TU Delft

Delft University of Technology

Classification and image processing with a semi-discrete scheme for fidelity forced Allen–Cahn on graphs

Budd, Jeremy; van Gennip, Yves; Latz, Jonas

DOI
10.1002/gamm.202100004

Publication date
2021

Document Version
Final published version

Published in
GAMM Mitteilungen

Citation (APA)
Budd, J., van Gennip, Y., & Latz, J. (2021). Classification and image processing with a semi-discrete scheme for fidelity forced Allen–Cahn on graphs. *GAMM Mitteilungen*, *44*(1), 1-43. Article e202100004. https://doi.org/10.1002/gamm.202100004

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Check for updates

DOI: 10.1002/gamm.202100004

ORIGINAL PAPER

GAMM Mitteilungen

Classification and image processing with a semi-discrete scheme for fidelity forced Allen–Cahn on graphs

Jeremy Budd[1] | **Yves van Gennip**[1] | **Jonas Latz**[2]

[1]Delft Institute of Applied Mathematics (DIAM), Technische Universiteit Delft, Delft, The Netherlands

[2]Department of Applied Mathematics and Theoretical Physics (DAMTP), University of Cambridge, Cambridge, UK

Correspondence
Jeremy Budd, Mathematics and Computer Science, EWI, TU Delft, Van Mourik Broekmanweg 6, Delft 2628 XE, The Netherlands.
Email: j.m.budd-1@tudelft.nl

Abstract

This paper introduces a semi-discrete implicit Euler (SDIE) scheme for the Allen-Cahn equation (ACE) with fidelity forcing on graphs. The continuous-in-time version of this differential equation was pioneered by Bertozzi and Flenner in 2012 as a method for graph classification problems, such as semi-supervised learning and image segmentation. In 2013, Merkurjev et. al. used a Merriman-Bence-Osher (MBO) scheme with fidelity forcing instead, as heuristically it was expected to give similar results to the ACE. The current paper rigorously establishes the graph MBO scheme with fidelity forcing as a special case of an SDIE scheme for the graph ACE with fidelity forcing. This connection requires the use of the double-obstacle potential in the ACE, as was already demonstrated by Budd and Van Gennip in 2020 in the context of ACE without a fidelity forcing term. We also prove that solutions of the SDIE scheme converge to solutions of the graph ACE with fidelity forcing as the discrete time step converges to zero. In the second part of the paper we develop the SDIE scheme as a classification algorithm. We also introduce some innovations into the algorithms for the SDIE and MBO schemes. For large graphs, we use a QR decomposition method to compute an eigendecomposition from a Nyström extension, which outperforms the method used by, for example, Bertozzi and Flenner in 2012, in accuracy, stability, and speed. Moreover, we replace the Euler discretization for the scheme's diffusion step by a computation based on the Strang formula for matrix exponentials. We apply this algorithm to a number of image segmentation problems, and compare the performance with that of the graph MBO scheme with fidelity forcing. We find that while the general SDIE scheme does not perform better than the MBO special case at this task, our other innovations lead to a significantly better segmentation than that from previous literature. We also empirically quantify the uncertainty that this segmentation inherits from the randomness in the Nyström extension.

KEYWORDS

Allen-Cahn equation, fidelity constraint, graph dynamics, Nyström extension, Strang formula, threshold dynamics

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
© 2021 The Authors. *GAMM - Mitteilungen* published by Wiley-VCH GmbH.

1 | INTRODUCTION

In this paper, we investigate the Allen-Cahn gradient flow of the Ginzburg-Landau functional on a graph, and the Merriman-Bence-Osher (MBO) scheme on a graph, with fidelity forcing. We extend the definition of the semi-discrete implicit Euler (SDIE) scheme (introduced in [13] for the graph Allen-Cahn equation (ACE)) to the case of fidelity forcing, and prove that the key results of [13] hold true in the fidelity forced setting, that is

- the MBO scheme with fidelity forcing is a special case of the SDIE scheme with fidelity forcing; and
- the SDIE solution converges to the solution of Allen-Cahn with fidelity forcing as the SDIE time step tends to zero.

We then demonstrate how to employ the SDIE scheme as a classification algorithm, making a number of improvements upon the MBO-based classification in [32]. In particular, we have developed a stable method for extracting an eigendecomposition or singular value decomposition (SVD) from the Nyström extension [21, 36] that is both faster and more accurate than the previous method used in [6, 32]. Finally, we test the performance of this scheme as an alternative to graph MBO as a method for image processing on the "two cows" segmentation task considered in [6, 32].

Given an edge-weighted graph, the goal of two-class graph classification is to partition the vertex set into two subsets in such a way that the total weight of edges within each subset is high and the weight of edges between the two subsets is low. Classification differs from clustering by the addition of some a priori knowledge, that is, for certain vertices the correct classification is known beforehand. Graph classification has many applications, such as semi-supervised learning and image segmentation [6, 15].

All programming for this paper was done in MATLAB R2019a. Except within algorithm environments and URLs, all uses of `typewriter font` indicate built-in MATLAB functions.

1.1 | Contributions of this work

In this paper we have:

- Defined a double-obstacle ACE with fidelity forcing (Definition 2.6), and extended the theory of [13] to this equation (Theorem 2.7).
- Defined an SDIE scheme for this ACE (Definition 2.8) and following [13] proved that this scheme is a generalization of the fidelity forced MBO scheme (Theorem 2.9), derived a Lyapunov functional for the SDIE scheme (Theorem 2.12), and proved that the scheme converges to the ACE solution as the time-step tends to zero (Theorem 2.17).
- Described how to employ the SDIE scheme as a generalization of the MBO-based classification algorithm in [32].
- Developed a method, inspired by [3], using the QR decomposition to extract an approximate SVD of the normalized graph Laplacian from the Nyström extension (Algorithm 1), which avoids the potential for errors in the method from [6, 32] that can arise from taking the square root of a non-positive-semi-definite matrix, and empirically produces much better performance than the [6, 32] method (Figure 4) in accuracy, stability, and speed.
- Developed a method using the quadratic error Strang formula for matrix exponentials [39] for computing fidelity forced graph diffusion (Algorithm 2), which empirically incurs a lower error than the error incurred by the semi-implicit Euler method used in [32] (Figure 6), and explored other techniques with the potential to further reduce error (Table 1).
- Demonstrated the application of these algorithms to image segmentation, particularly the "two cows" images from [6, 32], compared the quality of the segmentation to those produced in [6, 32] (Figure 11), and investigated the uncertainty in these segmentations (Figure 14), which is inherited from the randomization in Nyström.

This work extends the work in [13] in four key ways. First, introducing fidelity forcing changes the character of the dynamics, for example, making graph diffusion affine, which changes a number of results/proofs, and it is thus of interest that the SDIE link continues to hold between the MBO scheme and the ACE. Second, this work for the first time considers the SDIE scheme as a tool for applications. Third, in developing the scheme for applications we have made a number of improvements to the methods used in the previous literature [32] for MBO-based classification, which result in a better segmentation of the "two cows" image than that produced in [32] or [6]. Fourth, we quantify the randomness that the segmentation inherits from the Nyström extension.

TABLE 1 Comparison of the relative ℓ^2 errors from the methods for approximating b on the image from Figure 5

Method	Relative ℓ^2 error for $\tau = 0.5$			Relative ℓ^2 error for $\tau = 4$		
	$\lvert V\rvert$=1600, $K = 1600$	$\lvert V\rvert$=1600, $K = 40$	$\lvert V\rvert$=6400, $K = 40$	$\lvert V\rvert$=1600, $K = 1600$	$\lvert V\rvert$=1600, $K = 40$	$\lvert V\rvert$=6400, $K = 40$
Semi-implicit Euler [32]	1.434×10^{-4}	0.4951	0.4111	2.882×10^{-4}	0.2071	0.1721
Woodbury identity	2.292×10^{-8}	0.5751	0.4607	1.038×10^{-7}	**0.1973**	**0.1537**
Midpoint rule (3.3)	0.0279	**0.1290**	**0.1110**	0.4279	0.6083	0.6113
Simpson's rule ($m = 500$) via Strang formula	2.592×10^{-9}	0.1335	0.1136	5.845×10^{-7}	0.5124	0.4827
MATLAB `integrate` via Strang formula	n/a	0.1335	0.1136	n/a	0.5124	0.4827
Simpson's rule ($m = 500$) via Yoshida method	$\mathbf{8.381 \times 10^{-14}}$	0.1335	0.1136	$\mathbf{9.335 \times 10^{-12}}$	0.5124	0.4827
MATLAB `integrate` via Yoshida method	n/a	0.1335	0.1136	n/a	0.5124	0.4827

Note: We did not compute `integrate` for $K = 1600$ as it ran too slowly. Bold entries indicate the smallest error in that column.

