first order for first-order equations and second order for second-order equations. What is more, standard fixed stencil methods are monotone only under restrictions on the diffusion matrix, such as diagonal dominance [9,12]. Results from [6,21] further illustrate the limitations of such methods for the monotone approximation of second order derivatives.

This implies that generally approximations have to be non-local on the discrete level, i.e. the distance between mesh points involved in the scheme at a given point grows in relation to the mesh width as the mesh is refined. Such schemes are referred to as wide stencils. For general diffusion matrices, first order accurate wide stencils of the type considered here have been proposed in [5,9], and a mixed fixed- and wide-stencil scheme in [19].

In this article, we analyse two issues arising in practice when numerically solving (1.1)–(1.3) using the class of schemes described in [5,9,20] to discretize the second order differential operator (1.4). This approximation combines wide stencils in the directions determined by the columns of the diffusion matrix σ^α and the drift b^α, together with (linear) interpolation. Following the notation in [9], we write the matrix $\sigma^\alpha \in \mathbb{R}^{d\times P}$ as $(\sigma_1^\alpha, \sigma_2^\alpha, \ldots, \sigma_P^\alpha)$, where $\sigma_p^\alpha \in \mathbb{R}^d$ for $p \in \{1, 2, \ldots, P\}$ denotes the p-th column of σ^α, and observe that for $k > 0$ and any smooth function ϕ,

$$\frac{1}{2}\mathrm{tr}\left[\sigma^\alpha\sigma^{\alpha\,T} D^2\phi(x)\right] = \frac{1}{2}\sum_{p=1}^{P} \frac{\phi(x + k\sigma_p^\alpha) - 2\phi(x) + \phi(x - k\sigma_p^\alpha)}{k^2} + \mathcal{O}(k^2), \tag{1.5}$$

$$b^\alpha D\phi(x) = \frac{\phi(x + k^2 b^\alpha) - \phi(x)}{k^2} + \mathcal{O}(k^2), \tag{1.6}$$

where $\mathcal{O}(k^2)$ is the local truncation error of the finite difference and for compactness we write $b^\alpha \equiv b^\alpha(t,x)$ and $\sigma^\alpha \equiv \sigma^\alpha(t,x)$. As these approximations will be used for points lying on a discrete spatial grid $\Omega_{\Delta x}$ with nodes $\{x_j : 1 \leq j \leq N\}$, the displaced points $x + k^2 b^\alpha$, $x \pm k\sigma_p^\alpha$ do not generally coincide with nodes of $\Omega_{\Delta x}$. Therefore, ϕ is replaced by an interpolant $\mathcal{I}_{\Delta x}\phi$ on that grid. We restrict our attention to linear interpolants, defined by the standard piecewise multilinear non-negative basis functions $\{w_j(\cdot) : 1 \leq j \leq N\}$ associated with the mesh nodes, such that for any function ϕ

$$(\mathcal{I}_{\Delta x}\phi)(x) = \sum_{j \in \mathcal{N}(x)} \phi(x_j) w_j(x), \tag{1.7}$$

for all $x \in \Omega$, $x_j \in \Omega_{\Delta x}$, where $\mathcal{N}(x)$ is the set of neighbours of x on the mesh $\Omega_{\Delta x}$, i.e. the mesh points with non-zero interpolation weight. The resulting scheme is referred to as the Linear Interpolation Semi-Lagrangian (LISL) scheme.

It is shown in [9] that the leading order terms of the local truncation error are proportional to k^2 and $\frac{\Delta x^2}{k^2}$, where the last quantity corresponds to the linear interpolation error in the finite difference formulae (1.5) and (1.6) by replacing ϕ by its interpolant. Therefore, by choosing $k = \sqrt{\Delta x}$, the resulting scheme is locally of first order in Δx.

Following the notation in [9], the LISL finite difference approximations for the differential operator in (1.4) can be expressed as

$$\begin{aligned} &L^{\alpha}_{\Delta x}[\mathcal{I}_{\Delta x}\phi](t,x) \\ &:= \sum_{p=1}^{M} \frac{(\mathcal{I}_{\Delta x}\phi)(t, x + y_p^{\alpha,+}(t,x)) - 2(\mathcal{I}_{\Delta x}\phi)(t,x) + (\mathcal{I}_{\Delta x}\phi)(t, x + y_p^{\alpha,-}(t,x))}{2\Delta x}, \end{aligned} \tag{1.8}$$

for $x \in \Omega_{\Delta x}$, and some $M \geq 1$.

Different schemes can be obtained depending on the values taken by M and $y_p^{\alpha,\pm}(t,x)$. In particular, [9] discusses the following three schemes:

Examples of LISL schemes.

1. **Scheme 1**: The approximation of Camilli and Falcone [5], corresponding to $y_p^{\alpha,\pm} = \pm\sqrt{\Delta x}\sigma_p^{\alpha} + \frac{\Delta x}{P} b^{\alpha}$ and $M = P$.
2. **Scheme 2**: The approximation in [9], corresponding to $y_p^{\alpha,\pm} = \pm\sqrt{\Delta x}\sigma_p^{\alpha}$ for $p \leq P$, $y_{P+1}^{\alpha,\pm} = \Delta x b^{\alpha}$, and $M = P + 1$.
3. **Scheme 3**: A more efficient version of the Camilli–Falcone approximation, corresponding to $y_p^{\alpha,\pm} = \pm\sqrt{\Delta x}\sigma_p^{\alpha}$ for $p < P$, $y_P^{\alpha,\pm} = \pm\sqrt{\Delta x}\sigma_P^{\alpha} + \Delta x b^{\alpha}$, and $M = P$.

The authors show that this family of discretizations of (1.4) is consistent and monotone. Monotonicity of the scheme is fulfilled as the discrete approximation $L^{\alpha}_{\Delta x}[\mathcal{I}_{\Delta x}\phi]$ is the composition of monotone finite differences and a monotone interpolation operation. Once discretized in space, the final scheme arises from discretising in time using the standard θ-time stepping scheme for $\theta \in [0, 1]$, where $\theta = 0$ corresponds to the explicit Euler time stepping and $\theta = 1$ to the implicit case, on a time grid represented by a strictly increasing sequence of points $\{t_n\}_{n=0}^{N_t+1}$ with $t_0 = 0$, $t_{N_t+1} = T$, and $\Delta t_n := t_n - t_{n-1} \leq \Delta t$ for all n. The scheme being monotone, it can be written as described in the following definition, where for any grid function $V : \{t_n\}_{n=0}^{N_t+1} \times \Omega_{\Delta x} \to \mathbb{R}$, $V_i^n \equiv V(t_n, x_i)$.

Definition 1.1 (*Equation* (4.1) *in* [9]) A scheme is said to be of positive type, if it can be written as

$$\max_{\alpha \in \mathcal{A}} \left\{ \mathcal{B}_{j,j}^{\alpha,n,n} U_j^n - \sum_{i \neq j} \mathcal{B}_{j,i}^{\alpha,n,n} U_i^n - \sum_{i=1}^{N} \mathcal{B}_{j,i}^{\alpha,n,n-1} U_i^{n-1} - F_j^{\alpha,n-1+\theta} \right\} = 0, \tag{1.9}$$

for $j = 1, \ldots, N$, on the discrete domain $\{t_n\}_{n=0}^{N_t+1} \times \Omega_{\Delta x}$, where U_i^n is the numerical solution at node (t_n, x_i) and all the coefficients $\mathcal{B}$ are non-negative.

For the convenience of the reader, we reproduce the expressions for $\mathcal{B}_{j,\cdot}^{\alpha,n,\cdot}$ of the LISL schemes as in [9], for all $1 \le i \ne j \le N$, $x_i, x_j \notin \partial\Omega$,

$$\mathcal{B}_{j,j}^{\alpha,n,n} = 1 + \theta\Delta t_n \left(\frac{M}{2\Delta x} - l_{j,j}^{\alpha,n} - c_j^{\alpha,n-1+\theta}\right), \qquad \mathcal{B}_{j,i}^{\alpha,n,n} = \theta\Delta t_n\, l_{j,i}^{\alpha,n},$$

$$\mathcal{B}_{j,j}^{\alpha,n,n-1} = 1 - (1-\theta)\Delta t_n \left(\frac{M}{2\Delta x} - l_{j,j}^{\alpha,n-1} - c_j^{\alpha,n-1+\theta}\right), \quad \mathcal{B}_{j,i}^{\alpha,n,n-1} = (1-\theta)\Delta t_n\, l_{j,i}^{\alpha,n-1},$$

where $c_j^{\alpha,n-1+\theta} = c^\alpha(t_{n-1} + \theta\Delta t, x_j)$ and

$$l_{j,i}^{\alpha,n} = \sum_{p=1}^{M} \frac{w_i(x_j + y_p^{\alpha,+}(t_n, x_j)) + w_i(x_j + y_p^{\alpha,-}(t_n, x_j))}{2\Delta x}.$$

The schemes described above have a wide stencil as the length of the stencil, being proportional to the ratio $k/\Delta x \sim 1/\sqrt{\Delta x}$, tends to ∞ as $\Delta x \to 0$. Hence, when applied on a bounded discrete grid, the stencil will generally exceed the domain for points close to its boundary. As discussed in [9], the overstepping may pose a problem depending on the equation and the type of boundary conditions imposed. We consider Dirichlet boundary conditions here.

Our first goal is to present and analyse a modification of the LISL scheme to deal with overstepping for problems on bounded domains with Dirichlet boundary conditions, and general drift and diffusion coefficients. We describe how to truncate the LISL stencil so that the truncation remains consistent and monotone. We prove that the resulting stencil for Scheme 2 above is of positive type (as per Definition 1.1), and since the coefficients $\mathcal{B}$ in (1.9) do not depend on U, it is also monotone. This is not the case for Schemes 1 and 3. We also observe that the truncation has both local and global impacts on the properties of the scheme. Locally, the modification of the scheme leads to a loss of accuracy of half an order in the consistency error, i.e. $\mathcal{O}(\sqrt{\Delta x})$ instead of $\mathcal{O}(\Delta x)$, due to the loss of symmetry. We compare the accuracy of the truncation with extrapolations of the boundary conditions by way of numerical tests for benchmark problems. As the mesh points requiring truncation of the scheme are restricted to an $\mathcal{O}(\sqrt{\Delta x})$ layer at the boundary, convergence rates close to $\mathcal{O}(\Delta x)$ are observed empirically for the new scheme. The truncation has a global effect in the sense that it modifies the CFL condition of explicit schemes by at least half an order, from $\Delta t = \mathcal{O}(\Delta x)$ to $\Delta t = \mathcal{O}(\Delta x^{3/2})$. As the empirical error is $\mathcal{O}(\Delta t) + \mathcal{O}(\Delta x)$ for fully implicit schemes, the computationally most efficient choice is $\Delta t \sim \Delta x$, outside the stability region of explicit schemes.

The second goal is therefore the use of implicit schemes and the efficient solution of the discrete system (1.9) using multigrid preconditioning. For $\theta \ne 0$, the coupling of the optimal control and the coefficients makes (1.9) a non-linear system of algebraic equations,

$$\max_{\alpha\in\mathcal{A}} \left(A_i^\alpha X - F_i^\alpha\right) = 0, \qquad i = 1, \ldots, N, \tag{1.10}$$

where A_i^α is the i-th row of a matrix A^α with elements $A_{i,j}^\alpha$, $i, j = 1, \ldots, N$, and control $\alpha \in \mathcal{A}$. Comparing with (1.9), $A_{i,j}^\alpha = \mathcal{B}_{i,j}^{\alpha,n,n}$, $F_i^\alpha = F_i^{\alpha,n-1+\theta}$, and $X = (X_i) = (U_i^n)$ is the solution vector for the n-th time step. The maximisation over α in (1.10) is row-wise and usually done by linear search. By construction of the LISL scheme, A^α is an M-matrix with non-negative row sum. Therefore, following results in [4], we can use policy iteration to compute U. Then, within each policy iteration, a linear system $A_i^{\alpha_i} X = F_i^{\alpha_i}$, $i = 1, \ldots, N$, with fixed control vector $(\alpha_i)_{1 \leq i \leq N}$ has to be solved. We find (in contrast to [19]) that this last step is the computationally most costly part of the overall algorithm if direct linear solvers or standard iterative solvers are used.[1] We therefore study multigrid preconditioners (see Table 16.)

In the literature on multigrid for HJB equations, two main approaches are observed: on the one hand, multigrid is applied directly to the non-linear problem, as in [3,13,14]; and on the other hand, multigrid is applied to a linearised problem, as in [1]. In particular, [3,14] provide the first multigrid algorithms for HJB equations and prove convergence, while [13] presents a novel smoother for HJB equations based on damped value iteration [17]. These articles have in common the use of standard fixed stencil finite difference approximations and the use of a geometric structure when building the hierarchy of multigrid subspaces.

The novelty of this article is to study the application of multigrid preconditioning to a wide stencil discretization. We will demonstrate, both by Fourier analysis of a model problem and by numerical tests in a more complex application, that standard geometric multigrid does not give mesh-size independent convergence.

We then investigate algebraic multigrid methods. The basis for the specific algorithm we use was introduced in [24] for linear elliptic PDEs. It empirically showed that "aggregation based methods could yield robust[2] and convergent schemes if used as preconditioners of a Krylov method, and were part of an enhanced multigrid cycle, not simple V- or W-cycles" as considered in [31]. By enhanced multigrid cycles, the authors refer to recursive schemes in which at each coarse level the solution to the residual equation is computed using a number of Krylov subspace iterations as in [26] or with a semi-iterative method based on Chebyshev polynomials called the AMLI cycle, see Section 5.6 of [34]. The aggregates were formed using heuristic criteria following coupling in the strongest direction.

In [22] the authors introduced an aggregation-based multigrid method with guaranteed convergence rate for symmetric M-matrices with non-negative row sum. A LISL discretization matrix is only symmetric in very specific cases with limited practical interest. For non-symmetric matrices, in [25] convergence of a simplified two-grid scheme using aggregation is proved for non-singular M-matrices with non-negative row and column sums. This requirement ensures that the symmetric part of the coefficient matrix A given by $A + A^T$ meets the assumptions in [22] and allows the use of its theoretically justified algorithms. We will derive conditions on the coefficients of the HJB equation such that this theory applies, and show empirically that aggregation-based multigrid gives roughly mesh-size independent convergence.

The rest of the article is organised as follows. Section 2 discusses the truncation of the LISL scheme for points whose stencil exceeds the domain and compares its performance to naïve extrapolations of the boundary conditions. Section 3 considers the application of three different multigrid methods to linear systems where the coefficient matrix arises from LISL discretizations. Section 4 contains the final remarks.

[1] Which of the two steps is more costly depends crucially on the type of control problem. The optimisation step is typically fast if the control is taken from a finite set, if the local control problem is analytically solvable (e.g., quadratic or of 'bang-bang'-type), or if the coefficients are a smooth convex function of the control, such that standard Newton-type methods can be used. It will be more costly, if the optimal control has to be approximated by exhaustive search over a discretised control set, especially if the dimension of the control space is higher than the spatial dimension of the PDE. In the examples considered in this paper, the control is scalar and we optimise by linear search over a one-dimensional control mesh.

[2] In this context, a robust method is referred to as one showing good performance for a large range of problems without changing the smoother.

2 Boundary Treatment for the LISL Scheme

In this section, we analyse adaptations of Schemes 1–3 for initial-boundary value problems on bounded domains. As described in the introduction, for points x close to the boundaries of the domain, the stencil points $x+y_p^{\alpha,\pm}(t,x)$ in (1.8) generally do not lie in a mesh element. In the following, we therefore discuss the truncation of (1.8) so that the resulting scheme remains monotone, consistent, and L^∞-stable. The proposed truncation samples the boundary points on the straight lines defined by the point x and $x+y_p^{\alpha,\pm}(t,x)$ and adjusts the corresponding finite difference weights for consistency.