1.2 | Background

In the continuum, a major class of techniques for classification problems relies upon the minimization of total variation (TV), for example, the famous Mumford-Shah [34] and Chan-Vese [16] algorithms. These methods are linked to Ginzburg-Landau methods by the fact that the Ginzburg-Landau functional Γ-converges to TV [29, 33] (a result that continues to hold in the graph context [23]). This motivated a common technique of minimizing the Ginzburg-Landau functional in place of TV, for example, in [20] two-class Chan-Vese segmentation was implemented by replacing TV with the Ginzburg-Landau functional; the resulting energy was minimized by using a fidelity forced MBO scheme.

Inspired by this continuum work, in [6] a method for graph classification was introduced based on minimizing the Ginzburg-Landau functional on a graph by evolving the graph ACE. The a priori information was incorporated by including a fidelity forcing term, leading to the equation

$$\frac{du}{dt} = -\Delta u - \frac{1}{\varepsilon} W' \circ u - \hat{\mu} P_Z(u - \tilde{f}),$$

where u is a labeling function which, due to the influence of a double-well potential (eg, $W(x) = x^2(x-1)^2$) will take values close to 0 and 1, indicating the two classes. The a priori knowledge is encoded in the reference $\tilde{f}$ which is supported on Z, a subset of the node set with corresponding projection operator P_Z. In the first term Δ denotes the graph Laplacian and $\varepsilon, \hat{\mu} > 0$ are parameters. All these ingredients will be explained in more detail in Sections 1.3 and 2.

In [32] an alternative method was introduced: a graph MBO scheme with fidelity forcing. The original MBO scheme, introduced in a continuum setting in [4] to approximate motion by mean curvature, is an iterative scheme consisting of diffusion alternated with a thresholding step. In [32] this scheme was discretized for use on graphs and the fidelity forcing term $-M(u - \tilde{f})$ (where M is a diagonal nonnegative matrix, see Section 2 for details) was added to the diffusion. Heuristically, this MBO scheme was expected to behave similarly to the graph ACE as the thresholding step resembles a "hard" version of the "soft" double-well potential nonlinearity in the ACE.

In [13] it was shown that the graph MBO scheme *without* fidelity forcing could be obtained as a special case of a SDIE scheme for the ACE (without fidelity forcing), if the smooth double-well potential was replaced by the double-obstacle potential defined in (1.2), and that solutions to the SDIE scheme converge to the solution of the graph ACE as the time step converges to zero. This double-obstacle potential was studied for the continuum ACE in [8-10] and was used in the graph context in [11]. In [14] a result similar to that of [13] was obtained for a mass-conserving graph MBO scheme. In this paper such a result will be established for the graph MBO scheme *with* fidelity forcing.

In [24] it was shown that the graph MBO scheme *pins* (or *freezes*) when the diffusion time is chosen too small, meaning that a single iteration of the scheme will not introduce any change as the diffusion step will not have pushed the value at any node past the threshold. In [13] it was argued that the SDIE scheme for graph ACE provides a relaxation of the MBO scheme: the hard threshold is replaced by a gradual threshold, which should allow for the use of smaller diffusion times without experiencing pinning. The current paper investigates what impact that has in practical problems.

1.3 | Groundwork

We briefly summarize the framework for analysis on graphs, following the summary in [13] of the detailed presentation in [24]. A graph $G=(V,E)$ will henceforth be defined to be a finite, simple, undirected, weighted, and connected graph without self-loops with vertex set V, edge set $E\subseteq V^2$, and weights $\{\omega_{ij}\}_{i,j\in V}$ with $\omega_{ij}\geq 0$, $\omega_{ij}=\omega_{ji}$, $\omega_{ii}=0$, and $\omega_{ij}>0$ if and only if $ij\in E$. We define the following function spaces on G (where $X\subseteq\mathbb{R}$, and $T\subseteq\mathbb{R}$ an interval):

$$\mathcal{V}:=\{u:V\to\mathbb{R}\},\qquad \mathcal{V}_X:=\{u:V\to X\},\qquad \mathcal{E}:=\{\varphi:E\to\mathbb{R}\}.$$
$$\mathcal{V}_{t\in T}:=\{u:T\to\mathcal{V}\},\qquad \mathcal{V}_{X,t\in T}:=\{u:T\to\mathcal{V}_X\}.$$

Defining $d_i:=\sum_{j\in V}\omega_{ij}$ to be the *degree* of vertex $i\in V$, we define inner products on $\mathcal{V}$ (or $\mathcal{V}_X$) and $\mathcal{E}$ (where $r\in[0,1]$):

$$\langle u,v\rangle_{\mathcal{V}}:=\sum_{i\in V}u_iv_id_i^r,\quad \langle\varphi,\phi\rangle_{\mathcal{E}}:=\frac{1}{2}\sum_{i,j\in V}\varphi_{ij}\phi_{ij}\omega_{ij},$$

and define the inner product on $\mathcal{V}_{t\in T}$ (or $\mathcal{V}_{X,t\in T}$)

$$(u,v)_{t\in T}:=\int_T\langle u(t),v(t)\rangle_{\mathcal{V}}\,dt=\sum_{i\in V}d_i^r\,(u_i,v_i)_{L^2(T;\mathbb{R})}.$$

These induce inner product norms $\|\cdot\|_{\mathcal{V}}$, $\|\cdot\|_{\mathcal{E}}$, and $\|\cdot\|_{t\in T}$. We also define on $\mathcal{V}$ the norm

$$\|u\|_\infty:=\max_{i\in V}|u_i|.$$

Next, we define the L^2 space:

$$L^2(T;\mathcal{V}):=\{u\in\mathcal{V}_{t\in T}\,|\,\|u\|_{t\in T}<\infty\},$$

and, for T an open interval, we define the Sobolev space $H^1(T;\mathcal{V})$ as the set of $u\in L^2(T;\mathcal{V})$ with weak derivative $du/dt\in L^2(T;\mathcal{V})$ defined by

$$\forall\varphi\in C_c^\infty(T;\mathcal{V})\ \left(u,\frac{d\varphi}{dt}\right)_{t\in T}=-\left(\frac{du}{dt},\varphi\right)_{t\in T},$$

where $C_c^\infty(T;\mathcal{V})$ is the set of $\varphi\in\mathcal{V}_{t\in T}$ that are infinitely differentiable with respect to time $t\in T$ and are compactly supported in T. By [13, Proposition 2.1], $u\in H^1(T;\mathcal{V})$ if and only if $u_i\in H^1(T;\mathbb{R})$ for each $i\in V$. We define the local H^1 space on any interval T (and likewise define the local L^2 space $L^2_{loc}(T;\mathcal{V})$):

$$H^1_{loc}(T;\mathcal{V}):=\left\{u\in\mathcal{V}_{t\in T}\;\middle|\;\forall a,b\in T,\ u\in H^1((a,b);\mathcal{V})\right\}.$$

For $A\subseteq V$, we define the *characteristic function* of A, $\chi_A\in\mathcal{V}$, by

$$(\chi_A)_i:=\begin{cases}1, & \text{if } i\in A,\\ 0, & \text{if } i\notin A.\end{cases}$$

Next, we introduce the graph gradient and Laplacian:

$$(\nabla u)_{ij}:=\begin{cases}u_j-u_i, & ij\in E,\\ 0, & \text{otherwise},\end{cases}\qquad (\Delta u)_i:=d_i^{-r}\sum_{j\in V}\omega_{ij}(u_i-u_j).$$

Note that Δ is positive semi-definite and self-adjoint with respect to $\mathcal{V}$. As shown in [24], these operators are related via:

$$\langle u, \Delta v\rangle_{\mathcal{V}} = \langle \nabla u, \nabla v\rangle_{\mathcal{E}}.$$

We can interpret Δ as a matrix. Define $D := diag(d)$ (ie, $D_{ii} := d_i$, and $D_{ij} := 0$ otherwise) to be the *diagonal matrix of degrees*. Then writing ω for the matrix of weights ω_{ij} we get

$$\Delta := D^{-r}(D - \omega).$$

From Δ we define the *graph diffusion operator*:

$$e^{-t\Delta}u := \sum_{n\geq 0} \frac{(-1)^n t^n}{n!}\Delta^n, u$$

where $v(t) = e^{-t\Delta}u$ is the unique solution to $dv/dt = -\Delta v$ with $v(0) = u$. Note that $e^{-t\Delta}\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is the vector of ones. By [13, Proposition 2.2] if $u \in H^1(T;\mathcal{V})$ and T is bounded below, then $e^{-t\Delta}u \in H^1(T;\mathcal{V})$ with

$$\frac{d}{dt}\left(e^{-t\Delta}u\right) = e^{-t\Delta}\frac{du}{dt} - e^{-t\Delta}\Delta u.$$

We recall from functional analysis the notation, for any linear operator $F : \mathcal{V} \to \mathcal{V}$,

$$\begin{aligned}
\sigma(F) &:= \{\lambda : \lambda \text{ an eigenvalue of } F\}\\
\rho(F) &:= \max\{|\lambda| : \lambda \in \sigma(F)\}\\
||F|| &:= \sup_{||u||_{\mathcal{V}}=1} ||Fu||_{\mathcal{V}}
\end{aligned}$$

and recall the standard result that if F is self-adjoint then $||F|| = \rho(F)$.

Finally, we recall some notation from [13]: for problems of the form $\operatorname{argmin}_x f(x)$ we write $f \simeq g$ and say f and g are *equivalent* when $g(x) = af(x) + b$ for $a > 0$ and b independent of x. As a result, replacing f by g does not affect the minimizers.

Lastly, we define the non-fidelity-forced versions of the graph MBO scheme, the graph ACE, and the SDIE scheme.

The MBO scheme is an iterative, two-step process, originally developed in [4] to approximate motion by mean curvature. On a graph, it is defined in [32] by the following iteration: for $u_n \in \mathcal{V}_{\{0,1\}}$, and $\tau > 0$ the *time step*,

1. $v_n := e^{-\tau\Delta}u_n$, that is, the diffused state of u_n after a time τ.
2. $u_{n+1} := \Theta(v_n)$ where Θ is defined by, for all $i \in V$ and $v \in \mathcal{V}$,

$$(\Theta(v))_i := \begin{cases} 1, & \text{if } v_i \geq 1/2,\\ 0, & \text{if } v_i < 1/2. \end{cases} \tag{1.1}$$

To define the *graph ACE*, we first define the *graph Ginzburg-Landau functional* as in [13] by

$$GL_\varepsilon(u) := \frac{1}{2}||\nabla u||_{\mathcal{E}}^2 + \frac{1}{\varepsilon}\langle W \circ u, \mathbf{1}\rangle_{\mathcal{V}}$$

where W is a double-well potential and $\varepsilon > 0$ is a scaling parameter. Then the ACE results from taking the $\langle\cdot,\cdot\rangle$ gradient flow of GL_ε, which for W differentiable is given by the ODE (where $\nabla_{\mathcal{V}}$ is the Hilbert space gradient on $\mathcal{V}$):

$$\frac{du}{dt} = -\nabla_{\mathcal{V}} GL_\varepsilon(u) = -\Delta u - \frac{1}{\varepsilon}W' \circ u.$$

To facilitate the SDIE link from [13] between the ACE and the MBO scheme, we will henceforth take W to be defined as:

$$W(x) := \begin{cases} \frac{1}{2}x(1-x), & \text{for } 0 \leq x \leq 1, \\ \infty, & \text{otherwise,} \end{cases} \tag{1.2}$$

the *double-obstacle potential* studied by Blowey and Elliott [8-10] in the continuum and Bosch et al. [11] on graphs. As W is not differentiable, we redefine the ACE via the subdifferential of W. As in [13] we say that a pair $(u, \beta) \in \mathcal{V}_{[0,1],t\in T} \times \mathcal{V}_{t\in T}$ is a solution to the double-obstacle ACE on an interval T if $u \in H^1_{loc}(T; \mathcal{V})$ and for a.e. $t \in T$

$$\varepsilon \frac{du}{dt}(t) + \varepsilon \Delta u(t) + \frac{1}{2}\mathbf{1} - u(t) = \beta(t), \qquad \beta(t) \in \mathcal{B}(u(t)),$$

where $\mathcal{B}(u)$ is the set (for $I_{[0,1]}(x) := 0$ if $x \in [0,1]$ and $I_{[0,1]}(x) := \infty$ otherwise)

$$\mathcal{B}(u) := \left\{ \alpha \in \mathcal{V} \,\middle|\, \forall i \in V, \alpha_i \in -\partial I_{[0,1]}(u_i) \right\}. \tag{1.3}$$

That is, $\mathcal{B}(u) = \emptyset$ if $u \notin \mathcal{V}_{[0,1]}$, and for $u \in \mathcal{V}_{[0,1]}$ it is the set of $\beta \in \mathcal{V}$ such that

$$\beta_i \in \begin{cases} [0, \infty), & u_i = 0, \\ \{0\}, & 0 < u_i < 1, \\ (-\infty, 0], & u_i = 1. \end{cases}$$

Finally, the SDIE scheme for the graph ACE is defined in [13] by the formula

$$u_{n+1} = e^{-\tau\Delta} u_n - \frac{\tau}{\varepsilon} W' \circ u_{n+1}$$

or more accurately, given the above detail with the subdifferential,

$$(1-\lambda)u_{n+1} - e^{-\tau\Delta}u_n + \frac{\lambda}{2}\mathbf{1} = \lambda\beta_{n+1},$$

where $\lambda := \tau/\varepsilon$ and $\beta_{n+1} \in \mathcal{B}(u_{n+1})$. The key results of [13] are then that:

- When $\tau = \varepsilon$, this scheme is exactly the MBO scheme.
- For ε fixed and $\tau \downarrow 0$, this scheme converges to the solution of the double-obstacle ACE (which is a well-posed ODE).

1.4 | Paper outline

The paper is structured as follows. In Section 1.3 we introduced important concepts and notation for the rest of the paper. Section 2 contains the main theoretical results of this paper. It defines the graph MBO scheme with fidelity forcing, the graph ACE with fidelity forcing, and the SDIE scheme for graph ACE with fidelity forcing. It proves well-posedness for the graph ACE with fidelity forcing and establishes the rigorous link between a particular SDIE scheme and the graph MBO with fidelity forcing. Moreover, it introduces a Lypunov functional for the SDIE scheme with fidelity forcing and proves convergence of solutions of the SDIE schemes to the solution of the graph ACE with fidelity forcing. In Section 3 we explain how the SDIE schemes can be used for graph classification. In particular, the modifications to the existing MBO-based classification algorithms based on the QR decomposition and Strang formula are introduced. Section 4 presents a comparison of the SDIE and MBO scheme for image segmentation applications, and an investigation of the uncertainty in these segmentations. In Appendix A it is shown that the application of the Euler method used in [32] can be seen as an approximation of the Lie product formula.

2 | THE ACE, THE MBO SCHEME, AND THE SDIE SCHEME WITH FIDELITY FORCING

2.1 | The MBO scheme with fidelity forcing

Following [20, 32], we introduce fidelity forcing into the MBO scheme by first defining a fidelity forced diffusion.

Definition 2.1 (Fidelity forced graph diffusion). For $u \in H^1_{loc}([0,\infty);\mathcal{V})$ and $u_0 \in \mathcal{V}$ we define fidelity forced diffusion to be:

$$\frac{du}{dt}(t) = -\Delta u(t) - M\big(u(t) - \tilde{f}\big) =: -Au(t) + M\tilde{f}, \qquad u(0) = u_0, \tag{2.1}$$

where $M := diag(\mu)$ for $\mu \in \mathcal{V}_{[0,\infty)} \setminus \{\mathbf{0}\}$ the fidelity parameter, $A := \Delta + M$, and $\tilde{f} \in \mathcal{V}_{[0,1]}$ is the *reference*. We define $Z := supp(\mu) \neq \emptyset$, which is the reference data we enforce fidelity on. Note that μ_i parameterizes the strength of the fidelity to the reference at vertex i. For the purposes of this section we shall treat μ and $\tilde{f}$ (and therefore M and Z) as fixed and given. Moreover, since $\tilde{f}$ only ever appears in the presence of M, we define $f := M\tilde{f}$ which is supported only on Z. Note that $f_i := \mu_i \tilde{f}_i \in [0, \mu_i]$.

Note. This fidelity term generalizes slightly the one used (for ACE) in [6], in which $\mu := \hat{\mu}\chi_Z$ for $\hat{\mu} > 0$ a parameter (ie, fidelity was enforced with equal strength on each vertex of the reference data) yielding $M = \hat{\mu}P_Z$ where P_Z is the projection map:

$$(P_Z u)_i = \begin{cases} u_i, & \text{if } i \in Z, \\ 0, & \text{if } i \notin Z. \end{cases}$$

This generalization has practical relevance, for example if the confidence in the accuracy of the reference were higher at some vertices of the reference data than at others it might be advantageous to use a fidelity parameter that is non-constant on the reference data. This is due to the link between the value of the fidelity parameter at a vertex and the statistical precision (ie, the inverse of the variance of the noise) of the reference at that vertex (see [7, Section 3.3] for details).