2.1 Definition of Truncated Stencils

We take $\Omega \subset \mathbb{R}^d$ for $d \geq 2$. We first outline how the method can be defined on a general domain with curved boundary, but later (especially in the numerical tests) focus for simplicity on rectangular domains. We start with a Cartesian mesh on $\mathbb{R}^d$ with uniform mesh width Δx and then choose $\Omega_{\Delta x}$ as all the points which lie inside Ω. See Fig. 1.

We now fix a mesh node $x \in \Omega_{\Delta x}$. There are two distinct situations where interpolation at the point $x+y_p^{\alpha,\pm}(t,x)$ as per (1.8) is not possible for given t, α and p:

A. $x+y_p^{\alpha,\pm}(t,x) \notin \bar{\Omega}$ (bottom left in Fig. 1);
B. $x+y_p^{\alpha,\pm}(t,x) \in \bar{\Omega}$, but the element it is contained in has vertices outside $\bar{\Omega}$ (top right).

We say the stencil "oversteps". In such cases, the objective is to find truncated or extended stencil vectors $\hat{y}_p^{\alpha,\pm}(t,x)$ and corresponding finite difference weights $A_p^\alpha \equiv A_p^\alpha(t,x)$ and $B_p^\alpha \equiv B_p^\alpha(t,x)$, such that $x+\hat{y}_p^{\alpha,\pm}(t,x) \in \partial\Omega$ and the truncated scheme

$$\hat{L}^\alpha_{\Delta x}[\mathcal{I}_{\Delta x}\phi](t,x) := \sum_{p=1}^{M} \frac{A_p^\alpha(\mathcal{I}_{\Delta x}\phi)(t,x+\hat{y}_p^{\alpha,+}(t,x)) - (A_p^\alpha + B_p^\alpha)\phi(t,x) + B_p^\alpha(\mathcal{I}_{\Delta x}\phi)(t,x+\hat{y}_p^{\alpha,-}(t,x))}{2\Delta x} \tag{2.1}$$

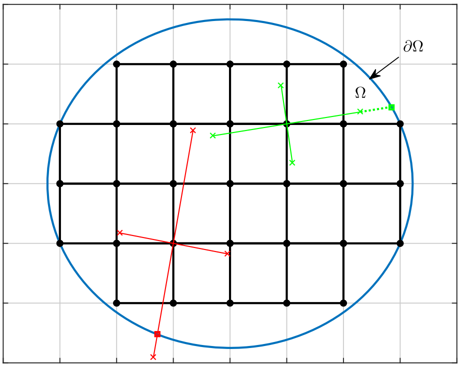

Fig. 1 Truncation and extrapolation of the stencil for an elliptical domain and a mesh made of *square cells*. The modified stencil samples the domain boundary

is a consistent approximation of (1.4) as $\Delta x \to 0$. If the stencil does not overstep, we have that $\hat{y}_p^{\alpha,\pm}(t,x) = y_p^{\alpha,\pm}(t,x)$ and $A_p^\alpha = B_p^\alpha = 1$. If it does, for any t we define

$$\hat{y}_p^{\alpha,\pm}(t,x) = \mu_p^{\alpha,\pm}(t,x) y_p^{\alpha,\pm}(t,x), \quad \text{where}$$
$$\mu_p^{\alpha,\pm}(t,x) = \min\left\{\mu \geq 0 \, : \, x + \mu y_p^{\alpha,\pm}(t,x) \in \partial\Omega\right\}.$$

In case A, this means $\mu < 1$, while in case B we have $\mu > 1$.

In the remainder of this section we restrict our attention to the truncation of the scheme on rectangular domains, in which case the elements of the Cartesian mesh cover exactly the domain and case B does not occur. Moreover, this means that interior mesh points cannot be arbitrarily close to the boundary, but are always at least Δx away.[3] This allows the derivation of CFL conditions for the explicit schemes as given below in Sect. 2.3.

2.2 Consistency Conditions

In the truncated scheme (2.1) there are M pairs of weights, which can be chosen freely, subject to positivity, in order to obtain a consistent scheme. As we will see below, this is only possible for Scheme 2.

In the following, we denote $[[1, j]] \equiv [1, j] \cap \mathbb{Z}$ and for a vector $v \in \mathbb{R}^d$, $(v)_i$ denotes its i-th element. As in the introduction, we have that $b^\alpha \in \mathbb{R}^d$, and $\sigma^\alpha = (\sigma_1^\alpha, \ldots, \sigma_p^\alpha, \ldots, \sigma_P^\alpha) \in \mathbb{R}^{d\times P}$ where $\sigma_p^\alpha \in \mathbb{R}^d$ denotes the p-th column vector. For compactness, we omit the dependence of the coefficients and the stencil related functions with respect to the position, that is $b^\alpha \equiv b^\alpha(t,x)$, $\sigma_p^\alpha \equiv \sigma_p^\alpha(t,x)$, $y_p^{\alpha,\pm} \equiv y_p^{\alpha,\pm}(t,x)$ and $\mu_p^{\alpha,\pm} \equiv \mu_p^{\alpha,\pm}(t,x)$. We add a second subscript taking values 1, 2 or 3 to A_p^α, B_p^α and $y_p^{\alpha,\pm}$ to make the discretization scheme explicit.

Proposition 2.1 *The truncated version of Schemes 1 and 3 is generally not consistent.*

Proof By Taylor expansion of a smooth test function we find that the consistency conditions for Scheme 1 are

$$\sum_{p\in\mathcal{P}} \left(A_{1,p}^\alpha (\hat{y}_{1,p}^{\alpha,+})_i + B_{1,p}^\alpha (\hat{y}_{1,p}^{\alpha,-})_i\right) = 2\Delta x \frac{|\mathcal{P}|}{P}(b^\alpha)_i + o(\Delta x),$$

$$\sum_{p\in\mathcal{P}} \left(A_{1,p}^\alpha (\hat{y}_{1,p}^{\alpha,+})_{i_1} (\hat{y}_{1,p}^{\alpha,+})_{i_2} + B_{1,p}^\alpha (\hat{y}_{1,p}^{\alpha,-})_{i_1} (\hat{y}_{1,p}^{\alpha,-})_{i_2}\right) = 2\Delta x \sum_{p\in\mathcal{P}} (\sigma_p^\alpha)_{i_1} (\sigma_p^\alpha)_{i_2} + o(\Delta x),$$

where $\mathcal{P} \subseteq [[1, P]]$ denotes the set of stencils overstepping the domain and $i, i_1, i_2 \in [[1, d]]$.

In Scheme 1, there are $2|\mathcal{P}| \leq 2d$ variables, but $(d^2+3d)/2$ equations, d from the condition on the Jacobian and $(d^2+d)/2$ from the condition on the Hessian. This overdetermined system has a solution only if there is linear dependence between the equations. Except for special cases, e.g. $|\mathcal{P}| = 0$ or σ_p^α parallel to b^α for some p, this is not the case. Hence, in general the truncated Scheme 1 is not consistent.

We observe that the same principle applies to Scheme 3 for $y_{3,P}^{\alpha,\pm} = \pm\sqrt{\Delta x}\sigma_P^\alpha + \Delta x b^\alpha$. □

For example, consider $x_0 = (0,0)^T$, $\bar{\Omega} = [-5, 1]^2$, $\sqrt{\Delta x}\sigma_1^\alpha(x_0) = (2,0)^T$, $\sqrt{\Delta x}\sigma_2^\alpha(x_0) = (0,1)^T$, and $\Delta x b^\alpha(x_0) = (0,1)^T$, then the truncated version of Scheme

[3] This can also be enforced in the general case by removing the outermost layer of cells, such that again a distance of Δx between non-boundary mesh points and the domain boundary is ensured.

1 is not consistent, but the one for Scheme 3 is. However, if $\Delta x b^\alpha(x_0) = (1,1)^T$ then neither of them is consistent.

We conclude that for points whose stencil oversteps the boundary, the approximations of the first and second derivative should be considered separately, as done in Scheme 2.

Proposition 2.2 *For Scheme 2 and all $p \in [[1, P+1]]$, let $\mu_p^{\alpha,\pm} \in (0,1]$ be the largest constant such that $x + \mu\, y_{2,p}^{\alpha,\pm} \in \bar{\Omega}$ for all $\mu \in [0, \mu_p^{\alpha,\pm}]$, and define*

$$A^\alpha_{2,P+1} = B^\alpha_{2,P+1} = \frac{1}{\mu^{\alpha,+}_{P+1}} \left(= \frac{1}{\mu^{\alpha,-}_{P+1}} \right), \tag{2.2}$$

and, for $p \in [[1, P]]$,

$$A^\alpha_{2,p} = \frac{2}{(\mu_p^{\alpha,+})^2 + \mu_p^{\alpha,+}\mu_p^{\alpha,-}}, \qquad B^\alpha_{2,p} = \frac{2}{(\mu_p^{\alpha,-})^2 + \mu_p^{\alpha,-}\mu_p^{\alpha,+}}. \tag{2.3}$$

Then the scheme defined by (2.1) is consistent unless both $\mu_p^{\alpha,+}, \mu_p^{\alpha,-} \sim \mathcal{O}(\sqrt{\Delta x})$.

Proof If the stencil oversteps, then the truncated stencil consists of the point at the intersection between the boundary $\partial\Omega$ and one of the segments $\{x, x + \sqrt{\Delta x}\sigma_p^\alpha\}$, $\{x, x - \sqrt{\Delta x}\sigma_p^\alpha\}$, or $\{x, x + \Delta x b^\alpha\}$. For each point (t, x) Scheme 2 requires the calculation of at most $2P+1$ different weights, i.e. $2P$ for the second order term and one for the first order term. For the latter we have that $\hat{y}^{\alpha,+}_{2,P+1} = \hat{y}^{\alpha,-}_{2,P+1}$, therefore $A^\alpha_{2,P+1} = B^\alpha_{2,P+1}$. Ignoring the interpolation error for the time being, the coefficients are obtained from the consistency conditions (up to a term $o(\Delta x)$),

$$(A^\alpha_{2,P+1} + B^\alpha_{2,P+1})(\hat{y}^{\alpha,\pm}_{2,P+1})_i = 2\Delta x (b^\alpha)_i, \quad \forall i \in [[1,d]], \tag{2.4}$$

for the first order term, and

$$A^\alpha_{2,p}(\hat{y}^{\alpha,+}_{2,p})_i + B^\alpha_{2,p}(\hat{y}^{\alpha,-}_{2,p})_i = 0, \quad \forall i \in [[1,d]], \tag{2.5}$$

$$A^\alpha_{2,p}(\hat{y}^{\alpha,+}_{2,p})_{i_1}(\hat{y}^{\alpha,+}_{2,p})_{i_2} + B^\alpha_{2,p}(\hat{y}^{\alpha,-}_{2,p})_{i_1}(\hat{y}^{\alpha,-}_{2,p})_{i_2} = 2\Delta x (\sigma_p^\alpha)_{i_1}(\sigma_p^\alpha)_{i_2}, \quad \forall (i_1, i_2) \in [[1,d]]^2, \tag{2.6}$$

for the second order term.

By construction of the truncated stencil (2.4) and (2.5) are linearly dependent across i, and (2.6) across i_1 and i_2, resulting in one (linearly independent) equation for the first order term weights and two for $A^\alpha_{2,p}$, $B^\alpha_{2,p}$, with solutions given by

$$A^\alpha_{2,P+1} = B^\alpha_{2,P+1} = \Delta x \frac{(b^\alpha)_i}{(\hat{y}^{\alpha,\pm}_{2,P+1})_i}, \tag{2.7}$$

and

$$A^\alpha_{2,p} = \frac{2\Delta x (\sigma_p^\alpha)_i^2}{(\hat{y}^{\alpha,+}_{2,p})_i((\hat{y}^{\alpha,+}_{2,p})_i - (\hat{y}^{\alpha,-}_{2,p})_i)}, \qquad B^\alpha_{2,p} = \frac{2\Delta x (\sigma_p^\alpha)_i^2}{(\hat{y}^{\alpha,-}_{2,p})_i((\hat{y}^{\alpha,-}_{2,p})_i - (\hat{y}^{\alpha,+}_{2,p})_i)}, \tag{2.8}$$

which are seen to be equivalent to Eqs. (2.2) and (2.3).

The contribution to the consistency error of (2.1) from the bilinear interpolation operator $\mathcal{I}$ is bounded by $(\Delta x)^{-1} \sum_p (|A_p| + |B_p|)(\Delta x)^2$, which is goes to 0 if $|A_p| + |B_p| = o((\Delta x)^{-1})$ for all p, which is violated if and only if $\mu_p^{\alpha,+}, \mu_p^{\alpha,-} \sim \mathcal{O}(\sqrt{\Delta x})$. □

Corollary 2.3 *For the truncated Scheme 2,* (2.1), (2.2) *and* (2.3)*, the following holds:*

(a) *The scheme is of positive type and monotone with $A^\alpha_{2,p}, B^\alpha_{2,p} \geq 1$ for all $p \in [[1, P+1]]$.*
(b) *For points x within a distance $\mathcal{O}(\Delta x)$ of the boundary and $p \neq P+1$, as $\Delta x \to 0$,*

$$\text{if } |\hat{y}^{\alpha,+}_{2,p}| < \sqrt{\Delta x}|\sigma^\alpha_p| \text{ and } |\hat{y}^{\alpha,-}_{2,p}| = \sqrt{\Delta x}|\sigma^\alpha_p| \implies A^\alpha_{2,p} \sim \mathcal{O}(\Delta x^{-1/2}) \text{ and } \lim_{\Delta x \to 0} B^\alpha_{2,p} = 2,$$
$$\text{if } |\hat{y}^{\alpha,-}_{2,p}| < \sqrt{\Delta x}|\sigma^\alpha_p| \text{ and } |\hat{y}^{\alpha,+}_{2,p}| = \sqrt{\Delta x}|\sigma^\alpha_p| \implies \lim_{\Delta x \to 0} A^\alpha_{2,p} = 2 \text{ and } B^\alpha_{2,p} \sim \mathcal{O}(\Delta x^{-1/2}),$$
$$\text{if } |\hat{y}^{\alpha,\pm}_{2,p}| < \sqrt{\Delta x}|\sigma^\alpha_p| \implies A^\alpha_{2,p}, B^\alpha_{2,p} \sim \mathcal{O}(\Delta x^{-1}).$$

(c) *The local consistency error for points with truncation and $p \neq P+1$ is $\mathcal{O}(\sqrt{\Delta x})$ if only one side of the stencil oversteps, and $\mathcal{O}(1)$ if both sides overstep.*

Proof The claim in (a) follows from (2.2), (2.3), and the fact that $\mu^{\alpha,\pm}_p \in (0,1]$ and the coefficients $A^\alpha_{2,p}, B^\alpha_{2,p}$ do not depend on the numerical solution U. The limits in (b) follow from (2.3) and noting that if the stencil oversteps for a point x lying $\mathcal{O}(\Delta x)$ away from the boundary, but at least Δx by the assumption made on the mesh, then $\mu^{\alpha,+}_p \sim \mathcal{O}(\sqrt{\Delta x})$ and/or $\mu^{\alpha,-}_p \sim \mathcal{O}(\sqrt{\Delta x})$, but not $o(\sqrt{\Delta x})$.