Proposition 2.2. *A is invertible with $\sigma(A) \subseteq (0, ||\Delta|| + ||\mu||_\infty]$.*

Proof. For the lower bound, we show that A is strictly positive definite. Let $u \neq \mathbf{0}$ be written $u = v + \alpha\mathbf{1}$ for $v \perp \mathbf{1}$. Then

$$\langle u, Au\rangle_{\mathcal{V}} = \langle v, \Delta v\rangle_{\mathcal{V}} + \langle u, Mu\rangle_{\mathcal{V}}$$

and note that both terms on the right-hand side are nonnegative. Next, if $v \neq \mathbf{0}$ then

$$\langle u, Au\rangle_{\mathcal{V}} \geq \langle v, \Delta v\rangle_{\mathcal{V}} = ||\nabla v||^2_{\mathcal{E}} > 0$$

since $v \perp \mathbf{1}$ and hence $\nabla v \neq \mathbf{0}$, since G is connected. Else, $v = \mathbf{0}$ so $\alpha \neq 0$ and

$$\langle u, Au\rangle_{\mathcal{V}} = \alpha^2 \langle \mathbf{1}, \mu\rangle_{\mathcal{V}} > 0.$$

For the upper bound: A is the sum of self-adjoint matrices, so is self-adjoint and hence has largest eigenvalue equal to $||A|| = ||\Delta + M|| \leq ||\Delta|| + ||M|| = ||\Delta|| + ||\mu||_\infty$. ∎

Theorem 2.3. *For given $u_0 \in \mathcal{V}$, (2.1) has a unique solution in $H^1_{loc}([0,\infty);\mathcal{V})$. The solution u to (2.1) is $C^1((0,\infty);\mathcal{V})$ and is given by the map (where I denotes the identity matrix):*

$$u(t) = S_t u_0 := e^{-tA}u_0 + A^{-1}(I - e^{-tA})f. \tag{2.2}$$

This solution map has the following properties:

i. *If $u_0 \leq v_0$ vertexwise, then for all $t \geq 0$, $S_t u_0 \leq S_t v_0$ vertexwise.*
ii. *$S_t : \mathcal{V}_{[0,1]} \to \mathcal{V}_{[0,1]}$ for all $t \geq 0$, that is, if $u_0 \in \mathcal{V}_{[0,1]}$ then $u(t) \in \mathcal{V}_{[0,1]}$.*

Proof. It is straightforward to check directly that (2.2) satisfies (2.1) and is C^1 on $(0, \infty)$. Uniqueness is given by a standard Picard-Lindelöf argument (see eg, [40, Corollary 2.6]).

i. By definition, $S_t v_0 - S_t u_0 = e^{-tA}(v_0 - u_0)$. Thus it suffices to show that e^{-tA} is a nonnegative matrix for $t \geq 0$. Note that the off-diagonal elements of $-tA$ are nonnegative: for $i \neq j$, $-tA_{ij} = -t\Delta_{ij} = td_i^r \omega_{ij} \geq 0$. Thus for some $a > 0$, $Q := aI - tA$ is a nonnegative matrix and thus e^Q is a nonnegative matrix. It follows that $e^{-tA} = e^{-a}e^Q$ is a nonnegative matrix.
ii. Let $u_0 \in \mathcal{V}_{[0,1]}$ and recall that $\tilde{f} \in \mathcal{V}_{[0,1]}$. Suppose that for some $t > 0$ and some $i \in V$, $u_i(t) < 0$. Then

$$\min_{t' \in [0,t]} \min_{i \in V} u_i(t') < 0$$

and since each u_i is continuous this minimum is attained at some $t^* \in [0, t]$ and $i^* \in V$. Fix such a t^*. Then for any i^* minimizing $u(t^*)$, since $u_{i^*}(t^*) < 0$ we must have $t^* > 0$, so u_{i^*} is differentiable at t^* with $du_{i^*}/dt(t^*) = 0$. However by (2.1)

$$\frac{du_{i^*}}{dt}(t^*) = -(\Delta u(t^*))_{i^*} + \mu_{i^*}(\tilde{f}_{i^*} - u_{i^*}(t^*)).$$

We claim that we can choose a suitable minimizer i^* such that this is strictly positive. First, since any such i^* is a minimizer of $u(t^*)$, and $\tilde{f}_i \geq 0$ for all i, it follows that each term is nonnegative. Next, suppose such an i^* has a neighbor j such that $u_j(t^*) > u_{i^*}(t^*)$, then it follows that $(\Delta u(t^*))_{i^*} < 0$ and we have the claim. Otherwise, all the neighbors of that i^* are also minimizers of $u(t^*)$. Repeating this same argument on each of those, we either have the claim for the above reason or we find a minimizer $i^* \in Z$, since G is connected. But in that case $\mu_{i^*}(\tilde{f}_{i^*} - u_{i^*}(t^*)) \geq -\mu_{i^*} u_{i^*}(t^*) > 0$, since μ is strictly positive on Z, and we again have the claim. Hence $du_{i^*}/dt(t^*) > 0$, a contradiction. Therefore $u_i(t) \geq 0$ for all t. The case for $u_i(t) \leq 1$ is likewise.

■

Definition 2.4 (Graph MBO with fidelity forcing). For $u_0 \in \mathcal{V}_{[0,1]}$ we follow [20, 32], and define the sequence of MBO iterates by diffusing with fidelity for a time $\tau \geq 0$ and then thresholding, that is

$$(u_{n+1})_i = \begin{cases} 1, & \text{if } (S_\tau u_n)_i \geq 1/2, \\ 0, & \text{if } (S_\tau u_n)_i < 1/2, \end{cases} \tag{2.3}$$

where S_τ is the solution map from (2.2). Note that (2.3) has variational form similar to that given for graph MBO in [24], which we can then re-write as in [13]:

$$u_{n+1} \in \operatorname*{argmin}_{u \in \mathcal{V}_{[0,1]}} \langle \mathbf{1} - 2S_\tau u_n, u \rangle_{\mathcal{V}} \simeq \frac{1}{2\tau} \langle \mathbf{1} - u, u \rangle_{\mathcal{V}} + \frac{||u - S_\tau u_n||_{\mathcal{V}}^2}{2\tau}. \tag{2.4}$$

2.2 | The ACE with fidelity forcing

To derive the ACE with fidelity forcing, we re-define the Ginzburg-Landau energy to include a fidelity term (recalling the potential W from (1.2)):

$$GL_{\varepsilon,\mu,\tilde{f}}(u) := \frac{1}{2} ||\nabla u||_{\mathcal{E}}^2 + \frac{1}{\varepsilon} \langle W \circ u, \mathbf{1} \rangle_{\mathcal{V}} + \frac{1}{2} \langle u - \tilde{f}, M(u - \tilde{f}) \rangle_{\mathcal{V}}. \tag{2.5}$$

Taking the $\langle \cdot, \cdot \rangle_{\mathcal{V}}$ gradient flow of (2.5) we obtain the ACE with fidelity:

$$\varepsilon \frac{du}{dt}(t) + \varepsilon(\Delta u(t) + M(u(t) - \tilde{f})) + \frac{1}{2}\mathbf{1} - u(t) = \beta(t), \qquad \beta(t) \in \mathcal{B}(u(t)). \tag{2.6}$$

Where $\mathcal{B}(u(t))$ is defined as in (1.3). Recalling that $A \coloneqq \Delta + M$ and $f \coloneqq M\tilde{f}$, we can rewrite the ODE in (2.6) as

$$\varepsilon \frac{du}{dt}(t) + \varepsilon A u(t) - \varepsilon f + \frac{1}{2}\mathbf{1} - u(t) = \beta(t).$$

As in [13], we can give an explicit expression for β given sufficient regularity on u.

Theorem 2.5. *Let (u, β) obey (2.6) at a.e. $t \in T$, with $u \in H^1_{loc}(T; \mathcal{V}) \cap C^0(T; \mathcal{V}) \cap \mathcal{V}_{[0,1], t \in T}$. Then for all $i \in V$ and a.e. $t \in T$,*

$$\beta_i(t) = \begin{cases} \frac{1}{2} + \varepsilon(\Delta u(t))_i - \varepsilon f_i, & u_i(t) = 0, \\ 0, & u_i(t) \in (0,1), \\ -\frac{1}{2} + \varepsilon(\Delta u(t))_i + \varepsilon(\mu_i - f_i), & u_i(t) = 1. \end{cases} \tag{2.7}$$

Hence at a.e. $t \in T$,

$$\beta(t) \in \mathcal{V}_{[-1/2, 1/2]}.$$

Proof. Follows as in [13, Theorem 2.2] *mutatis mutandis*. ∎

Thus following [13] we define the double-obstacle ACE with fidelity forcing.

Definition 2.6 (Double-obstacle ACE with fidelity forcing). Let T be an interval. Then a pair $(u, \beta) \in \mathcal{V}_{[0,1], t \in T} \times \mathcal{V}_{t \in T}$ is a solution to double-obstacle ACE with fidelity forcing on T when $u \in H^1_{loc}(T; \mathcal{V}) \cap C^0(T; \mathcal{V})$ and for almost every $t \in T$,

$$\varepsilon \frac{du}{dt}(t) + \varepsilon A u(t) - \varepsilon f + \frac{1}{2}\mathbf{1} - u(t) = \beta(t), \qquad \beta(t) \in \mathcal{B}(u(t)). \tag{2.8}$$

We frequently will refer to just u as a solution to (2.8), since β is a.e. uniquely determined as a function of u by (2.7).

We now demonstrate that this has the same key properties, *mutatis mutandis*, as the ACE in [13].

Theorem 2.7. *Let $T = [0, T_0]$ or $[0, \infty)$. Then*:

(a) *(Existence) For any given $u_0 \in \mathcal{V}_{[0,1]}$, there exists a (u, β) as in Definition 2.6 with $u(0) = u_0$.*
(b) *(Comparison principle) If $(u, \beta), (v, \gamma) \in \mathcal{V}_{[0,1], t \in T} \times \mathcal{V}_{t \in T}$ with $u, v \in H^1_{loc}(T; \mathcal{V}) \cap C^0(T; \mathcal{V})$ satisfy*

$$\varepsilon \frac{du}{dt}(t) + \varepsilon A u(t) - \varepsilon f + \frac{1}{2}\mathbf{1} - u(t) \geq \beta(t), \qquad \beta(t) \in \mathcal{B}(u(t)), \tag{2.9}$$

and

$$\varepsilon \frac{dv}{dt}(t) + \varepsilon A v(t) - \varepsilon f + \frac{1}{2}\mathbf{1} - v(t) \leq \gamma(t), \qquad \gamma(t) \in \mathcal{B}(v(t)), \tag{2.10}$$

vertexwise at a.e. $t \in T$, and $v(0) \leq u(0)$ vertexwise, then $v(t) \leq u(t)$ vertexwise for all $t \in T$.
(c) *(Uniqueness) If (u, β) and (v, γ) are as in Definition 2.6 with $u(0) = v(0)$ then $u(t) = v(t)$ for all $t \in T$ and $\beta(t) = \gamma(t)$ at a.e. $t \in T$.*
(d) *(Gradient flow) For u as in Definition 2.6, $GL_{\varepsilon, \mu, \tilde{f}}(u(t))$ monotonically decreases.*

(e) *(Weak form)* $u \in \mathcal{V}_{[0,1],t\in T} \cap H^1_{loc}(T;\mathcal{V}) \cap C(T;\mathcal{V})$ *(and associated* $\beta(t) = \varepsilon \frac{du}{dt}(t) + \varepsilon A u(t) - \varepsilon f + \frac{1}{2}\mathbf{1} - u(t)$ *a.e.) is a solution to* (2.8) *if and only if for almost every* $t \in T$ *and all* $\eta \in \mathcal{V}_{[0,1]}$

$$\left\langle \varepsilon \frac{du}{dt} + \varepsilon A u(t) - \varepsilon f + \frac{1}{2}\mathbf{1} - u(t), \eta - u(t) \right\rangle_{\mathcal{V}} \geq 0. \tag{2.11}$$

(f) *(Explicit form)* $(u,\beta) \in \mathcal{V}_{[0,1],t\in T} \times \mathcal{V}_{t\in T}$ *satisfies Definition* 2.6 *if and only if for a.e.* $t \in T$, $\beta(t) \in \mathcal{B}(u(t))$, $\beta(t) \in \mathcal{V}_{[-1/2,1/2]}$ *and (for* $B := A - \varepsilon^{-1} I$ *and* $\varepsilon^{-1} \notin \sigma(A)$*):*

$$u(t) = e^{-tB}u(0) + B^{-1}\left(I - e^{-tB}\right)\left(f - \frac{1}{2\varepsilon}\mathbf{1}\right) + \frac{1}{\varepsilon}\int_0^t e^{-(t-s)B}\beta(s)\, ds. \tag{2.12}$$

(g) *(Lipschitz regularity) For* u *as in Definition* 2.6, *if* $\varepsilon^{-1} \notin \sigma(A)$, *then* $u \in C^{0,1}(T;\mathcal{V})$.

(h) *(Well-posedness) Let* $u_0, v_0 \in \mathcal{V}_{[0,1]}$ *define the ACE trajectories* u, v *as in Definition* 2.6, *and suppose* $\varepsilon^{-1} \notin \sigma(A)$. *Then, for* $\xi_1 := \min \sigma(A)$,

$$||u(t) - v(t)||_{\mathcal{V}} \leq e^{-\xi_1 t} e^{t/\varepsilon} ||u_0 - v_0||_{\mathcal{V}}. \tag{2.13}$$

Proof.

(a) We prove this as Theorem 2.17.

(b) We follow the proof of [13, Theorem B.2]. Letting $w := v - u$ and subtracting (2.9) from (2.10), we have that

$$\varepsilon \frac{dw}{dt}(t) + \varepsilon A w(t) - w(t) \leq \gamma(t) - \beta(t)$$

vertexwise at a.e. $t \in T$. Next, take the inner product with $w_+ := \max(w, 0)$, the vertexwise positive part of w:

$$\varepsilon \left\langle \frac{dw}{dt}(t), w_+(t) \right\rangle_{\mathcal{V}} + \varepsilon \langle A w(t), w_+(t)\rangle_{\mathcal{V}} - \langle w(t), w_+(t)\rangle_{\mathcal{V}} \leq \langle \gamma(t) - \beta(t), w_+(t)\rangle_{\mathcal{V}}.$$

As in the proof of [13, Theorem B.2], the $RHS \leq 0$. For the rest of the proof to go through as in that theorem, it suffices to check that $\langle A w(t), w_+(t)\rangle \geq \langle A w_+(t), w_+(t)\rangle$. But by [13, Proposition B.1], $\langle \Delta w(t), w_+(t)\rangle_{\mathcal{V}} \geq \langle \Delta w_+(t), w_+(t)\rangle_{\mathcal{V}}$, and $\langle M w(t), w_+(t)\rangle = \langle M w_+(t), w_+(t)\rangle$ since M is diagonal, so the proof follows as in [13, Theorem B.2].

(c) Follows from (b): if (u,β) and (v,γ) have $u(0) = v(0)$ and both solve (2.8), then applying the comparison principle in both directions gives that $u(t) = v(t)$ at all $t \in T$. Then (2.7) gives that $\beta(t) = \gamma(t)$ at a.e. $t \in T$.

(d) We prove this in Theorem 2.20 for the solution given by Theorem 2.17, which by uniqueness is the general solution.

(e) Let u solve (2.8). Then for a.e. $t \in T$, $\beta(t) \in \mathcal{B}(u(t))$, and at such t we have $\langle \beta(t), \eta - u(t)\rangle_{\mathcal{V}} \geq 0$ for all $\eta \in \mathcal{V}_{[0,1]}$ as in the proof of [13, Theorem 3.8], so u satisfies (2.11). Next, if u satisfies (2.11) at $t \in T$, then as in the proof of [13, Theorem 3.8], $\beta(t) := \varepsilon \frac{du}{dt}(t) + \varepsilon A u(t) - \varepsilon f + \frac{1}{2}\mathbf{1} - u(t) \in \mathcal{B}(u(t))$, as desired.

(f) Let $(u,\beta) \in \mathcal{V}_{[0,1],t\in T} \times \mathcal{V}_{t\in T}$. **If:** We check that (2.12) satisfies (2.8). Note first that we can rewrite (2.8) as

$$\frac{du}{dt}(t) + B u(t) - f + \frac{1}{2\varepsilon}\mathbf{1} = \frac{1}{\varepsilon}\beta(t).$$

Next, let u be as in (2.12). Then it is easy to check that

$$\frac{du}{dt}(t) = -B e^{-tB}u(0) + e^{-tB}\left(f - \frac{1}{2\varepsilon}\mathbf{1}\right) + \frac{1}{\varepsilon}\beta(t) - \frac{1}{\varepsilon}B\int_0^t e^{-(t-s)B}\beta(s)\, ds$$

and that this satisfies (2.8). Next, we check the regularity of u. The continuity of u is immediate, as it is a sum of two smooth terms and the integral of a locally bounded function. To check that $u \in H^1_{loc}$: u is bounded, so is locally L^2,

and by above du/dt is a sum of (respectively) two smooth functions, a bounded function and the integral of a locally bounded function, so is locally bounded and hence locally L^2.

Only if: We saw that (2.12) solves (2.8) with $\beta(t) \in \mathcal{B}(u(t))$ and $\beta(t) \in \mathcal{V}_{[-1/2,1/2]}$, and by (c) such solutions are unique.