To prove (c) we use Taylor expansions for each p and conclude using the limits in *b)*. Let $\phi : \bar{\Omega} \to \mathbb{R}$ be a smooth function and for any $p \in (\mathcal{P} \cap [[1, P]])$, where $\mathcal{P}$ denotes the set of stencils overstepping the domain, then by Taylor expansion and the consistency conditions (2.5)–(2.6) the local consistency error τ for the p-th addend of (2.1) using multi-index notation is given by

$$\tau := \frac{A^\alpha_p \phi(t, x + \hat{y}^{\alpha,+}_p) - (A^\alpha_p + B^\alpha_p)\phi(t,x) + B^\alpha_p \phi(t, x + \hat{y}^{\alpha,-}_p)}{2\Delta x} - \frac{1}{2}\mathrm{tr}[\sigma^\alpha_p \sigma^{\alpha,T}_p D^2\phi]$$
$$= \frac{1}{2\Delta x} \sum_{|\beta| \geq 3} \frac{1}{|\beta|!} (A^\alpha_p (\hat{y}^{\alpha,+}_p)^\beta + B^\alpha_p (\hat{y}^{\alpha,-}_p)^\beta) D^\beta \phi,$$

where, due to the truncation of the stencil, the scheme is not central and therefore the terms for odd $|\beta|$ do not cancel out. If only one side of the stencil oversteps then for $|\beta| = 3$

$$\frac{A^\alpha_p (\hat{y}^{\alpha,+}_p)^\beta + B^\alpha_p (\hat{y}^{\alpha,-}_p)^\beta}{\Delta x} \sim \mathcal{O}(\sqrt{\Delta x}),$$

whereas if both sides overstep then the error from interpolation dominates and is $\mathcal{O}(1)$ for points $\mathcal{O}(\Delta x)$ from the boundary, as seen at the end of the proof of Proposition 2.2. □

Remark 2.1 (Two-sided overstepping) We note that it is possible for both sides of the stencil to overstep if the diffusion direction σ^α_p is (almost) parallel to the domain boundary, for points close to a locally convex smooth boundary with high curvature in that direction, as well as close to corners; see Remark 2.4 and Table 5 below.

The scheme is consistent at points with two-sided overstepping if the truncated scheme is not interpolated at the boundary but uses the exact boundary values. In that case, the consistency error for those points is $\mathcal{O}(\Delta x)$.

2.3 Properties of the Truncated Stencil

The changes in the finite difference weights of scheme (2.1) introduced by the truncation, modify the positivity conditions given in Lemma 4.1 in [9]. We will show that the scheme remains conditionally L_∞-stable and monotone, but the CFL conditions are more restrictive in the truncated case for time-stepping schemes with $\theta < 1$. We start by writing the scheme on a discrete time-space grid with mesh parameters Δt and Δx as

$$\begin{aligned}
&\hat{L}^{\alpha}_{\Delta x}[\mathcal{I}_{\Delta x}\phi(t,\cdot)](t_n,x_j)\\
&\quad=\sum_{p=1}^{M}\frac{1}{2\Delta x}\Big[A_p^{\alpha,n}(\mathcal{I}_{\Delta x}\phi(t_n,\cdot))(x_j+\hat{y}_p^{\alpha,+})-(A_p^{\alpha,n}+B_p^{\alpha,n})\phi(t_n,x_j)\\
&\qquad+B_p^{\alpha,n}(\mathcal{I}_{\Delta x}\phi(t_n,\cdot))(x_j+\hat{y}_p^{\alpha,-})\Big]\\
&\quad=\sum_{p=1}^{M}\Bigg\{\sum_{i\in\mathcal{N}(x_j+\hat{y}_p^{\alpha,+})}\frac{1}{2\Delta x}\Big[A_p^{\alpha,n}w_i(x_j+\hat{y}_p^{\alpha,+})\Big](\phi(t_n,x_i)-\phi(t_n,x_j))\\
&\qquad+\sum_{i\in\mathcal{N}(x_j+\hat{y}_p^{\alpha,-})}\frac{1}{2\Delta x}\Big[B_p^{\alpha,n}w_i(x_j+\hat{y}_p^{\alpha,-})\Big](\phi(t_n,x_i)-\phi(t_n,x_j))\Bigg\}\\
&\quad=\sum_{i=1}^{N}\sum_{p=1}^{M}\frac{A_p^{\alpha,n}w_i(x_j+\hat{y}_p^{\alpha,+})+B_p^{\alpha,n}w_i(x_j+\hat{y}_p^{\alpha,-})}{2\Delta x}(\phi(t_n,x_i)-\phi(t_n,x_j))\\
&\quad=\sum_{i=1}^{N}\hat{l}_{j,i}^{\alpha,n}(\phi(t_n,x_i)-\phi(t_n,x_j)),
\end{aligned}\tag{2.9}$$

where $\mathcal{N}$ is the set of neighbours as in (1.7), and

$$\hat{l}_{j,i}^{\alpha,n}=\sum_{p=1}^{M}\frac{A_p^{\alpha,n}w_i(x_j+\hat{y}_p^{\alpha,+}(t_n,x_j))+B_p^{\alpha,n}w_i(x_j+\hat{y}_p^{\alpha,-}(t_n,x_j))}{2\Delta x}.$$

The first equality follows from (2.1), the second from (1.7) and since for all $1\le i,j\le N$

$$w_j(x)\ge 0,\quad w_i(x_j)=\delta_{ij},\quad\text{and}\quad\sum_{i\in\mathcal{N}(x)}w_i(x)\equiv 1,\tag{2.10}$$

for multi-linear interpolation. Here,

$$\sum_{i=1}^{N}\hat{l}_{j,i}^{\alpha,n}=\sum_{p=1}^{M}\frac{A_p^{\alpha,n}+B_p^{\alpha,n}}{2\Delta x}\ge\frac{M}{\Delta x},$$

with equality only in the absence of domain overstepping for all $p\in[[1,M]]$ at (t_n,x_j,α).

Writing the overall scheme in the form (1.9) of Definition 1.1, we have that

$$\begin{aligned}
&\sup_{\alpha}\Bigg\{\Bigg[1+\theta\Delta t_n\Bigg(\sum_{p=1}^{M}\frac{A_p^{\alpha,n}+B_p^{\alpha,n}}{2\Delta x}-\hat{l}_{j,j}^{\alpha,n}-c_j^{\alpha,n-1+\theta}\Bigg)\Bigg]U_j^n-\theta\Delta t_n\sum_{i\neq j}\hat{l}_{j,i}^{\alpha,n}U_i^n+\\
&\quad-\Bigg[1-(1-\theta)\Delta t_n\Bigg(\sum_{p=1}^{M}\frac{A_p^{\alpha,n-1}+B_p^{\alpha,n-1}}{2\Delta x}-\hat{l}_{j,j}^{\alpha,n-1}-c_j^{\alpha,n-1+\theta}\Bigg)\Bigg]U_j^{n-1}+\\
&\quad-(1-\theta)\Delta t_n\sum_{i\neq j}\hat{l}_{j,i}^{\alpha,n-1}U_i^{n-1}-\Delta t_n f_j^{\alpha,n-1+\theta}\Bigg\}=0.
\end{aligned}\tag{2.11}$$

It is straightforward to write down the expressions for the coefficients in (1.9):

$$\begin{aligned}
\mathcal{B}^{\alpha,n,n}_{j,j} &= 1+\theta\Delta t_n\left(\sum_{p=1}^{M}\frac{A^{\alpha,n}_p+B^{\alpha,n}_p}{2\Delta x}-\hat{l}^{\alpha,n}_{j,j}-c^{\alpha,n-1+\theta}_j\right),\\
\mathcal{B}^{\alpha,n,n-1}_{j,j} &= 1-(1-\theta)\Delta t_n\left(\sum_{p=1}^{M}\frac{A^{\alpha,n-1}_p+B^{\alpha,n-1}_p}{2\Delta x}-\hat{l}^{\alpha,n-1}_{j,j}-c^{\alpha,n-1+\theta}_j\right),\\
\mathcal{B}^{\alpha,n,n}_{j,i} &= \theta\Delta t_n\,\hat{l}^{\alpha,n}_{j,i},\qquad \mathcal{B}^{\alpha,n,n-1}_{j,i} = (1-\theta)\Delta t_n\,\hat{l}^{\alpha,n-1}_{j,i}.
\end{aligned}$$

Remark 2.2 In writing down (2.9), we assumed that the value at the boundary is interpolated from other mesh points, which is feasible on rectangular cuboids, but not for general domain boundaries. In both cases, the Dirichlet boundary value at $x_j+\hat{y}^{\alpha,\pm}_p$ can be used. This has the advantage that interpolation error is avoided. Moreover, as this value then contributes to the right-hand-side f of Eq. (2.11) instead of the off-diagonal matrix elements, the system matrix becomes more diagonally dominant. This is advantageous for the iterative solution, see Sect. 3.4.

The next proposition contains the positivity conditions for the coefficients $\mathcal{B}$ defined above.

Proposition 2.4 *The scheme* (2.11) *is of positive type if the following conditions hold,*

$$(1-\theta)\Delta t_n\left[\sum_{p=1}^{M}\frac{A^{\alpha,n-1}_p+B^{\alpha,n-1}_p}{2\Delta x}-c^{\alpha,n-1+\theta}_i\right]\le 1,\quad \textit{and}\quad \theta\Delta t_n c^{\alpha,n-1+\theta}_i\le 1,\quad (2.12)$$

for all α, n, i.

Corollary 2.5 *In the case of overstepping and* $\theta<1$, *monotonicity requires that* $\Delta t\sim\mathcal{O}(\Delta x^{3/2})$ *if only one side of the diffusion stencils oversteps, or* $\Delta t\sim\mathcal{O}(\Delta x^2)$ *if both sides overstep. However, if the stencil is not truncated, the positivity condition remains as in* [9], *that is* $\Delta t\sim\mathcal{O}(\Delta x)$.

Proof From Corollary 2.3, if the corresponding stencil is truncated on one side $A^{\alpha,n-1}_{\cdot}+B^{\alpha,n-1}_{\cdot}\sim\mathcal{O}(\Delta x^{-1/2})$ for sufficiently small Δx, $A^{\alpha,n-1}_{\cdot}+B^{\alpha,n-1}_{\cdot}\sim\mathcal{O}(\Delta x^{-1})$ if both sides are truncated, whereas if there is no overstepping, $A^{\alpha,n-1}_{\cdot}+B^{\alpha,n-1}_{\cdot}\sim\mathcal{O}(1)$. □

The L^∞-stability follows from the proof of Lemma 4.1 in [9] and the new CFL conditions in Proposition 2.4.

2.4 Numerical Experiments

To test the truncation of the stencil, we consider Problems A and B in Section 9.3 from [9]. Both problems follow the formulation in (1.1)–(1.3) with homogeneous Dirichlet boundary conditions and have smooth solutions.

Problem A (see Section 9.3 from [9]). It has exact solution $u(t,x_1,x_2)=\left(\frac{3}{2}-t\right)\sin x_1\sin x_2$, and coefficients and control set are given by

$$\begin{aligned}
f^\alpha = \left(\frac{1}{2}-t\right)\sin x_1\sin x_2+\left(\frac{3}{2}-t\right)\Big[&\sqrt{\cos^2 x_1\sin^2 x_2+\sin^2 x_1\cos^2 x_2+}\\
&-2\sin(x_1+x_2)\cos(x_1+x_2)\cos x_1\cos x_2\Big],
\end{aligned}$$

$$c^\alpha = 0,\ b^\alpha = \alpha, \quad \sigma^\alpha = \sqrt{2}\begin{pmatrix}\sin(x_1 + x_2)\\ \cos(x_1 + x_2)\end{pmatrix}, \quad \mathcal{A} = \{\alpha \in \mathbb{R}^2 : \alpha_1^2 + \alpha_2^2 = 1\}.$$

Problem B (see Section 9.3 from [9]). It has exact solution $u(t, x_1, x_2) = (2 - t)\sin(x_1)$ $\sin(x_2)$, and coefficients and control set

$$f^\alpha = (1 - t)\sin x_1 \sin x_2 - 2\alpha_1\alpha_2(2 - t)\cos x_1 \cos x_2,$$
$$c^\alpha = 0,\ b^\alpha = 0, \quad \sigma^\alpha = \sqrt{2}\begin{pmatrix}\alpha_1\\ \alpha_2\end{pmatrix}, \quad \mathcal{A} = \{\alpha \in \mathbb{R}^2 : \alpha_1^2 + \alpha_2^2 = 1\}.$$

Both problems are solved on the domain $(t, x_1, x_2) \in [0, T] \times [-\pi, \pi]^2$ with $T = \frac{1}{2}$. We discretize the spatial domain using Cartesian grids with $N_x \times N_x$ equispaced nodes and for the control set $\mathcal{A}$ we take N_α equally spaced points. Here, $\mathcal{I}_{\Delta x}$ is the usual bilinear interpolator on rectangles.

For illustration of the stencil and its non-locality, the top row of Fig. 2 represents the stencil for Problems A and B on a Cartesian grid of 11 × 11 points and 10 points in the control set $\mathcal{A}$. Colour coded lines link the stencil points with the node where the numerical solution is computed, the different colours correspond to the different $\hat{y}^{\alpha,\cdot}_{\cdot}$. On top of some of the stencil points we print the value of the finite difference weights, for compactness we set $A \equiv A^\alpha_{2,1}(x)$, $B \equiv B^\alpha_{2,1}(x)$ and $C \equiv (\mu^\alpha_{2,2}(x))^{-1}$, following the notation in (2.3) and (2.2). The bottom row of Fig. 2 represents the non-locality of the diffusion stencil by counting the number of stencil points at a given distance from the central node. The distance is measured as multiples of Δx and given by $\left\lfloor \frac{(\sigma^\alpha(x))_i}{\sqrt{\Delta x}} \right\rfloor$, where the grid is of size 641 × 641 and 10 points in the control set $\mathcal{A}$.

Problems A and B were obviously chosen in [9] for their periodic solutions, to be able to analyse the convergence of the scheme without the complication of boundary conditions. Here, we do not make use of the periodicity but only use the values at the boundary and not outside the domain.

We note that the problems being linear in t, a single time step with $\Delta t = T$ suffices to obtain an exact solution in t. However, in order to check the effect of the truncation on the stability, in addition to $\Delta t = T$, we also investigate Δt equal to $\frac{\Delta x}{4}$, $\Delta x^{3/2}$, and Δx^2. We report the ∞-norm of the errors over two regions: the first one comprising the whole domain, and the second one comprising part of the interior of the domain.

We consider explicit and implicit time stepping schemes, corresponding to $\theta = 0$ and $\theta = 1$ respectively. For the explicit scheme in the case of overstepping we test the following modifications of the scheme:

1. truncation of the stencil as discussed in Sect. 2.2 (Table 1 for Problem A and Table 12 for Problem B);
2. constant extrapolation of the boundary value in the direction of the semi-Lagrangian step (Table 2 for Problem A and Table 13 for Problem B);
3. linear extrapolation of the boundary value in the direction of the semi-Lagrangian step (Table 3 for Problem A and Table 14 for Problem B).

For the implicit case we only consider the first modification, i.e. truncation of the stencil (Table 4 for Problem A and Table 15 for Problem B).