(g) We follow the proof of [13, Theorem 3.13]. Let $0 \le t_1 < t_2$. Since (2.8) is time-translation invariant, we have by (f) that

$$u(t_2) = e^{-(t_2-t_1)B}u(t_1) + B^{-1}\left(I - e^{-(t_2-t_1)B}\right)\left(f - \frac{1}{2\varepsilon}\mathbf{1}\right) + \frac{1}{\varepsilon}\int_0^{t_2-t_1} e^{-sB}\beta(t_2-s)\,ds$$

and so, writing $B_s := (e^{-sB} - I)/s$ for $s > 0$ (which we note commutes with B),

$$u(t_2) - u(t_1) = (t_2-t_1)B_{t_2-t_1}\left(u(t_1) - B^{-1}\left(f - \frac{1}{2\varepsilon}\mathbf{1}\right)\right) + \frac{1}{\varepsilon}\int_0^{t_2-t_1} e^{-sB}\beta(t_2-s)\,ds.$$

Note that B_s is self-adjoint, and as $-B$ has largest eigenvalue less than ε^{-1} we have $||B_s|| < (e^{s/\varepsilon} - 1)/s$, with RHS monotonically increasing in s for $s > 0$.[1] Since $f \in \mathcal{V}_{[0,||\mu||_\infty]}$ and for all t, $\beta(t) \in \mathcal{V}_{[-1/2,1/2]}$ and $u(t) \in \mathcal{V}_{[0,1]}$, we have for $t_2 - t_1 < 1$:

$$\begin{aligned}
\frac{||u(t_2) - u(t_1)||_\mathcal{V}}{t_2 - t_1} &\le ||B_{t_2-t_1}|| \cdot \left(1 + ||\mu||_\infty ||B^{-1}|| + \frac{1}{2\varepsilon}||B^{-1}||\right)||\mathbf{1}||_\mathcal{V} + \frac{1}{\varepsilon}\operatorname*{ess\,sup}_{s\in[0,t_2-t_1]}\left|\left|e^{-sB}\beta(t_2-s)\right|\right|_\mathcal{V} \\
&< \frac{e^{(t_2-t_1)/\varepsilon} - 1}{t_2 - t_1} \cdot \left(1 + ||\mu||_\infty ||B^{-1}|| + \frac{1}{2\varepsilon}||B^{-1}||\right)||\mathbf{1}||_\mathcal{V} + \frac{1}{\varepsilon}\sup_{s\in[0,t_2-t_1]}\left|\left|e^{-sB}\right|\right| \cdot \frac{1}{2}||\mathbf{1}||_\mathcal{V} \\
&\le \frac{e^{(t_2-t_1)/\varepsilon} - 1}{t_2 - t_1} \cdot \left(1 + ||\mu||_\infty ||B^{-1}|| + \frac{1}{2\varepsilon}||B^{-1}||\right)||\mathbf{1}||_\mathcal{V} + \frac{1}{\varepsilon}e^{(t_2-t_1)/\varepsilon} \cdot \frac{1}{2}||\mathbf{1}||_\mathcal{V} \\
&< \left(\left(1 + ||\mu||_\infty ||B^{-1}|| + \frac{1}{2\varepsilon}||B^{-1}||\right)\left(e^{1/\varepsilon} - 1\right) + \frac{1}{2\varepsilon}e^{1/\varepsilon}\right)||\mathbf{1}||_\mathcal{V}
\end{aligned}$$

and for $t_2 - t_1 \ge 1$ we simply have

$$\frac{||u(t_2) - u(t_1)||_\mathcal{V}}{t_2 - t_1} \le ||u(t_2) - u(t_1)||_\mathcal{V} \le ||\mathbf{1}||_\mathcal{V}$$

completing the proof.

(h) We prove this as Theorem 2.21 for the solution given by Theorem 2.17, which by uniqueness is the general solution.

■

Note. Given the various forward references in the above proof, we take care to avoid circularity by not using the corresponding results until they have been proven.

2.3 | The SDIE scheme with fidelity forcing and link to the MBO scheme

Definition 2.8 (SDIE scheme with fidelity forcing, cf. [13, Definition 4.1]). For $u_0 \in \mathcal{V}_{[0,1]}$, $n \in \mathbb{N}$, and $\lambda := \tau/\varepsilon \in [0, 1]$ we define the SDIE scheme iteratively:

$$(1 - \lambda)u_{n+1} - \mathcal{S}_\tau u_n + \frac{\lambda}{2}\mathbf{1} = \lambda\beta_{n+1} \tag{2.14}$$

for a $\beta_{n+1} \in \mathcal{B}(u_{n+1})$ to be characterized in Theorem 2.9.

As in [13], we have the key theorem linking the MBO scheme and the SDIE schemes for the ACE.

[1] $\frac{d}{ds}((e^{s/\varepsilon} - 1)/s) = s^{-2}e^{s/\varepsilon}\left(e^{-s/\varepsilon} - 1 + s/\varepsilon\right) > 0$ for $s > 0$.

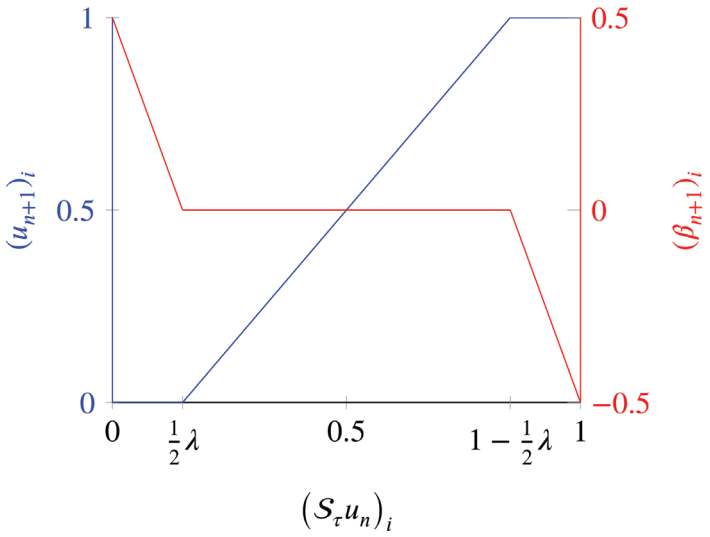


FIGURE 1 Plot of the SDIE updates u_{n+1} (blue, left axis, see (2.16)) and β_{n+1} (orange, right axis) at vertex i for $0 \leq \lambda < 1$ as a function of the fidelity forced diffused value at i. Cf. [13, Figure 1]

Theorem 2.9 (Cf. [13, Theorem 4.2]). *For $\lambda \in [0,1]$, the pair (u_{n+1}, β_{n+1}) is a solution to the SDIE scheme (2.14) for some $\beta_{n+1} \in \mathcal{B}(u_{n+1})$ if and only if u_{n+1} solves:*

$$u_{n+1} \in \operatorname*{argmin}_{u \in \mathcal{V}_{[0,1]}} \lambda \langle u, \mathbf{1} - u \rangle_{\mathcal{V}} + ||u - S_\tau u_n||^2_{\mathcal{V}}. \tag{2.15}$$

Note that for $\lambda = 1$ (2.15) is equivalent to the variational problem (2.4) that defines the MBO scheme. Furthermore, (2.15) has unique solution for $\lambda \in [0,1)$

$$(u_{n+1})_i = \begin{cases} 0, & \text{if } (S_\tau u_n)_i < \frac{1}{2}\lambda, \\ \frac{1}{2} + \frac{(S_\tau u_n)_i - 1/2}{1-\lambda}, & \text{if } \frac{1}{2}\lambda \leq (S_\tau u_n)_i < 1 - \frac{1}{2}\lambda, \\ 1, & \text{if } (S_\tau u_n)_i \geq 1 - \frac{1}{2}\lambda, \end{cases} \tag{2.16}$$

with corresponding $\beta_{n+1} = \lambda^{-1}\left((1-\lambda)u - S_\tau u_n + \frac{\lambda}{2}\mathbf{1}\right)$, and solutions for $\lambda = 1$

$$(u_{n+1})_i \in \begin{cases} \{1\}, & (S_\tau u_n)_i > 1/2, \\ [0,1], & (S_\tau u_n)_i = 1/2, \\ \{0\}, & (S_\tau u_n)_i < 1/2. \end{cases} \tag{2.17}$$

(ie, the MBO thresholding) with corresponding $\beta_{n+1} = \frac{1}{2}\mathbf{1} - S_\tau u_n$.

Proof. Identical to the proof of [13, Theorem 4.2] with the occurrences of "$e^{-\tau\Delta}$" in each instance replaced by "S_τ." ∎

As in [13, Figure 1], we can plot (2.16) to visualize the SDIE scheme (2.14) as a piecewise linear relaxation of the MBO thresholding rule. Next, we note that we have the same Lipschitz continuity property from [13, Theorem 4.4].

Theorem 2.10 (Cf. [13, Theorem 4.4]). *For $\lambda \in [0,1)$[2] and all $n \in \mathbb{N}$, if u_n and v_n are defined according to Definition 2.8 with initial states $u_0, v_0 \in \mathcal{V}_{[0,1]}$ and $\xi_1 := \min \sigma(A)$ then*

$$||u_n - v_n||_{\mathcal{V}} \leq e^{-n\xi_1 \tau}(1-\lambda)^{-n}||u_0 - v_0||_{\mathcal{V}}. \tag{2.18}$$

Proof. Follows as in [13, Theorem 4.4] *mutatis mutandis*. ∎

[2]For the MBO case $\lambda = 1$ the thresholding is discontinuous so we do not get an analogous property.

2.4 | A Lyapunov functional for the SDIE scheme

Lemma 2.11 (Cf. [24, Lemma 4.6]). *The functional on* $\mathcal{V}$

$$J(u) := \langle u, \mathbf{1} - 2A^{-1}(I - e^{-\tau A})f - e^{-\tau A}u\rangle_{\mathcal{V}}$$

has the following properties:

i. *It is strictly concave.*
ii. *It has first variation at* u

$$L_u(v) := \left\langle v, \mathbf{1} - 2A^{-1}(I - e^{-\tau A})f - 2e^{-\tau A}u\right\rangle_{\mathcal{V}} = \langle v, \mathbf{1} - 2S_\tau u\rangle_{\mathcal{V}}.$$

Proof. Let $w := \mathbf{1} - 2A^{-1}(I - e^{-\tau A})f$. We expand around u:

$$\begin{aligned} J(u+tv) &= \langle u + tv, w - e^{\tau A}(u+tv)\rangle_{\mathcal{V}} \\ &= \langle u, w - e^{\tau A}u\rangle_{\mathcal{V}} + t\langle v, w - e^{\tau A}u\rangle_{\mathcal{V}} - t\langle u, e^{-tA}v\rangle_{\mathcal{V}} - t^2\langle v, e^{-\tau A}v\rangle_{\mathcal{V}}. \end{aligned}$$

i. $\frac{d^2}{dt^2}J(u+tv) = -2\langle v, e^{-\tau A}v\rangle_{\mathcal{V}} < 0$ for $v \neq \mathbf{0}$.
ii. Since e^{-tA} is self-adjoint, $J(u+tv) = J(u) + tL_u(v) + \mathcal{O}(t^2)$.

■

Theorem 2.12 (Cf. [13, Theorem 4.9]). *For* $0 \le \lambda \le 1$ *we define on* $\mathcal{V}_{[0,1]}$ *the functional*

$$H(u) := J(u) + (\lambda - 1)\langle u, \mathbf{1} - u\rangle_{\mathcal{V}}. \tag{2.19}$$

This has uniform lower bound

$$H(u) \ge -2\tau||f||_{\mathcal{V}}||\mathbf{1}||_{\mathcal{V}} \tag{2.20}$$

and is a Lyapunov functional for (2.14), *that is,* $H(u_{n+1}) \le H(u_n)$ *with equality if and only if* $u_{n+1} = u_n$ *for* u_{n+1} *defined by* (2.14). *In particular, we have that*

$$H(u_n) - H(u_{n+1}) \ge (1-\lambda)\,||u_{n+1} - u_n||_{\mathcal{V}}^2\,. \tag{2.21}$$

Proof. We can rewrite H as:

$$\begin{aligned} H(u) &= \lambda\langle u, \mathbf{1} - u\rangle_{\mathcal{V}} + \langle u, u - 2A^{-1}(I - e^{-\tau A})f - e^{-\tau A}u\rangle_{\mathcal{V}} \\ &\ge \langle u, (I - e^{-\tau A})u\rangle_{\mathcal{V}} - 2\langle u, A^{-1}(I - e^{-\tau A})f\rangle_{\mathcal{V}} && \text{since } u \in \mathcal{V}_{[0,1]} \\ &\ge -2\langle u, A^{-1}(I - e^{-\tau A})f\rangle_{\mathcal{V}} && \text{since } I - e^{-\tau A} \text{ is positive definite} \\ &\ge -2||f||_{\mathcal{V}}||u||_{\mathcal{V}}\left|\left|A^{-1}(I - e^{-\tau A})\right|\right| \\ &\ge -2||f||_{\mathcal{V}}||\mathbf{1}||_{\mathcal{V}}\left|\left|A^{-1}(I - e^{-\tau A})\right|\right| \ge -2\tau||f||_{\mathcal{V}}||\mathbf{1}||_{\mathcal{V}}, \end{aligned}$$

where the final line follows since $A^{-1}(I - e^{-\tau A})$ is self-adjoint (since A is) and has eigenvalues

$$\left\{\frac{1 - e^{-\tau\lambda}}{\lambda}\;\middle|\;\lambda \in \sigma(A)\right\}$$

so we have by Proposition 2.2 that

$$\left|\left|A^{-1}(I - e^{-\tau A})\right|\right| \le \sup_{x\in(0,||\Delta||+||\mu||_\infty]} \frac{1 - e^{-\tau x}}{x}$$

$$= \lim_{x\to 0} \frac{1-e^{-\tau x}}{x} \quad \text{as } x \mapsto x^{-1}(1-e^{-\tau x}) \text{ is monotonically decreasing}^3$$
$$= \tau.$$

Next we show that H is a Lyapunov functional. By the concavity of J:

$$\begin{aligned}
H(u_n) - H(u_{n+1}) &= J(u_n) - J(u_{n+1}) + (1-\lambda)\langle u_{n+1}, \mathbf{1} - u_{n+1}\rangle_{\mathcal{V}} - (1-\lambda)\langle u_n, \mathbf{1} - u_n\rangle_{\mathcal{V}} \\
&\geq L_{u_n}(u_n - u_{n+1}) + (1-\lambda)\langle u_{n+1}, \mathbf{1} - u_{n+1}\rangle_{\mathcal{V}} - (1-\lambda)\langle u_n, \mathbf{1} - u_n\rangle_{\mathcal{V}} \ (*) \\
&= \langle u_n - u_{n+1}, \mathbf{1} - 2\mathcal{S}_\tau u_n\rangle_{\mathcal{V}} + (1-\lambda)\langle u_{n+1}, \mathbf{1} - u_{n+1}\rangle_{\mathcal{V}} - (1-\lambda)\langle u_n, \mathbf{1} - u_n\rangle_{\mathcal{V}} \\
&= \langle u_n - u_{n+1}, \mathbf{1} - 2\mathcal{S}_\tau u_n\rangle_{\mathcal{V}} + (1-\lambda)(\langle u_{n+1} - u_n, \mathbf{1}\rangle_{\mathcal{V}} + \langle u_n, u_n\rangle_{\mathcal{V}} - \langle u_{n+1}, u_{n+1}\rangle_{\mathcal{V}}) \\
&= \langle u_n - u_{n+1}, \lambda\mathbf{1} - 2\mathcal{S}_\tau u_n + (1-\lambda)u_{n+1} + (1-\lambda)u_n\rangle_{\mathcal{V}} \\
&= \langle u_n - u_{n+1}, 2\lambda\beta_{n+1} + (1-\lambda)(u_n - u_{n+1})\rangle_{\mathcal{V}} \text{ by (2.14)} \\
&\geq (1-\lambda)\,||u_{n+1} - u_n||_{\mathcal{V}}^2 \geq 0
\end{aligned}$$

with equality in (*) if and only if $u_{n+1} = u_n$ as the concavity of J is strict, and where the last line follows since by $\beta_{n+1} \in \mathcal{B}(u_{n+1})$

$$(\beta_{n+1})_i((u_n)_i - (u_{n+1})_i) = \begin{Bmatrix} \text{“}\geq 0\text{”} \cdot (u_n)_i, & (u_{n+1})_i = 0 \\ 0, & (u_{n+1})_i \in (0,1) \\ \text{“}\leq 0\text{”} \cdot ((u_n)_i - 1), & (u_{n+1})_i = 1 \end{Bmatrix} \geq 0$$

and so $\langle u_n - u_{n+1}, \beta_{n+1}\rangle_{\mathcal{V}} \geq 0$. ∎

Corollary 2.13. *For $\lambda = 1$ (ie, the MBO case) the sequence u_n defined by (2.14) is eventually constant.*

For $0 \leq \lambda \leq 1$, the sum

$$\sum_{n=0}^{\infty} ||u_{n+1} - u_n||_{\mathcal{V}}^2$$

converges, and hence

$$\lim_{n\to\infty} ||u_{n+1} - u_n||_{\mathcal{V}} = 0.$$

Proof. For the first claim, the proof follows as in [24, Proposition 4.6] *mutatis mutandis*. For the second claim, the proof follows as in [13, Corollary 4.10] *mutatis mutandis*. ∎

2.5 | Convergence of the SDIE scheme with fidelity forcing

Following [13], we first derive the nth term for the semi-discrete scheme.