The results confirm the impact of the truncation on the stability of the scheme, when $\theta = 0$. However, when $\theta = 1$, we do not observe any instability regardless of the size of the time step. When stable, the truncation of the stencil outperforms the two extrapolations of the boundary conditions considered. Furthermore, as the mesh and time steps are refined,

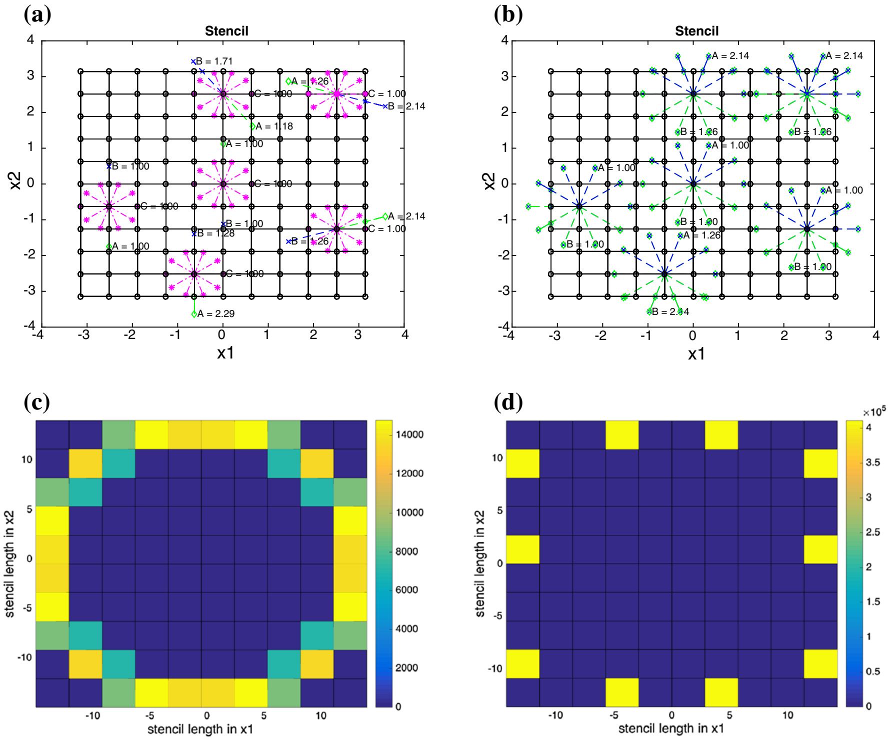

Fig. 2 Graphical representation of the stencil over a two-dimensional Cartesian grid of size 11 × 11 and 10 equally spaced points in the control set $\mathcal{A}$. The finite difference weights corresponding to some of the points are printed, where for simplicity the weights are labelled $A \equiv A^{\alpha}_{2,1}(x)$, $B \equiv B^{\alpha}_{2,1}(x)$ and $C \equiv (\mu^{\alpha}_{2,2}(x))^{-1}$, following the notation in (2.3) and (2.2). To illustrate the non-locality of the scheme as the grid is refined, the second row represents the histograms of the shortest displacement from the central node for a grid of size 641 × 641 for both problems. The radius of the stencil in σ^{α} is 14.27 for this grid, given by $\frac{\|\sigma^{\alpha}\|_2}{\sqrt{\Delta x}} = \sqrt{640/\pi}$. **a** Stencil for Problem A in [9] on a Cartesian 11 × 11 grid for 10 sample points in the control set $\mathcal{A}$. **b** Stencil for Problem B in [9] on a Cartesian 11 × 11 grid for 10 sample points in the control set $\mathcal{A}$. **c** Histogram of $\left\lfloor \frac{(\sigma^{\alpha}(x))_i}{\sqrt{\Delta x}} \right\rfloor$ in Problem A for all $x \in \Omega_{\Delta x}$ where $\Omega_{\Delta x}$ is a Cartesian grid with $\Delta x = \frac{2\pi}{640}$, 10 points in the control set $\mathcal{A}$, and $i \in \{1, 2\}$ is the dimension index. **d** Histogram of $\left\lfloor \frac{(\sigma^{\alpha}(x))_i}{\sqrt{\Delta x}} \right\rfloor$ in Problem B for all $x \in \Omega_{\Delta x}$ where $\Omega_{\Delta x}$ is a Cartesian grid with $\Delta x = \frac{2\pi}{640}$, 10 points in the control set $\mathcal{A}$, and $i \in \{1, 2\}$ is the dimension index

only the truncated scheme, if stable, achieves convergence orders close to $\mathcal{O}(\Delta x)$ when the error at $t = T$ is measured on the entire spatial grid. This can be explained without rigorous proof by the observation that the truncation error of order $\sqrt{\Delta x}$ is restricted to a boundary layer of width $\sqrt{\Delta x}$. Therefore, as seen from the last two columns in Table 4, choosing Δt of order higher than 1 in Δx does not improve the accuracy of the numerical results and leads to computational inefficiency.

Remark 2.3 Regarding the discretization of the control set, we take $N_{\alpha} = 40$ equally spaced points. For this choice, the discretization error of the LISL scheme is found to dominate the control discretization error for the problems and the space-time mesh sizes considered.

Table 1 Results using the truncation of the stencil for explicit method with $N_\alpha = 40$ for Problem A

N_x	$\Delta t = T$		$\Delta t = \frac{\Delta x}{4}$		$\Delta t = \Delta x^{\frac{3}{2}}$		$\Delta t = \Delta x^2$	
	Error	Rate	Error	Rate	Error	Rate	Error	Rate
(a) Error in L^∞-norm over $\Omega_{\Delta x}$								
41	1.42e−01	–	4.39e−02	–	4.39e−02	–	4.36e−02	–
81	1.04e−01	0.45	2.12e−02	1.05	2.11e−02	1.06	2.11e−02	1.05
161	7.36e−02	0.50	1.10e−02	0.94	1.10e−02	0.94	1.10e−02	0.94
321	5.28e−02	0.48	1.34e+23	−83.33	5.77e−03	0.93	5.76e−03	0.93
641	3.77e−02	0.48	5.07e+89	−221.17	3.10e−03	0.90	3.10e−03	0.89
(b) Error in L^∞-norm over $\Omega_{\Delta x} \cap [-\pi/2, \pi/2]^2$								
41	8.61e−02	–	4.38e−02	–	4.42e−02	–	4.35e−02	–
81	4.22e−02	1.03	2.12e−02	1.05	2.11e−02	1.06	2.11e−02	1.05
161	2.14e−02	0.98	1.10e−02	0.94	1.10e−02	0.95	1.10e−02	0.94
321	1.10e−02	0.96	1.84e+13	−50.57	5.71e−03	0.95	5.70e−03	0.95
641	5.96e−03	0.88	1.06e+72	−195.20	3.08e−03	0.89	3.08e−03	0.89

Table 2 Results using constant extrapolation of the boundary condition for explicit method with $N_\alpha = 40$ for Problem A

N_x	$\Delta t = T$		$\Delta t = \frac{\Delta x}{4}$		$\Delta t = \Delta x^{\frac{3}{2}}$		$\Delta t = \Delta x^2$	
	Error	Rate	Error	Rate	Error	Rate	Error	Rate
(a) Error in L^∞-norm over $\Omega_{\Delta x}$								
41	1.36e+00	–	3.68e−01	–	3.72e−01	–	3.65e−01	–
81	1.89e+00	-0.48	2.61e−01	0.49	2.62e−01	0.51	2.60e−01	0.49
161	2.67e+00	-0.49	1.80e−01	0.54	1.80e−01	0.54	1.80e−01	0.53
321	3.77e+00	-0.50	1.27e−01	0.51	1.27e−01	0.51	1.27e−01	0.51
641	5.34e+00	-0.50	9.18e−02	0.47	9.18e−02	0.47	9.18e−02	0.46
(b) Error in L^∞-norm over $\Omega_{\Delta x} \cap [-\pi/2, \pi/2]^2$								
41	1.59e−01	–	1.04e−01	–	1.05e−01	–	1.03e−01	–
81	8.15e−02	0.96	5.25e−02	0.99	5.26e−02	1.00	5.22e−02	0.98
161	4.22e−02	0.95	2.67e−02	0.98	2.66e−02	0.98	2.66e−02	0.97
321	2.18e−02	0.95	1.36e−02	0.97	1.36e−02	0.97	1.36e−02	0.97
641	1.21e−02	0.85	8.21e−03	0.73	8.20e−03	0.73	8.19e−03	0.73

Remark 2.4 Corollary 2.5 shows two different CFL conditions for the truncated stencil, the first one for diffusion stencils where only one side oversteps and a second one when both sides overstep. The results in Table 1 for Problem A and Table 12 for Problem B correspond to the former situation. To check the sharpness of the latter, we shift the spatial domain in Problem A in both directions by $\frac{7\pi}{8}$. The new spatial domain is thus $\bar{\Omega} = [\frac{-\pi}{8}, \frac{15\pi}{8}]^2$. Note that the solution itself is periodic with period 2π. This problem differs from the original one in that both sides of the diffusion stencil overstep for mesh points within a distance of $\mathcal{O}(\sqrt{\Delta x})$ to the bottom left corner, located at $(\frac{-\pi}{8}, \frac{-\pi}{8})$, where $\sigma^\alpha = (-1, 1)^T$. In Table 5

Table 3 Results using linear extrapolation for points out of the domain for explicit method with $N_\alpha = 40$ for Problem A

N_x	$\Delta t = T$		$\Delta t = \frac{\Delta x}{4}$		$\Delta t = \Delta x^{\frac{3}{2}}$		$\Delta t = \Delta x^2$	
	Error	Rate	Error	Rate	Error	Rate	Error	Rate
(a) Error in L^∞-norm over $\Omega_{\Delta x}$								
41	1.59e−01	–	1.04e−01	–	1.05e−01	–	1.03e−01	–
81	8.15e−02	0.96	5.25e−02	0.99	5.26e−02	1.00	5.22e−02	0.98
161	4.28e−02	0.93	5.62e−01	−3.42	5.63e−01	−3.42	5.58e−01	−3.42
321	2.75e−02	0.64	4.41e+03	−12.94	6.00e+03	−13.38	8.00e+03	−13.81
641	1.85e−02	0.57	2.77e+20	−55.80	2.70e+20	−55.32	1.37e+21	−57.25
(b) Error in L^∞-norm over $\Omega_{\Delta x} \cap [-\pi/2, \pi/2]^2$								
41	1.59e−01	–	1.04e−01	–	1.05e−01	–	1.03e−01	–
81	8.15e−02	0.96	5.25e−02	0.99	5.26e−02	1.00	5.22e−02	0.98
161	4.22e−02	0.95	2.67e−02	0.98	2.66e−02	0.98	2.66e−02	0.97
321	2.18e−02	0.95	1.96e+00	−6.20	2.07e+00	−6.28	2.23e+00	−6.39
641	1.21e−02	0.85	9.26e+14	−48.75	3.18e+15	−50.45	3.01e+15	−50.26

Table 4 Results using truncation for points out of the domain for implicit method with $N_\alpha = 40$ for Problem A

N_x	$\Delta t = T$		$\Delta t = \frac{\Delta x}{4}$		$\Delta t = \Delta x^{\frac{3}{2}}$		$\Delta t = \Delta x^2$	
	Error	Rate	Error	Rate	Error	Rate	Error	Rate
(a) Error in L^∞-norm over $\Omega_{\Delta x}$								
41	3.25e−02	–	4.21e−02	–	4.17e−02	–	4.24e−02	–
81	1.59e−02	1.03	2.08e−02	1.02	2.08e−02	1.01	2.09e−02	1.02
161	8.39e−03	0.92	1.09e−02	0.93	1.09e−02	0.93	1.10e−02	0.93
321	4.38e−03	0.94	5.75e−03	0.93	5.75e−03	0.93	5.76e−03	0.93
641	2.37e−03	0.89	3.09e−03	0.89	3.10e−03	0.89	3.10e−03	0.89
(b) Error in L^∞-norm over $\Omega_{\Delta x} \cap [-\pi/2, \pi/2]^2$								
41	3.25e−02	–	4.21e−02	–	4.17e−02	–	4.24e−02	–
81	1.59e−02	1.03	2.08e−02	1.02	2.08e−02	1.01	2.09e−02	1.02
161	8.39e−03	0.92	1.09e−02	0.93	1.09e−02	0.93	1.10e−02	0.93
321	4.35e−03	0.95	5.68e−03	0.94	5.69e−03	0.94	5.70e−03	0.95
641	2.37e−03	0.88	3.07e−03	0.89	3.08e−03	0.89	3.08e−03	0.89

we report the results for the explicit method using the truncation of the stencil. As expected, we find that we now need $\Delta t \sim \Delta x^2$ for stability.

Remark 2.5 For the explicit method using the truncation of the stencil, i.e. Tables 1, 12, and 5, focusing on the $\Delta t = T$ case, we notice that the convergence rate over the whole mesh $\Omega_{\Delta x}$ is approximately 0.5, whereas it is approximately 1.0 when the error is measured in the interior of the mesh. We also notice that there is a significant difference between the magnitude of the errors if measured over the whole grid or on a region in the interior. The

Table 5 Results using the truncation of the stencil for explicit method with $N_\alpha = 40$ for Problem A on a shifted domain, as described in Remark 2.4

N_x	$\Delta t = T$		$\Delta t = \frac{\Delta x}{4}$		$\Delta t = \Delta x^{\frac{3}{2}}$		$\Delta t = \Delta x^2$	
	Error	Rate	Error	Rate	Error	Rate	Error	Rate
(a) Error in L^∞-norm over $\Omega_{\Delta x}$								
41	1.55e−01	–	4.71e−02	–	4.76e−02	–	4.67e−02	–
81	1.12e−01	0.47	1.57e+05	−21.67	7.90e+05	−23.98	2.11e−02	1.15
161	8.04e−02	0.47	1.02e+33	−92.39	1.30e+35	−97.06	1.10e−02	0.94
321	5.80e−02	0.47	6.73e+103	−235.26	5.96e+138	−344.35	5.76e−03	0.93
641	4.22e−02	0.46	8.17e+276	−574.97	NaN	NaN	3.10e−03	0.89
(b) Error in L^∞-norm over $\Omega_{\Delta x} \cap [3\pi/8, 11\pi/8]^2$								
41	8.65e−02	–	4.70e−02	–	4.74e−02	–	4.66e−02	–
81	4.22e−02	1.04	2.07e−02	1.18	2.07e−02	1.19	2.06e−02	1.18
161	2.14e−02	0.98	1.18e+06	−25.76	1.18e+09	−35.73	1.08e−02	0.93
321	1.10e−02	0.96	7.99e+47	−138.96	4.94e+84	−251.21	5.59e−03	0.95
641	5.96e−03	0.88	9.81e+165	−392.28	NaN	NaN	3.02e−03	0.89

difference in the magnitude of the errors may be due to the fact that $\Delta t = T$ does not satisfy the CFL condition and that the CFL condition is more restrictive for points where the stencil is truncated. It is also at these points that the local consistency error is of order $\sqrt{\Delta x}$ as shown in Corollary 2.3. The situation is different for $\Delta t = \Delta x^2$ in the explicit case, or for any Δt in the implicit case. In these cases, the error convergence rates are approximately 1.0 when measured over the whole mesh and the errors over the whole grid and in the interior are comparable in magnitude.

3 Multigrid Preconditioning

In this section, we study the application of multigrid preconditioners together with policy iteration [4] to solve the non-linear system (1.10).

Geometric multigrid requires us to predefine a grid hierarchy based on the geometry of the problem. The variability of the width of the LISL stencil within a given grid (variable coefficients) and through the grid hierarchy makes it difficult, even for simple problems, to design an appropriate grid hierarchy and a good smoother. Moreover, the varying stencil requires us to build the coarse-grid version of the operator algebraically instead of using its coarse grid version, which further limits our knowledge of the problem as we go deeper into the grid hierarchy.

Another aspect to consider is related to the transfer operators. Standard grid interpolations provide approximations using the grid neighbours of a given node, whereas for the LISL stencil, being non-local, the solution at a given node may not be best approximated by its neighbours on the grid but by those on its stencil. These heuristics suggest that the algebraic approach to multigrid, fixing the smoother and building operator dependent intergrid transfer operators, may result in more efficient multigrid preconditioning for LISL discretizations.

Algebraic multigrid (AMG), introduced in [28], constructs “coarse grids” based on the matrix coefficients. However, as pointed out in Section 6.2 of the recent review on preconditioning [36], AMG coarsening may not reduce the number of variables fast enough from one grid to the next. A slow reduction in the number of unknowns and the use of the Galerkin principle to build the coarse system matrix with intergrid transfer operators using weighted averages increase the complexity of the multigrid scheme. To measure the complexity the following quantities are commonly used:

Definition 3.1 The grid complexity c_G is the total number of variables $N_\cdot$, on all multigrid levels, divided by the number of variables on the finest level N_1,

$$c_G = \frac{1}{N_1}\sum_{\ell=1}^{n_{\text{levels}}} N_\ell.$$

Definition 3.2 The algebraic complexity c_A is the total number of non-zero entries, in all matrices A_ℓ, divided by the number of non-zero entries of the finest level operator A_1,

$$c_A = \frac{1}{\text{nnz}(\text{A}_1)}\sum_{\ell=1}^{n_{\text{levels}}} \text{nnz}(\text{A}_\ell).$$

We will find a benefit to the convergence of constructing the “coarse grids” algebraically already for simple examples of LISL matrices (Sect. 3.3, in particular Table 6), and that algebraic construction of the grid hierarchy deals well with the varying LISL stencils (Sect. 3.5). However, there is an increase in complexity of AMG (see Table 6) mainly due to the use of interpolation in LISL discretizations.

Recent and on-going research on algebraic multigrid [24,25] shows how one can construct good multigrid cycles using simplified “intergrid” transfer operators based on aggregation of the unknown variables, thus avoiding the problem of increased complexity on coarser levels. In particular, [25] proves convergence of a simplified two-grid scheme using aggregation for non-singular M-matrices with non-negative row and column sums. We will show that these results apply for LISL discretizations matrices and justify the use of AGMG both theoretically and empirically.

3.1 On the Spectrum of LISL Matrices

To assess the suitability of preconditioning based on geometric multigrid, we start by considering the spectrum of LISL matrices for a simplified model. For illustration, we first calculate the eigenvalues and eigenvectors of the LISL discretization of the diffusion operator with constant coefficients, for any function $u : \mathbb{R}^d \to \mathbb{R}$

$$-\frac{1}{2}\nabla^T(\sigma\sigma^T)\nabla u = -\frac{1}{2}\sum_{i=1}^{d}\sigma_i^2\frac{\partial^2 u}{\partial x_i^2}, \tag{3.1}$$

where $\sigma \in \mathbb{R}^{d\times d}$ is a diagonal matrix with $(\sigma)_{ii} = \sigma_i$.

We start by considering the one-dimensional case on an equispaced grid $\Omega_{\Delta x}$, where $\Delta x > 0$ is the distance between two consecutive nodes. For $\sigma > 0$, we define

$$m := \left\lfloor \frac{\sigma}{\sqrt{\Delta x}} \right\rfloor, \quad \text{and} \quad \gamma := (m+1) - \frac{\sigma}{\sqrt{\Delta x}}, \tag{3.2}$$

where $m \in \mathbb{N}$ denotes the stencil length and $\gamma \in [0, 1]$ is the interpolation weight of the one-dimensional linear interpolation operator, such that for any real function $\phi : \mathbb{R} \to \mathbb{R}$ the linear interpolation operator on $\Omega_{\Delta x}$ is $\mathcal{I}_{\Delta x}(\phi)(x_i + \sqrt{\Delta x \sigma}) = \gamma\phi(x_i + m\Delta x) + (1 - \gamma)\phi(x_i + (m+1)\Delta x)$. Without loss of generality and for simplicity of the notation we assume that $\Delta x = 1$. Denote by L_N the following $N \times N$ Laplacian matrix

$$L_N := \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 2 \end{pmatrix}.$$

Let now $m = 2$ and $\Delta x = 1$, then the LISL discretization matrix is given by

$$L_{SL}^{N,m,\gamma} := \begin{pmatrix} 2 & 0 & -\gamma & -1+\gamma & 0 & \cdots & 0 \\ 0 & 2 & 0 & -\gamma & -1+\gamma & \cdots & 0 \\ -\gamma & 0 & 2 & 0 & -\gamma & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 2 \end{pmatrix}. \tag{3.3}$$

Noticing the structure in the diagonals, we re-write $L_{SL}^{N,m,\gamma}$ as

$$L_{SL}^{N,m,\gamma} = \gamma L_N^m + (1 - \gamma) L_N^{m+1},$$

where $L_N^m = L_{SL}^{N,m,1}$.

Using the properties of Kronecker products we can characterize the eigenvalues of the matrices L_N^m in terms of the eigenvalues of the standard L_N matrices. Denoting by $\lambda(L_N) \in \mathbb{R}^N$ and $V(L_N) \in \mathbb{R}^{N\times N}$ the eigenvalues and eigenvectors of L_N, respectively, we have that

$$\lambda(L_N^m) = \left[\lambda\left(L_{\lceil \frac{N}{m} \rceil}\right) \otimes e_1\right]_N + \left[\lambda\left(L_{\lfloor \frac{N}{m} \rfloor}\right) \otimes \sum_{i=2}^{N} e_i\right]_N,$$
$$V(L_N^m) = \left[V\left(L_{\lceil \frac{N}{m} \rceil}\right) \otimes \begin{pmatrix} 1 & 0 \\ 0 & \mathbf{0}_{m-1} \end{pmatrix}\right]_{N\times N} + \left[V\left(L_{\lfloor \frac{N}{m} \rfloor}\right) \otimes \begin{pmatrix} 0 & 0 \\ 0 & I_{m-1} \end{pmatrix}\right]_{N\times N},$$

where e_i is the i-th canonical basis vector of $\mathbb{R}^N$, I_N is the $N \times N$ identity matrix and $\mathbf{0}_m$ denotes the $m \times m$ zero matrix. By $[A]_{N\times N}$ we mean that we select the first N rows and N columns of A, and similar for $[v]_N$ for a vector N. This is required as N will in general not be a multiple of both m and $m + 1$ so the resulting matrices from the Kronecker product will be of size $\left\lceil \frac{N}{m} \right\rceil m$ and $\left\lceil \frac{N}{m+1} \right\rceil (m + 1)$ which are greater or equal to N.

In the presence of interpolation, that is, when $\gamma \in (0, 1)$, we are unable to provide any closed formula to the eigenvalues and eigenvectors of $L_{SL}^{N,m,\gamma} = \gamma L_N^m + (1 - \gamma) L_N^{m+1}$. Figure 3 contains graphs with the eigenvalues and some eigenvectors of the matrices $L_{SL}^{N,m,\gamma}$, L_N^m, L_N^{m+1} and L_N. The plots show that for LISL discretization matrices, in contrast to the standard case, small eigenvalues are not necessarily associated with smooth modes. As a result, these components cannot be represented accurately on the coarse mesh.

The spectrum of higher-dimensional constant coefficient Laplacians can be inferred from the spectrum of the one-dimensional matrices by means of Kronecker products. Next, we consider the properties of common smoothers when applied to LISL discretization matrices

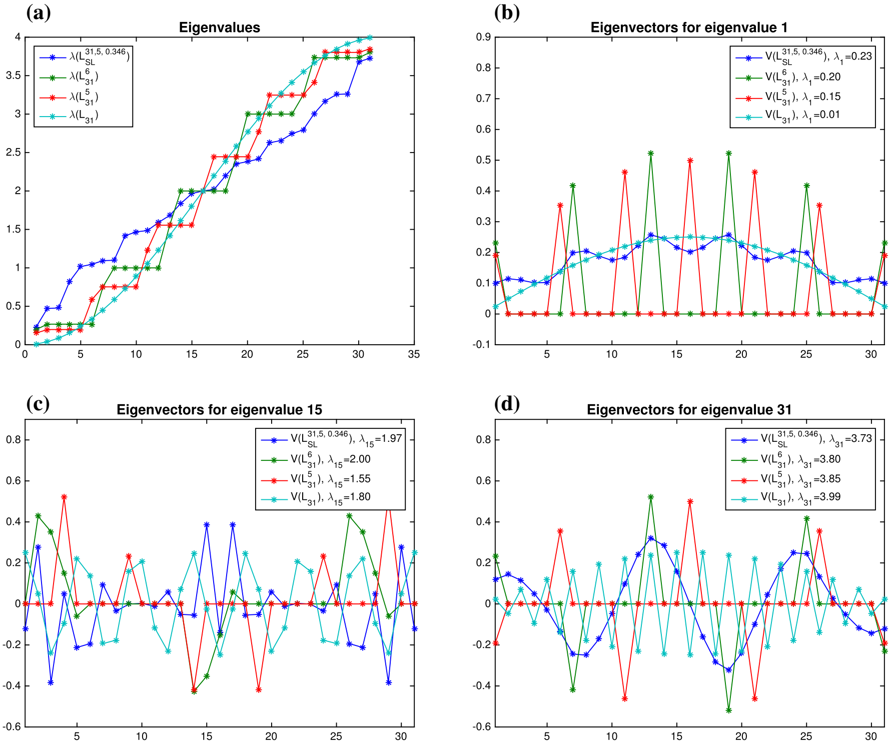

Fig. 3 Eigenvalues of $L_{SL}^{N,m,\gamma}$, L_N^m, L_N^{m+1} and L_N with parameter values $N = 31, m = 5$ and $\gamma = 0.346$ and the eigenvectors corresponding to three eigenvalues of the same matrices. **a** Comparison of the eigenvalues of $L_{SL}^{31,5,0.346}$, L_{31}^5, L_{31}^6 and L_{31} in increasing order. **b** Eigenvectors corresponding to the smallest eigenvalue for $L_{SL}^{31,5,0.346}$, L_{31}^5, L_{31}^6 and L_{31}. **c** Eigenvectors corresponding to the 15-th eigenvalue for $L_{SL}^{31,5,0.346}$, L_{31}^5, L_{31}^6 and L_{31}. **d** Eigenvectors corresponding to the largest eigenvalue for $L_{SL}^{31,5,0.346}$, L_{31}^5, L_{31}^6 and L_{31}

of the two-dimensional Laplacian and conclude with an example illustrating the impact of the diffusion coefficient on the convergence of geometric multigrid cycles.

3.2 Local Fourier Analysis of the Smoothers

We seek to analyse how a varying size stencil affects the properties of the standard Gauss-Seidel smoother. We base the analysis on Local Fourier Analysis (LFA) as described in Chapter 4 of [32] and state the *smoothing factors* μ_{loc} of Gauss-Seidel iterations when applied to wide stencil finite difference discretizations. The key to the analysis is the use of grid functions of the form $\varphi(\boldsymbol{\theta}, \boldsymbol{x}) = e^{i\boldsymbol{\theta}\cdot\boldsymbol{x}}$, where i is the imaginary unit, $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{\theta} \in [-\pi, \pi)^d$ and $\cdot$ is the inner product for vectors in $\mathbb{R}^d$. For simplicity we consider equispaced grids $\Omega_{\Delta x}$ with refinement parameter $\Delta x > 0$. Therefore, any $\boldsymbol{x} \in \Omega_{\Delta x}$ can be written as $\boldsymbol{x} \equiv \boldsymbol{x}_0 + \kappa \Delta x$ for some fixed $\boldsymbol{x}_0 \in \Omega_{\Delta x}$ and $\kappa \in \mathbb{Z}^d$. It is thus convenient to rescale the exponent of φ by Δx^{-1}.

The functions φ are important since, as shown in Lemma 4.2.1 of [32], “all grid functions $\varphi(\boldsymbol{\theta}, \boldsymbol{x})$ are (formal) eigenfunctions of any discrete operator which can be described by a difference stencil”. This property allows us to associate to each discrete finite difference operator $L_{\Delta x}$ a so-called symbol $\tilde{L}_{\Delta x}(\boldsymbol{\theta})$ defined by

$$L_{\Delta x}\varphi(\boldsymbol{\theta},\boldsymbol{x}) = \sum_{\kappa\in\mathbb{Z}^d} s_\kappa e^{i\boldsymbol{\theta}\cdot\kappa} = \tilde{L}_{\Delta x}(\boldsymbol{\theta})e^{i\boldsymbol{\theta}\cdot\kappa}, \tag{3.4}$$

where $s_\kappa \in \mathbb{R}$ is the finite difference coefficient at the location κ with respect to the node $\boldsymbol{x}_0$.

As in [32], we consider smoothers formed by a splitting $L_{\Delta x} = L^+_{\Delta x} + L^-_{\Delta x}$ of the discrete operator, i.e.

$$S_{\Delta x} = \left(L^+_{\Delta x}\right)^{-1} L^-_{\Delta x}.$$

Lemma 4.3.1 in [32] derives the expression for the symbol for the smoother as

$$\tilde{S}_{\Delta x}(\boldsymbol{\theta}) := \frac{\tilde{L}^-_{\Delta x}(\boldsymbol{\theta})}{\tilde{L}^+_{\Delta x}(\boldsymbol{\theta})},$$

where $\tilde{L}^+_{\Delta x}$ and $\tilde{L}^-_{\Delta x}$ are defined as for $L_{\Delta x}$ in (3.4).

With multigrid, the objective of the smoother is to dampen error components not reduced by the coarse grid correction. Therefore, assessing the properties of a given smoother requires fixing the coarse grid correction. We limit the study to the simplest coarsening strategy, that is if $\Omega_{\Delta x}$ is the fine grid then $\Omega_{2\Delta x}$ is the coarse grid. This leads to the definition of low and high frequencies below.

Definition 3.3 (*Definition* 4.2.1 *in* [32]) For the coarsening considered, we define the high and low frequencies as follows:

$$\varphi(\boldsymbol{\theta},\cdot) \text{ low frequency component } \Longleftrightarrow \boldsymbol{\theta} \in T^{\text{low}} := \left[-\frac{\pi}{2},\frac{\pi}{2}\right)^d;$$
$$\varphi(\boldsymbol{\theta},\cdot) \text{ high frequency component } \Longleftrightarrow \boldsymbol{\theta} \in T^{\text{high}} := [-\pi,\pi)^d \Big\backslash \left[-\frac{\pi}{2},\frac{\pi}{2}\right)^d.$$

Definition 3.4 (*Definition* 4.3.1 *in* [32]) The smoothing factor for standard coarsening is

$$\mu_{\text{loc}} = \mu_{\text{loc}}(S_{\Delta x}) := \sup\left\{|\tilde{S}_{\Delta x}(\boldsymbol{\theta})| : \boldsymbol{\theta} \in T^{\text{high}}\right\}.$$

We employ these definitions to compare the smoothing factors for the standard two-dimensional Laplacian, setting $d = 2$, discretised using standard local finite differences and the LISL discretization.

Example 3.1 (Example 4.3.4 in [32]) The smoothing factor for the Gauss-Seidel smoother for the standard Laplacian discretisation is given by

$$\mu_{\text{loc}} = \sup\left\{\left|\frac{e^{i\theta_1}+e^{i\theta_2}}{4-e^{-i\theta_1}-e^{-i\theta_2}}\right| : \boldsymbol{\theta} \in T^{\text{high}}\right\}.$$

Similarly, the smoothing factor for the LISL scheme can be derived. In the present case of pure diffusion, Schemes 1–3 coincide.

Example 3.2 Proceeding as in [32] for Example 3.1, the symbols $L^+_{\Delta x}$ and $L^-_{\Delta x}$ for the LISL discretizations are

$$\tilde{L}^+_{\Delta x}(\boldsymbol{\theta}) = \frac{1}{\Delta x}(4-\gamma_1 e^{-im_1\theta_1}-(1-\gamma_1)e^{-i(m_1+1)\theta_1}-\gamma_2 e^{-im_2\theta_2}-(1-\gamma_2)e^{-i(m_2+1)\theta_2}),$$
$$\tilde{L}^-_{\Delta x}(\boldsymbol{\theta}) = -\frac{1}{\Delta x}(\gamma_1 e^{im_1\theta_1}+(1-\gamma_1)e^{i(m_1+1)\theta_1}+\gamma_2 e^{im_2\theta_2}+(1-\gamma_2)e^{i(m_2+1)\theta_2}),$$

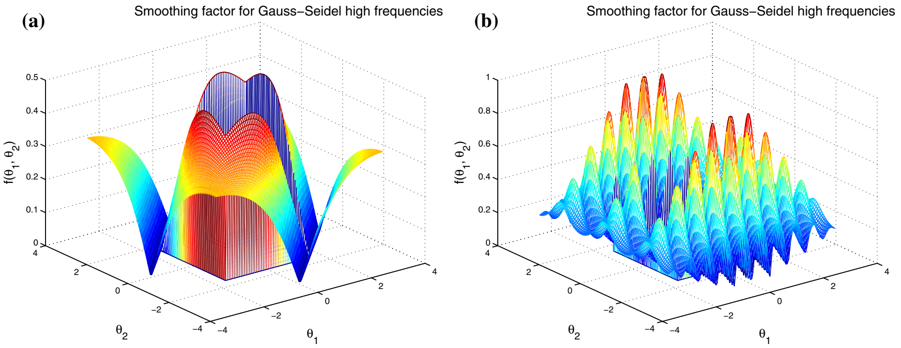

Fig. 4 Representation of the smoothing factor for high frequencies, i.e. $\theta \in [-\pi, \pi]^2 \setminus [-\pi/2, \pi/2]^2$, for the Gauss-Seidel iteration for the classical fixed stencil Finite Difference (FD) and the LISL schemes of the two-dimensional Laplacian operator. The maxima calculated numerically are 0.49 (theoretical value is 0.5) for the fixed stencil FD and 0.95 for the LISL scheme (lower is better). **a** Smoothing factor Gauss-Seidel applied to standard FD approximation of the Laplacian. **b** Smoothing factor Gauss-Seidel applied to LISL approximation of the Laplacian with stencil parameters $((m_1 = 9, m_2 = 3), (\gamma_1 = 0.5, \gamma_2 = 1))$

where m_i and γ_i are given by (3.2) replacing σ by σ_i. For compactness of notation, define

$$g(\theta, \gamma, m) := \gamma e^{im\theta} + (1 - \gamma)e^{i(m+1)\theta},$$

then the smoothing factor for a Gauss-Seidel smoother with standard coarsening and the LISL scheme is given by

$$\mu_{\text{loc}}(S^{SL}_{\Delta x}) = \sup\left\{\left|\frac{g_1 + g_2}{4 - \bar{g_1} - \bar{g_2}}\right| : \boldsymbol{\theta} \in T^{\text{high}}\right\}, \tag{3.5}$$

for $\boldsymbol{\theta} \in T^{\text{high}}$, where $g_1 \equiv g(\theta_1, \gamma_1, m_1)$, $g_2 \equiv g(\theta_2, \gamma_2, m_2)$, and $\bar{c}$ denotes the complex conjugate of the complex number c.

From (3.5) we see that as the non-locality of the discretization grows, i.e. $m_\cdot \to \infty$ then the smoothing factor approaches 1 (no smoothing) and so highly oscillatory modes will be transferred to the coarser subspace. Figure 4 compares the smoothing factor for the fixed stencil 5 point discretization and a specific semi-Lagrangian stencil.

Example 3.3 We can generalise the results in the previous example to the case of diffusion given by a vector $(\sigma_1, \sigma_2)^T$ not necessarily parallel to any of the axes. If σ_1 and σ_2 have the same sign, then

$$\tilde{L}^+_{\Delta x}(\boldsymbol{\theta}) = \frac{1}{\Delta x}(2 - \bar{g}(\theta_1, \gamma_1, m_1)\bar{g}(\theta_2, \gamma_2, m_2)),$$
$$\tilde{L}^-_{\Delta x}(\boldsymbol{\theta}) = -\frac{1}{\Delta x}(g(\theta_1, \gamma_1, m_1)g(\theta_2, \gamma_2, m_2)).$$

If, however, σ_1 and σ_2 have different signs, then

$$\tilde{L}^+_{\Delta x}(\boldsymbol{\theta}) = \frac{1}{\Delta x}(2 - g(\theta_1, \gamma_1, m_1)\bar{g}(\theta_2, \gamma_2, m_2)),$$
$$\tilde{L}^-_{\Delta x}(\boldsymbol{\theta}) = -\frac{1}{\Delta x}(\bar{g}(\theta_1, \gamma_1, m_1)g(\theta_2, \gamma_2, m_2)).$$

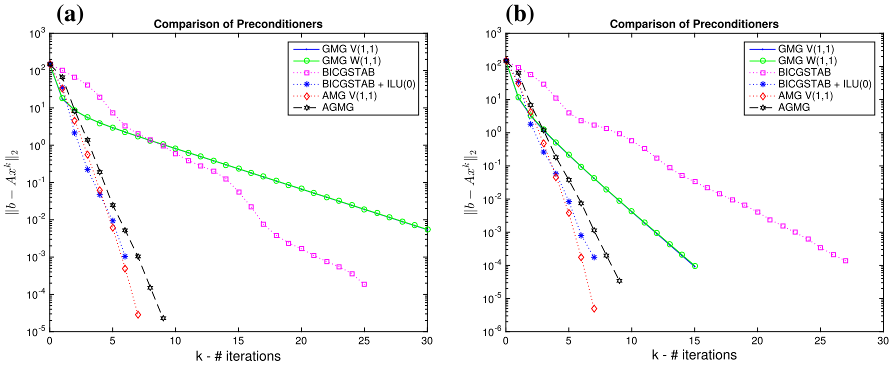

Fig. 5 Residual $\|b - Ax^k\|_2$ in the Eucledian norm at the end of the k-th iteration of different geometric and algebraic multigrid cycles when solving (3.1) on equispaced Cartesian grid of $[0, 1]^2$ with 257 nodes per dimension and with homogeneous Dirichlet boundary conditions. Geometric $V(\nu_1, \nu_2)$ and $W(\nu_1, \nu_2)$ cycles are considered, where ν_1 and ν_2 denote the number of pre- and post-smoothing steps. Their performance is compared to the iterative method BICGSTAB with and without preconditioner, and to two algebraic algorithms, AMG and AGMG from [28] and [25], respectively (see also Sects. 3.4, 3.5). Notice the almost overlapping of lines for geometric $V(\nu_1, \nu_2)$ and $W(\nu_1, \nu_2)$ cycles for equal ν_1 and ν_2 (see also Table 6, which shows almost identical rates for $\ell = 8$). **a** Residual after the k-th iteration for $\sigma = 2I_2$. **b** Residual after the k-th iteration for $\sigma = \sqrt{5}I_2$

To account for the fact that σ. can be negative, we re-define $m_i := \left\lfloor |\sigma_i|/\sqrt{\Delta x} \right\rfloor$ and $\gamma_i := (m_i + 1) - |\sigma_i|/\sqrt{\Delta x}$. The deterioration of the smoother for large m_i is present here too.

3.3 Performance of Geometric Multigrid

We conclude the discussion of geometric multigrid by testing its performance against an iterative solver used in [19], i.e. BICGSTAB [33] with and without ILU(0)[4] as preconditioner [29], and algebraic multigrid algorithms, namely, the classical Ruge-Stüben AMG [28] using our own implementation, and AGMG from [25], using the implementation from [23].

As benchmark examples, we choose a linear system $Ax = b$ whose coefficient matrix is the LISL discretization of (3.1) in the two-dimensional square $[0, 1]^2$ with Dirichlet boundary conditions, and $\sigma = 2I_2$ and $\sigma = \sqrt{5}I_2$, respectively, where I_2 is the 2×2 identity matrix. These values are chosen to study the effect of interpolation (which is always required in the second case and only for odd levels in the first) on the convergence and complexity of the methods, in particular on the convergence rates of geometric multigrid and the operator complexity of algebraic multigrid.

We use a Cartesian grid with equal number of equispaced nodes in both directions, the smoother is Gauss-Seidel, the prolongation operator bilinear interpolation, the restriction the transpose of the prolongation, and the coarse grid operator is constructed using the Galerkin principle.

Figure 5 presents the reduction of the residual, $r_k \equiv \|b - Ax^k\|_2$, against the number of iterations k with $x^0 = 0$, for a discrete mesh where the distance between two consecutive nodes is $\Delta x = 2^{-8}$. The algorithm is stopped whenever the relative residual, $\|b - Ax^k\|_2 / \|b\|_2$, measured in the Euclidean norm, is below the prescribed tolerance, in this case 10^{-6}.

Next, we study the residual reduction factor $\rho = (r_k/r_0)^{1/k}$, where k is the number of iterations required for the prescribed tolerance. We solve the problems above for different

[4] Incomplete LU factorization with the same sparsity pattern as the original system matrix.

Table 6 The residual reduction factor ρ for different mesh sizes and different multigrid algorithms, for the two-dimensional Laplace equation; the length of the stencil m as per (3.2)

ℓ	m	GMG V(1,1)	GMG W(1,1)	AMG	AGMG	BICGSTAB	BICGSTAB with ILU(0)
(a) ρ for $\sigma = 2I_2$							
6	16	0.4193	0.4204	0.0415	0.1015	0.2502	0.0151
7	22	0.2612	0.2666	0.0633	0.1502	0.5158	0.0767
8	32	0.7561	0.7564	0.0981	0.1551	0.5763	0.1234
9	45	0.5076	0.4905	0.1216	0.1858	0.7109	0.2392
10	64	0.8823	0.8841	0.1219	0.2001	0.7621	0.3382
(b) ρ for $\sigma = \sqrt{5}I_2$							
6	17	0.2999	0.2857	0.0403	0.1162	0.3718	0.0262
7	25	0.2565	0.2568	0.0532	0.1367	0.4408	0.0302
8	35	0.4348	0.4300	0.0740	0.1656	0.5938	0.1302
9	50	0.4480	0.4314	0.1124	0.2030	0.6751	0.1746
10	71	0.5547	0.4799	0.1272	0.1992	0.7478	0.3374

Table 7 Comparison of the residual reduction factor ρ for different system sizes and different solvers for the one dimensional Laplace equation. The system size is $2^\ell + 1$

ℓ	AGMG	BICGSTAB	BICGSTAB with ILU(0)
(a) ρ for $\sigma = 2$			
10	0.2486	0.6445	0
15	0.4680	0.9479	0.7105
20	0.5291	0.9815	0
21	0.6524	0.9935	0.9735
(b) ρ for $\sigma = \sqrt{5}$			
10	0.3298	0.7515	0.3079
15	0.5640	0.9312	0.7703
20	0.6347	0.9890	0.9550
21	0.4780	0.9940	0.9617

refinement levels ℓ, where the number of nodes per dimension is 2^ℓ +1. We observe in Table 6 that for even ℓ the convergence factor ρ corresponding to geometric multigrid cycles for $\sigma = 2I_2$ is significantly worse than that for $\sigma = \sqrt{5}I_2$. This is due to the lack of interpolation when $2^{\frac{\ell}{2}} \in \mathbb{N}$, as the step is a multiple of Δx, so for any mesh node $x_{\Delta x} \in \Omega_{\Delta x}$ $x_{\Delta x} \pm \sqrt{\Delta x}\sigma_i \in \Omega_{\Delta x}$. The lack of interpolation, and the equal stencil lengths in both directions, gives $\mu_{\mathrm{loc}} = 1$ in (3.5). Moreover, as shown in Fig. 3, the eigenvectors corresponding to small eigenvalues are highly oscillatory and hence not resolved sufficiently on the coarse mesh.

Regarding the BICGSTAB iterative solver, we observe the benefit of using ILU(0) as preconditioner, however, the significant increase in the convergence rate as the mesh is refined (and hence the condition number of the matrix increases) suggests that convergence is not asymptotically mesh size independent. To further illustrate this, Table 7 contains the residual reduction factors for AGMG and BICGSTAB with and without preconditioner when solving (3.1) in the one-dimensional domain [0, 1] with homogeneous Dirichlet boundary conditions and discretized using the LISL scheme. We consider the cases $\sigma = 2$ and $\sigma = \sqrt{5}$. As

Table 8 Comparison of the grid and algebraic complexities as per Definitions 3.1 and 3.2 for different mesh sizes and different multigrid algorithms, for the two-dimensional case

ℓ	GMG		AMG		AGMG	
	c_G	c_A	c_G	c_A	c_G	c_A
(a) Complexities for $\sigma = 2I_2$						
6	1.31	3.66	1.75	1.74	1.24	1.18
7	1.32	2.69	1.76	6.92	1.26	1.26
8	1.33	3.90	1.72	2.05	1.33	1.28
9	1.33	2.75	1.71	11.79	1.25	1.31
10	1.33	3.97	1.70	2.16	1.25	1.23
(b) Complexities for $\sigma = \sqrt{5}I_2$						
6	1.31	2.59	1.67	3.78	1.18	1.10
7	1.32	2.69	1.74	6.48	1.24	1.22
8	1.33	2.65	1.71	7.47	1.20	1.22
9	1.33	2.76	1.69	9.34	1.22	1.30
10	1.33	2.67	1.64	7.39	1.32	1.45

discussed previously, for $\sigma = 2$ and ℓ even, no interpolation is required. In this case, ILU(0) exactly factorises the system matrix A and hence $\rho = 0$. However, when interpolation is needed, ρ approaches 1 for BICGSTAB as the mesh is refined but not for AGMG. Furthermore, for $\ell = 21$ BICGSTAB and BICGSTAB with ILU(0) require approximately 86 and 21 times the time required by AGMG for the relative residual to be below 10^{-6}. Additionally, let t_{tol} be the time required for the relative residual to be below 10^{-6} and N the number of unknowns, assuming that $t_{tol} \sim \mathcal{O}(N^a)$, empirically we observe that a equals 1.11 for AGMG, 1.50 for BICGSTAB and 1.36 for BICGSTAB with ILU(0) as preconditioner.

Returning to the two-dimensional case, the grid hierarchies in the geometric (GMG) and algebraic (AMG) multigrid have 5 levels, including the finest one. In the geometric case, the number of unknowns is 4 times smaller from one level to the next. In the AGMG case, the hierarchy is at most 4 levels deep. Table 8 reports the complexities for the three categories of algorithms considered. The results confirm the assertion in [36] that AMG coarsening, generally, need not reduce the number of unknowns fast enough. In the present setting, contrasting the case $\sigma = \sqrt{5}I_2$ with $\sigma = 2I_2$ shows that the growth in complexity is due to the interpolation, which creates a denser connectivity graph on the coarser levels.

3.4 Properties of the LISL Matrix

In this section, we discuss the theoretical foundation of *Aggregation-based Multigrid* (AGMG) for our specific application of wide stencil discretisations.

The key result is Lemma 3.1 in [25]. The non-negativity of the row sums of a LISL discretization matrix is obtained almost by construction. To see this, let $A \in \mathbb{R}^{N\times N}$ be the discretization matrix, where $N := |\Omega_{\Delta x}|$ is the number of mesh points, then the sum for the i-th row is

$$\sum_{j=1}^{N}(A)_{ij} = 1 + \Delta t\left(\frac{M}{\Delta x} - c_i^{\alpha,n}\right) - \frac{\Delta t}{\Delta x}M \geq 0,$$

where we have used the fact that for any $z \in \bar{\Omega}$, $\sum_{j=1}^{N} w_j(z) = 1$ and the CFL-type condition $1 - \Delta t c_i^{\alpha,n} \geq 0$, which is satisfied for sufficiently small Δt independent of Δx.

The following analysis of the non-negativity of the column sum makes use of the regularity of the coefficients b and σ. In particular, we assume that the coefficients are such that for all $p \in [[1, P]]$, and for any mesh points x_i, x_l and corresponding controls $\alpha_i, \alpha_l \in \mathcal{A}$ and $s \in [0, T]$ we have that

$$\begin{aligned} \|\sigma_p^{\alpha_i}(s, x_i) - \sigma_p^{\alpha_l}(s, x_l)\|_\infty &\leq L_\sigma \|x_i - x_l\|_\infty^\eta, \\ \|b^{\alpha_i}(s, x_i) - b^{\alpha_l}(s, x_l)\|_\infty &\leq L_b \|x_i - x_l\|_\infty^\beta, \end{aligned} \tag{3.6}$$

where $\beta \in (0, 1]$, $\eta \in \left(\frac{1}{2}, 1\right]$.

Remark 3.1 As stated in the introduction, we are working under the standard assumption of Lipschitz continuity of the coefficients in x and continuity in α. However, what we require in (3.6) is stronger, namely, if the control is inserted in the coefficients as a function of the state x, the resulting functions are Hölder continuous in x. This situation arises in every step of the policy iteration algorithm: a control vector (α_i) is determined by the optimisation step, and then a linear system with this control vector is solved for $(x)_i$. Generally, the optimal control is not a (Hölder) continuous function of the space variables, but there are many important examples where it is at least piecewise Hölder. It can be seen from the proof below that Proposition 3.1 still holds in this situation.

Remark 3.2 Lemma 3.1 in [25] also assumes that the system matrix is irreducible. LISL discretization matrices need not be irreducible, e.g. $L_{SL}^{2K,2,1}$ for any $K \in \mathbb{N}$ as in (3.3), however, this technical requirement could be overcome by adding an irreducible M-matrix, multiplied by a sufficiently small factor, to the LISL discretization matrix.

We also assume the use of multi-linear interpolation requiring 2^d points to approximate function values in $\mathbb{R}^d$ and the use of Cartesian grids.

Proposition 3.1 *Let A be the LISL discretization matrix of* (1.1) *for a given time step, on an equispaced Cartesian grid $\Omega_{\Delta x}$ of $\Omega \subset \mathbb{R}^d$ with $\Delta x > 0$, and a given vector of control values $(\alpha_i)_{i=1,\ldots,N}$, $\alpha_i \in \mathcal{A}$, associated with the mesh points x_i, $1 \leq i \leq N := |\Omega_{\Delta x}|$. Assume that* (3.6) *holds.*

Then the column sum of the matrix is non-negative provided

$$\Delta t \leq \frac{\Delta x}{\sup_{\alpha \in \mathcal{A}} |c^{\alpha,+}| + (\mathcal{M} - 1)(P + 1)}, \tag{3.7}$$

where $\mathcal{M}$ depends on the dimension of the domain d and the Lipschitz constants L_σ and L_b, but not on the mesh parameter Δx. Indeed, $\mathcal{M} = 3^d$ for sufficiently small Δx.

Proof We carry out the proof for Scheme 2, but an analogous analysis holds for Schemes 1 and 3 in the introduction. We also note that we can restrict the analysis to steps where no truncation is required, as in the case of truncation the weights only contribute (positively) to the diagonal of the matrix and the right-hand side of the equation (see Remark 2.2).

For simplicity of notation, we omit the dependence of the coefficient functions b and σ_p on the time variable t and the control. For any $i \neq j$ the matrix entry $(A)_{ij} \neq 0$ if and only if for any $1 \leq m \leq P + 1$ we require $\phi(x_j)$ to approximate – by means of linear interpolation – $\phi(x_i + y_m^{\pm}(x_i))$, where $y_m^{\pm}(x_i)$ is either $y_p^{\pm}(x_i) = \pm\sqrt{\Delta x}\sigma_p(x_i)$ for $1 \leq p \leq P$ or $y_{P+1}^{\pm}(x_i) = \Delta x b(x_i)$. For any two nodes i and l to contribute to the sum of column j, it is necessary that

$$\|x_l + y_m^{\pm}(x_l) - (x_i + y_m^{\pm}(x_i))\|_\infty < 2\Delta x.$$

As x_l and x_i lie on the grid, there exists a positive constant M such that $\|x_l - x_i\|_\infty = M\Delta x$. Then $(M+1)^d$ constitutes an upper bound on the number of terms the step $y_m^{\pm}(\cdot)$ contributes to the sum of column j.

We consider the different possible values for $y_m^{\pm}(x_l)$ separately. First, assume $y_m^{\pm}(x_l) = \Delta x b(x_l)$ and let $\Delta x \leq 1/(L_b\sqrt{d})$, then

$$2\Delta x > \|x_l + \Delta x b(x_l) - (x_i + \Delta x b(x_i))\|_\infty \geq M\Delta x - \Delta x^{1+\beta} L_b M^\beta, \tag{3.8}$$

where we have used the triangle inequality and the Hölder regularity of b. Re-arranging,

$$M < \frac{2}{1 - L_b M^{\beta-1} \Delta x^\beta},$$

such that $M \leq 2$ for sufficiently small Δx. Proceeding similarly for $y_m^{\pm}(x) = \pm\sqrt{\Delta x}\sigma_m(x)$, we obtain again $M \leq 2$ as $\Delta x \to 0$.

Denote $\mathcal{M}$ to be the maximum of all of the $(M+1)^d$s above, i.e. for different mesh points, then the column sum gives

$$\begin{aligned}\sum_{i=1}^{N}(A)_{ij} &= 1 + \Delta t\left(\frac{P+1}{\Delta x} - c_j^{\alpha,n}\right) - \frac{\Delta t}{2\Delta x}\sum_{i=1}^{N}\sum_{p=1}^{P+1} w_j(x_i + y_p(x_i))\\ &\geq 1 - \frac{\Delta t}{\Delta x}\left(c_j^{\alpha,n} + (\mathcal{M}-1)(P+1)\right).\end{aligned} \tag{3.9}$$

Therefore, non-negativity of the sum is guaranteed by condition (3.7). □

Remark 3.3 For the LISL scheme to be first order accurate, it is required that $\Delta t \sim \mathcal{O}(\Delta x)$. Therefore, the bound (3.7) does not impose problematic restrictions on the size of the time steps. We recall that $\Delta t = \mathcal{O}(\Delta x)$ and $\Delta t = \mathcal{O}(\Delta x^{3/2})$ (or even $\Delta t = \mathcal{O}(\Delta x^2)$) are the CFL conditions for the explicit schemes without and with truncation, respectively, see Corollary 2.5. Therefore, on bounded domains, fully implicit time stepping with policy iteration and AGMG preconditioning is the computationally most efficient overall algorithm among the ones considered.

3.5 Performance of the Algebraic Approaches

We compare the performance of the classical AMG implementation in the HSL library [15], and AGMG from [23] for the benchmark optimal control problems in Sect. 2.4. Both of these methods are used as preconditioners for a Krylov subspace method that is assumed to have converged when the relative residual is below 10^{-6}. In particular, we use MATLAB's implementation of GMRES [30] for AMG and GCR [10] for AGMG. The AMG preconditioner consists of one iteration of the standard V-cycle with two Gauss-Seidel pre- and post-smoothing steps, whereas AGMG uses one Gauss-Seidel pre- and post-smoothing step and the enhanced multigrid cycles mentioned in the introduction, see [24,25]. For completeness, we also include as benchmark MATLAB's sparse direct solver using UMFPACK [8]. The problems considered have smooth closed form solutions linear in t. As mentioned in the previous section, we employ policy iteration to solve the resulting non-linear discrete problem. The tests were run on a Linux machine under MATLAB 2015a, on a quad-core AMD 4.2GHz with 7.5GB of RAM and 15GB of swap.

In Fig. 6 we present the elapsed time solving linear systems for a single time step ($\Delta t = T$). Both MG methods provide a solution with the same accuracy as the sparse direct solver

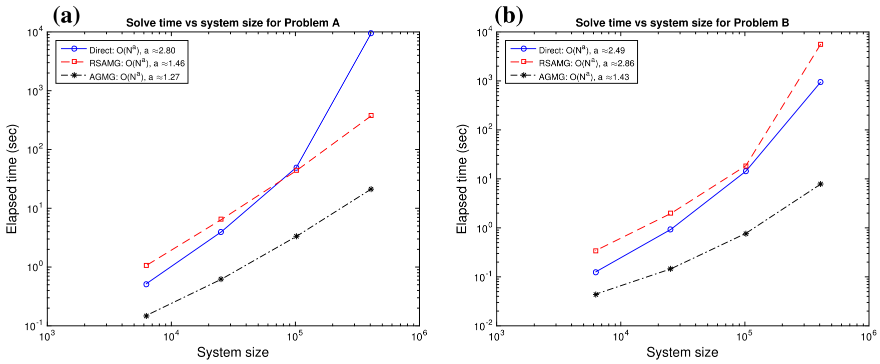

Fig. 6 Total number of seconds for solving the linear systems versus the size of the systems for each of the linear system solvers considered. We use equispaced Cartesian grids in space with 81, 161, 321 and 641 nodes per dimension and one time step. **a** Total time on solver for Problem A. **b** Total time on solver for Problem B

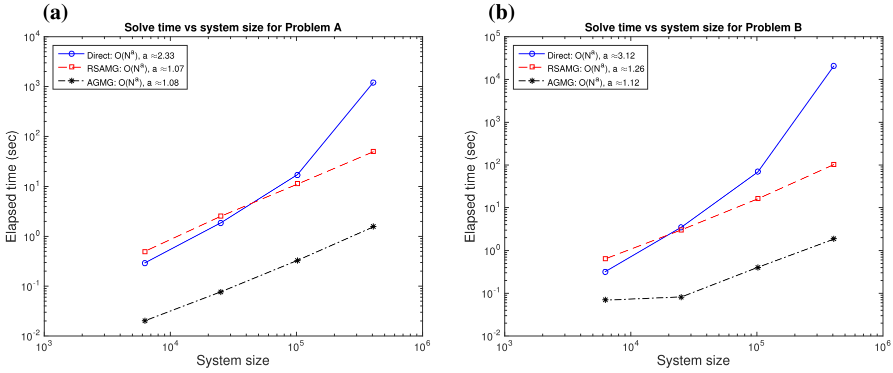

Fig. 7 Average number of seconds per time step for solving the linear systems versus the size of the systems. We use equispaced Cartesian grids in space with 81, 161, 321 and 641 nodes per dimension and $\Delta t = \Delta x$. **a** Average time on solver for Problem A. **b** Average time on solver for Problem B

but with improved scalability. AGMG outperforms AMG and the sparse direct solver in both problems. Figure 7 shows the average time spent solving linear systems per time step when $\Delta t = \Delta x$. Reducing Δt makes the system matrix more diagonally dominant and as a consequence easier to precondition. This effect is noticeable for Problem B using AMG as preconditioner, see Table 9.

Table 10 and Table 11 report memory consumption and quantities related to the Krylov subspace method and to the coarsening. As commented in the previous sections, AMG results in grid and algebraic complexities higher than AGMG's. The coarsening for both of the methods is stopped when the coarse level system is cheap to solve exactly compared to the starting system, specifically, we stop whenever the number of unknowns at the coarse level is comparable to the cubic root of the initial number of unknowns. The effect of simplifying the intergrid transfer operators can be observed on the coarse to fine stencil ratio (C/F stencil). For AGMG, the stencil on the coarsest level is similar to the initial one, whereas for AMG it is significantly denser. The fact that aggregation-based coarsening strategies yield coarse matrices with similar sparsity as the original one

Table 9 Average seconds per time step solving linear systems

N_x	Direct		AMG		AGMG	
	$\Delta t = T$	$\Delta t = \Delta x$	$\Delta t = T$	$\Delta t = \Delta x$	$\Delta t = T$	$\Delta t = \Delta x$
Problem A						
321	49.60	17.93	43.62	10.07	3.31	0.41
641	9.50e+03	4.33e+03	373.77	48.82	21.22	1.84
Problem B						
321	14.35	68.70	18.59	16.19	0.77	0.40
641	950.62	2.09e+04	5.64e+03	102.57	7.82	1.85

Table 10 Peak memory consumption statistics in gigabytes (GB) of the MATLAB process sampled using the shell command `top`

N_x	Direct		AMG		AGMG	
	VIRT	RES	VIRT	RES	VIRT	RES
(a) Peak memory consumption measured in GB for $\Delta t = T$						
Problem A						
321	11.40	5.29	10.63	5.10	9.83	4.14
641	25.99	7.13	14.40	7.05	12.66	6.76
Problem B						
321	11.71	6.50	9.84	5.14	9.84	5.15
641	23.83	7.10	18.51	7.13	9.90	5.12
(b) Peak memory consumption measured in GB for $\Delta t = \Delta x$						
Problem A						
321	8.39	3.20	6.41	2.38	6.48	2.39
641	21.80	7.22	10.61	6.68	10.52	6.64
Problem B						
321	13.76	7.09	9.83	3.67	9.83	3.71
641	26.11	7.26	12.72	7.23	9.90	5.17

VIRT is the total amount of virtual memory used by MATLAB, whereas RES is the non-swapped physical memory (limited to 7.5)

was noted in [16]. Moreover, AGMG yields shallower hierarchies due to higher coarsening factors. The effect of reducing Δt is also appreciated in this ratio. We observe that the direct method's consumption increases dramatically while for the MG methods, we note the relation between the memory requirement and the algebraic complexity of the method.

The number of Krylov iterations highlight previous comments on the fact that aggregation-based multigrid methods are not efficient if used as stand-alone solvers: in all test cases, AGMG used more iterations than AMG per policy iteration. However, AGMG used as a preconditioner to a Krylov subspace method provides accurate solutions faster and cheaper than the other two solvers considered.

Table 11 Quantities related to the Krylov subspace iteration and multigrid coarsening

Solver	N_x	Avg Krylov It	# levels	C/F stencil	c_G	c_A
(a) Krylov iterations and coarsening related quantities for $\Delta t = T$						
Problem A						
AMG	321	4.00	9.0	26.25	2.61	7.06*
	641	4.29	11.0	22.33	2.42	4.63*
AGMG	321	12.67	5.83	1.30	1.81	1.97
	641	17.14	6.14	1.36	1.61	1.77
Problem B						
AMG	321	5.00	7.0	86.26	2.08	9.60*
	641	6.67	9.5	221.43	2.23	11.61*
AGMG	321	12.00	5.00	0.50	1.60	1.92
	641	15.00	5.50	0.39	1.53	1.57
(b) Krylov iterations and coarsening related quantities for $\Delta t = \Delta x$						
Problem A						
AMG	321	3.00	9.98	16.30	2.76	5.90*
	641	3.00	12.0	16.60	2.75	5.12*
AGMG	321	6.00	2.00	0.23	1.00	1.00
	641	6.00	2.00	0.26	1.00	1.00
Problem B						
AMG	321	2.98	8.61	48.65	2.47	8.61*
	641	2.99	11.27	107.77	2.70	11.12*
AGMG	321	4.99	2.96	0.31	1.07	1.02
	641	5.01	2.97	0.33	1.15	1.10

Avg Krylov It contains the average number of Krylov iterations over all time steps and all policy iterations; *# levels* contains the average depth in the grid hierarchy; *C/F stencil* contains the ratio between the stencil at the coarsest level and that on the finest level (lower is better). On the finest level, the stencil is close to 11 for Problem A and close to 8 for Problem B. The last two columns report the grid and algebraic complexity as per Definitions 3.1 and 3.2. As the full matrix hierarchy was not available from [15] for AMG, but only the coarsest and finest matrices, the starred algebraic complexities are estimates based on the assumption of a geometrically decreasing complexity between the coarsest and finest level, which is likely to be a significant underestimate

4 Conclusions

This article discusses two aspects of practical importance associated with wide stencil discretizations of second order non-linear parabolic differential operators. First, we study the truncation of the stencil for problems on bounded domains, as a result of the method overstepping the boundaries for nodes in a surrounding layer. Our main result details the construction of such truncation and proves that the resulting scheme remains consistent, monotone and conditionally stable. Numerical examples confirm that the truncation improves the accuracy of the approach compared to constant and linear extrapolation of the boundary conditions, and the modification of the CFL condition of the scheme.

Second, motivated by the stringent CFL condition of explicit time stepping schemes, we consider implicit schemes and the application of multigrid methods to solve the resulting discrete non-linear system of equations efficiently. Using theoretical and empirical arguments, we show the need to employ multigrid methods based on algebraic ideas. We show that aggregation-based methods are well suited for the discretization schemes considered and justify their use by proving that under mild conditions on the mesh refinement parameters the LISL discretization matrices are M-matrices with non-negative row and column sums. The algorithms are shown to compare favourably against AMG and sparse direct solvers.

Although we only considered linear interpolation, much of the analysis, including in particular the matrix properties in Sect. 3.4, will also hold if other limited interpolations (see, e.g., [9,35]), are used, as only the properties in (2.10) are critical.

To conclude, we emphasise that monotone schemes for general diffusions in two and more dimensions are necessarily non-local, so that the question of boundary truncation is not restricted to the class of schemes studied in this paper.

Acknowledgements Julen Rotaetxe Arto was partially supported by the *Programa de Formaci?n de Investigadores del DEUI del Gobierno Vasco*.

5 Appendix

See Tables 12, 13, 14, 15, and 16.

Table 12 Results using stencil truncation for explicit method with $N_\alpha = 40$ for Problem B

N_x	$\Delta t = T$		$\Delta t = \frac{\Delta x}{4}$		$\Delta t = \Delta x^{\frac{3}{2}}$		$\Delta t = \Delta x^2$	
	Error	Rate	Error	Rate	Error	Rate	Error	Rate
(a) Error in L^∞-norm over $\Omega_{\Delta x}$								
41	1.73e−01	–	3.91e−02	–	3.95e−02	–	3.88e−02	–
81	1.39e−01	0.32	1.84e−02	1.09	1.84e−02	1.10	1.83e−02	1.09
161	1.07e−01	0.38	8.71e−03	1.08	8.70e-03	1.08	8.68e−03	1.07
321	8.05e−02	0.41	1.39e+43	−150.16	4.12e−03	1.08	4.11e−03	1.08
641	5.95e−02	0.44	1.77e+153	−365.76	2.17e−03	0.92	2.17e−03	0.92
(b) Error in L^∞-norm over $\Omega_{\Delta x} \cap [-\pi/2, \pi/2]^2$								
41	5.71e−02	–	3.91e−02	–	3.95e−02	–	3.88e−02	–
81	2.74e−02	1.06	1.84e−02	1.09	1.84e−02	1.10	1.83e−02	1.09
161	1.31e−02	1.06	8.71e−03	1.08	8.70e−03	1.08	8.68e−03	1.07
321	6.57e−03	0.99	8.34e+28	−102.92	4.12e−03	1.08	4.11e−03	1.08
641	3.28e−03	1.00	1.09e+127	−325.93	2.17e−03	0.92	2.17e−03	0.92

Table 13 Results using constant extrapolation of the boundary condition for explicit method with $N_\alpha = 40$ for Problem B

N_x	$\Delta t = T$		$\Delta t = \frac{\Delta x}{4}$		$\Delta t = \Delta x^{\frac{3}{2}}$		$\Delta t = \Delta x^2$	
	Error	Rate	Error	Rate	Error	Rate	Error	rate
(a) Error in L^∞-norm over $\Omega_{\Delta x}$								
41	1.25e+00	–	3.79e−01	–	3.82e−01	–	3.75e−01	–
81	1.99e+00	-0.67	3.55e−01	0.09	3.55e−01	0.11	3.53e−01	0.09
161	3.04e+00	-0.61	2.92e−01	0.28	2.92e−01	0.28	2.92e−01	0.27
321	4.52e+00	-0.57	2.35e−01	0.32	2.35e−01	0.32	2.35e−01	0.31
641	6.62e+00	-0.55	1.77e−01	0.41	1.77e−01	0.41	1.77e−01	0.41
(b) Error in L^∞-norm over $\Omega_{\Delta x} \cap [-\pi/2, \pi/2]^2$								
41	5.71e−02	–	6.38e−02	–	6.34e−02	–	6.40e−02	–
81	2.74e−02	1.06	5.72e−02	0.16	5.72e−02	0.15	5.68e−02	0.17
161	1.31e−02	1.06	4.51e−02	0.34	4.51e−02	0.34	4.49e−02	0.34
321	6.57e−03	0.99	3.71e−02	0.28	3.71e−02	0.28	3.70e−02	0.28
641	3.28e−03	1.00	2.89e−02	0.36	2.89e−02	0.36	2.88e−02	0.36

Table 14 Results using linear extrapolation for points out of the domain for explicit method with $N_\alpha = 40$ for Problem B

N_x	$\Delta t = T$		$\Delta t = \frac{\Delta x}{4}$		$\Delta t = \Delta x^{\frac{3}{2}}$		$\Delta t = \Delta x^2$	
	Error	Rate	Error	Rate	Error	Rate	Error	Rate
(a) Error in L^∞-norm over $\Omega_{\Delta x}$								
41	5.71e−02	–	8.46e−02	–	8.29e−02	–	8.60e−02	–
81	3.12e−02	0.87	2.43e+02	−11.49	1.82e+02	−11.10	1.67e+03	−14.25
161	2.89e−02	0.11	7.90e+18	−54.85	8.95e+20	−62.10	1.64e+31	−92.99
321	2.38e−02	0.28	1.51e+70	−170.36	9.26e+93	−242.55	1.51e+164	−441.69
641	1.87e−02	0.35	1.14e+207	−454.70	NaN	NaN	NaN	NaN
(b) Error in L^∞-norm over $\Omega_{\Delta x} \cap [-\pi/2, \pi/2]^2$								
41	5.71e−02	–	3.91e−02	–	3.95e−02	–	3.88e−02	–
81	2.74e−02	1.06	1.84e−02	1.09	1.84e−02	1.10	5.19e−02	−0.42
161	1.31e−02	1.06	1.36e+09	−36.10	1.54e+11	−42.92	2.82e+21	−75.52
321	6.57e−03	0.99	5.30e+52	−144.81	3.24e+76	−217.00	5.27e+146	−416.15
641	3.28e−03	1.00	6.39e+176	−412.19	NaN	NaN	NaN	NaN

Table 15 Results using truncation for points out of the domain for implicit method with $N_\alpha = 40$ for Problem B

N_x	$\Delta t = T$		$\Delta t = \frac{\Delta x}{4}$		$\Delta t = \Delta x^{\frac{3}{2}}$		$\Delta t = \Delta x^2$	
	Error	Rate	Error	Rate	Error	Rate	Error	Rate
(a) Error in L^∞-norm over $\Omega_{\Delta x}$								
41	3.00e−02	–	3.76e−02	–	3.72e−02	–	3.79e−02	–
81	1.40e−02	1.10	1.80e−02	1.06	1.80e−02	1.05	1.45e−02	1.38
161	6.34e−03	1.15	6.36e−03	1.50	6.37e−03	1.50	7.72e−03	0.91
321	3.04e−03	1.06	3.38e−03	0.91	3.50e−03	0.86	3.01e−03	1.36
641	1.53e−03	0.99	1.77e−03	0.93	1.76e−03	1.00	1.66e−03	0.85
(b) Error in L^∞-norm over $\Omega_{\Delta x} \cap [-\pi/2, \pi/2]^2$								
41	3.00e−02	–	3.76e−02	–	3.72e−02	–	3.79e−02	–
81	1.40e−02	1.10	1.80e−02	1.06	1.80e−02	1.05	1.45e−02	1.38
161	6.34e−03	1.15	6.31e−03	1.51	6.37e−03	1.50	7.72e−03	0.91
321	3.04e−03	1.06	3.38e−03	0.90	3.50e−03	0.86	3.01e−03	1.36
641	1.53e−03	0.99	1.77e−03	0.93	1.76e−03	1.00	1.66e−03	0.85

Table 16 Percentage of computational time spent in linear solvers for the Examples in Sect. 3.5

N_x	Direct		AMG		AGMG	
	$\Delta t = T(\%)$	$\Delta t = \Delta x(\%)$	$\Delta t = T(\%)$	$\Delta t = \Delta x(\%)$	$\Delta t = T(\%)$	$\Delta t = \Delta x(\%)$
Problem A						
161	44.17	69.93	64.21	75.42	12.40	7.63
321	53.34	82.36	69.24	77.24	15.09	10.82
641	78.01	85.91	34.20	67.47	5.53	10.37
Problem B						
161	8.46	80.71	49.16	79.35	7.27	8.92
321	26.09	94.36	77.48	76.28	12.88	8.83
641	95.25	97.65	98.64	87.68	2.48	17.06

References

1. Akian, M., Séquier, P., Sulem, A.: A finite horizon multidimensional portfolio selection problem with singular transactions. In: Proceedings of the 34th IEEE Conference on Decision and Control, 1995, vol 3, pp. 2193–2198. IEEE (1995)
2. Barles, G., Souganidis, P.E.: Convergence of approximation schemes for fully nonlinear second order equations. Asymptot. Anal. **4**(3), 271–283 (1991)
3. Bloß, M., Hoppe, R.H.W.: Numerical computation of the value function of optimally controlled stochastic switching processes by multi-grid techniques. Numer. Funct. Anal. Optim. **10**(3–4), 275–304 (1989)
4. Bokanowski, O., Maroso, S., Zidani, H.: Some convergence results for Howard's algorithm. SIAM J. Numer. Anal. **47**(4), 3001–3026 (2009)
5. Camilli, F., Falcone, M.: An approximation scheme for the optimal control of diffusion processes. ESAIM Math. Model. Numer. Anal. Modélisation Mathématique et Analyse Numérique **29**(1), 97–122 (1995)
6. Crandall, G.M., Lions, P.-L.: Convergent difference schemes for nonlinear parabolic equations and mean curvature motion. Numerische Mathematik **75**(1), 17–41 (1996)

7. Crandall, M.G., Ishii, H., Lions, P.-L.: User's guide to viscosity solutions of second order partial differential equations. Bull. Am. Math. Soc. **27**(1), 1–67 (1992)
8. Davis, T.A.: Algorithm 832: UMFPACK V4.3–+an unsymmetric-pattern multifrontal method. ACM Trans. Math. Softw. **30**(2), 196–199 (2004)
9. Debrabant, K., Jakobsen, E.R.: Semi-Lagrangian schemes for linear and fully non-linear diffusion equations. Math. Comput. **82**(283), 1433–1462 (2013)
10. Eisenstat, S.C., Elman, H.C., Schultz, M.H.: Variational iterative methods for nonsymmetric systems of linear equations. SIAM J. Numer. Anal. **20**(2), 345–357 (1983)
11. Forsyth, P.A., Labahn, G.: Numerical methods for controlled Hamilton–Jacobi–Bellman PDEs in finance. J. Comput. Finance **11**(2), 1 (2007)
12. Forsyth, P.A., Vetzal, K.R.: Numerical methods for nonlinear PDEs in finance. In: Duan, J.-C., Härdle, W.K., Gentle, J.E. (eds.) Handbook of computational finance. Springer Handbooks of Computational Statistics, pp. 503–528. Springer, Berlin (2012)
13. Han, D., Wan, J.W.L.: Multigrid methods for second order Hamilton–Jacobi–Bellman and Hamilton–Jacobi–Bellman–Isaacs equations. SIAM J. Sci. Comput. **35**(5), S323–S344 (2013)
14. Hoppe, R.: Multi-grid methods for Hamilton–Jacobi–Bellman equations. Numerische Mathematik **49**(2–3), 239–254 (1986)
15. HSL: A collection of Fortran codes for large scale scientific computation. http://www.hsl.rl.ac.uk/ (2015)
16. Kim, H., Xu, J., Zikatanov, L.: A multigrid method based on graph matching for convection-diffusion equations. Numer. Linear Algebra Appl. **10**(1–2), 181–195 (2003)
17. Kushner, H.J., Dupuis, P.: Numerical Methods for Stochastic Control Problems in Continuous Time, vol. 24. Springer, Berlin (2001)
18. Lions, P.-L.: Optimal control of diffusion processes and Hamilton–Jacobi–Bellman equations part 2: viscosity solutions and uniqueness. Commun. Partial Differ. Eq. **8**(11), 1229–1276 (1983)
19. Ma, K., Forsyth, P.A.: An unconditionally monotone numerical scheme for the two factor uncertain volatility model. IMA J. Numer. Anal. (2016). doi:10.1093/imanum/drw025
20. Menaldi, J.-L.: Some estimates for finite difference approximations. SIAM J. Control Optim. **27**(3), 579–607 (1989)
21. Motzkin, T.S., Wasow, W.: On the approximation of linear elliptic differential equations by difference equations with positive coefficients. J. Math. Phys. **31**(1), 253–259 (1952)
22. Napov, A., Notay, Y.: An algebraic multigrid method with guaranteed convergence rate. SIAM J. Sci. Comput. **34**(2), A1079–A1109 (2012)
23. Notay, Y.: Homepage for AGMG. http://homepages.ulb.ac.be/~ynotay/AGMG/
24. Notay, Y.: An aggregation-based algebraic multigrid method. Electron. Trans. Numer. Anal. **37**(6), 123–146 (2010)
25. Notay, Y.: Aggregation-based algebraic multigrid for convection-diffusion equations. SIAM J. Sci. Comput. **34**(4), A2288–A2316 (2012)
26. Notay, Y., Vassilevski, P.S.: Recursive Krylov-based multigrid cycles. Numer. Linear Algebra Appl. **15**(5), 473–487 (2008)
27. 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)
28. Ruge, J., Stüben, K.: Algebraic multigrid. Multigrid Methods **3**, 73–130 (1987)
29. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia (2003)
30. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **7**(3), 856–869 (1986)
31. Stüben, K.: A review of algebraic multigrid. J. Comput. Appl. Math., **128**(1-2), 281–309, 2001. Numerical Analysis 2000. Vol. VII: Partial Differential Equations
32. Trottenberg, U., Oosterlee, C., Schüller, A.: Multigrid. Academic Press, Cambridge (2001)
33. van der Vorst, H.A.: Bi-CGSTAB: a fast and smoothly converging variant of Bi–CG for the solution of nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **13**(2), 631–644 (1992)
34. Vassilevski, P.S.: Multilevel Block Factorization Preconditioners: Matrix-Based Analysis and Algorithms for Solving Finite Element Equations. Springer, Berlin (2008)
35. Warin, X.: Some non-monotone schemes for time dependent Hamilton-Jacobi-Bellman equations in stochastic control. J. Sci. Comput. **66**, 1122–1147 (2013)
36. Wathen, A.J.: Preconditioning. Acta Numer. **24**, 329–376 (2015)