Proposition 2.14 (Cf. [13, Proposition 5.1]). *For the sake of notation, define*:

$$w := -\frac{1}{2}\lambda\mathbf{1} + A^{-1}(I - e^{-\tau A})f.$$

Then for $\lambda \in [0,1)$ the sequence generated by (2.14) is given by:

$$u_n = (1-\lambda)^{-n}e^{-n\tau A}u_0 + \sum_{k=1}^{n}(1-\lambda)^{-k}e^{-(k-1)\tau A}w + \frac{\lambda}{1-\lambda}\sum_{k=1}^{n}(1-\lambda)^{-(n-k)}e^{-(n-k)\tau A}\beta_k. \tag{2.22}$$

[3] $\frac{d}{dx}(x^{-1}(1-e^{-\tau x})) = x^{-2}e^{-\tau x}(1+\tau x - e^{\tau x}) \leq 0.$

Proof. We can rewrite (2.14) as

$$(1-\lambda)u_{n+1} = e^{-\tau A}u_n + A^{-1}(I-e^{-tA})f - \frac{1}{2}\lambda\mathbf{1} + \lambda\beta_{n+1} = e^{-\tau A}u_n + w + \lambda\beta_{n+1}.$$

We then check (2.22) inductively. The $n=0$ case is trivial, and we have that

$$\begin{aligned}
&(1-\lambda)^{-1}e^{-\tau A}u_n + (1-\lambda)^{-1}w + \frac{\lambda}{1-\lambda}\beta_{n+1} \\
&= (1-\lambda)^{-(n+1)}e^{-(n+1)\tau A}u_0 + \sum_{k=1}^{n}(1-\lambda)^{-(k+1)}e^{-k\tau A}w + \frac{\lambda}{1-\lambda}\sum_{k=1}^{n}(1-\lambda)^{-((n+1)-k)}e^{-((n+1)-k)\tau A}\beta_k \\
&\quad + (1-\lambda)^{-1}w + \frac{\lambda}{1-\lambda}\beta_{n+1} \\
&= (1-\lambda)^{-(n+1)}e^{-(n+1)\tau A}u_0 + \sum_{k=0}^{n}(1-\lambda)^{-(k+1)}e^{-k\tau A}w + \frac{\lambda}{1-\lambda}\sum_{k=1}^{n+1}(1-\lambda)^{-((n+1)-k)}e^{-((n+1)-k)\tau A}\beta_k \\
&= u_{n+1}
\end{aligned}$$

completing the proof. ∎

Next, we consider the asymptotics of each term in (2.22).

Theorem 2.15. *Considering $\mathcal{O}$ relative to the limit of $\tau \downarrow 0$ and $n\to\infty$ with $n\tau - t \in [0,\tau)$ for some fixed $t\geq 0$ and for fixed $\varepsilon > 0$ (with $\varepsilon^{-1}\notin\sigma(A)$)*[4]*, and recalling that $\lambda := \tau/\varepsilon$, $B := A - \varepsilon^{-1}I$ and $w := -\frac{1}{2}\lambda\mathbf{1} + A^{-1}(I-e^{-\tau A})f$:*

i. $(1-\lambda)^{-n}e^{-n\tau A}u_0 = e^{t/\varepsilon}e^{-tA}u_0 + \mathcal{O}(\tau) = e^{-tB}u_0 + \mathcal{O}(\tau)$.
ii. $\sum_{k=1}^{n}(1-\lambda)^{-k}e^{-(k-1)\tau A}w = B^{-1}(I-e^{-tB})f - \frac{1}{2\varepsilon}B^{-1}(I-e^{-tB})\mathbf{1} + \mathcal{O}(\tau)$.
iii. $\frac{\lambda}{1-\lambda}\sum_{k=1}^{n}(1-\lambda)^{-(n-k)}e^{-(n-k)\tau A}\beta_k = \lambda\sum_{k=1}^{n}e^{-(n-k)\tau B}\beta_k + \mathcal{O}(\tau)$.

Hence by (2.22)*, the SDIE term obeys*

$$u_n = e^{-tB}u_0 + B^{-1}(I-e^{-tB})f - \frac{1}{2\varepsilon}B^{-1}(I-e^{-tB})\mathbf{1} + \lambda\sum_{k=1}^{n}e^{-(n-k)\tau B}\beta_k + \mathcal{O}(\tau). \tag{2.23}$$

Proof. Let $n\tau - t =: \eta_n = \mathcal{O}(\tau)$. Note that $e^{\eta_n X} = I + \mathcal{O}(\tau)$ for any bounded matrix X.

Note that $\mathcal{O}(\tau)$ is the same as $\mathcal{O}(\lambda)$, and also that, for bounded (in τ) invertible matrices X and Y with bounded (in τ) inverses, $X = Y + \mathcal{O}(\tau)$ if and only if $X^{-1} = Y^{-1} + \mathcal{O}(\tau)$.[5]

i. $||(1-\lambda)^{-n}e^{-n\tau A}u_0 - e^{-tB}u_0||_{\mathcal{V}} \leq ||(1-\lambda)^{-n}e^{-n\tau A} - e^{-tB}|| \cdot ||u_0||_{\mathcal{V}}$, so it suffices to consider $(1-\lambda)^{-n}e^{-n\tau A} - e^{-tB}$. Since $(1-\lambda)^{-n} = e^{n\lambda} + \mathcal{O}(\tau^2)$ we infer that

$$(1-\lambda)^{-n}e^{-n\tau A} = e^{t/\varepsilon}e^{\eta_n/\varepsilon}e^{-tA}e^{-\eta_n A} + \mathcal{O}(\tau^2) = e^{-tB} + \mathcal{O}(\tau).$$

ii. We note that

$$\sum_{k=1}^{n}(1-\lambda)^{-k}e^{-(k-1)\tau A} = ((1-\lambda)I - e^{-\tau A})^{-1}(I-(1-\lambda)^{-n}e^{-n\tau A}) = ((1-\lambda)I - e^{-\tau A})^{-1}(I-e^{-tB}) + \mathcal{O}(\tau).$$

We next consider each term of w individually. First, we seek to show that

$$((1-\lambda)I - e^{-\tau A})^{-1}(I-e^{-tB})A^{-1}(I-e^{-\tau A})f = B^{-1}(I-e^{-tB})f + \mathcal{O}(\tau)$$

[4]More precisely, we will say for real (matrix) valued g, $g(\tau,n) = \mathcal{O}(\tau)$ if and only if $limsup||g(\tau,n)/\tau|| < \infty$ as $(\tau,n)\to(0,\infty)$ in $\{(\rho,m)|\rho>0,\ m\rho - t\in[0,\rho)\}$ with the subspace topology induced by the standard topology on $(0,\infty)\times\mathbb{N}$.
[5]Suppose $X^{-1} = Y^{-1} + \mathcal{O}(\tau)$. Then $X = (Y^{-1}+\mathcal{O}(\tau))^{-1} = Y(I+\mathcal{O}(\tau))^{-1} = Y(I+\mathcal{O}(\tau)) = Y + \mathcal{O}(\tau)$.

so it suffices to show that

$$((1-\lambda)I - e^{-\tau A})^{-1}A^{-1}(I - e^{-\tau A}) = B^{-1} + \mathcal{O}(\tau).$$

This holds if and only if

$$\begin{aligned} B &= ((1-\lambda)I - e^{-\tau A})A(I - e^{-\tau A})^{-1} + \mathcal{O}(\tau) \\ &= A - \lambda A(I - e^{-\tau A})^{-1} + \mathcal{O}(\tau) \\ &= A - \varepsilon^{-1}\tau A\left(\tau A - \frac{1}{2}\tau^2 A^2 + - \cdots\right)^{-1} + \mathcal{O}(\tau) \\ &= A - \varepsilon^{-1}\left(I - \frac{1}{2}\tau A + - \cdots\right)^{-1} + \mathcal{O}(\tau) \\ &= A - \varepsilon^{-1}I + \mathcal{O}(\tau) \end{aligned}$$

and since $B = A - \varepsilon^{-1}I$ the result follows. Next we seek to show that

$$((1-\lambda)I - e^{-\tau A})^{-1}(I - e^{-tB})\frac{1}{2}\lambda\mathbf{1} = \frac{1}{2\varepsilon}B^{-1}(I - e^{-tB})\mathbf{1} + \mathcal{O}(\tau)$$

so it suffices to show that

$$((1-\lambda)I - e^{-\tau A})^{-1}\tau = B^{-1} + \mathcal{O}(\tau)$$

which holds if and only if

$$B = \tau^{-1}((1-\lambda)I - e^{-\tau A}) + \mathcal{O}(\tau) = A - \varepsilon^{-1}I + \mathcal{O}(\tau)$$

and since $B = A - \varepsilon^{-1}I$ the result follows.

iii. We follow [13, Proposition 5.1] and consider the difference

$$\begin{aligned} &\left|\left|\frac{\lambda}{1-\lambda}\sum_{k=1}^{n}(1-\lambda)^{-(n-k)}e^{-(n-k)\tau A}\beta_k - \lambda\sum_{k=1}^{n}e^{-(n-k)\tau B}\beta_k\right|\right|_{\mathcal{V}} \\ &= \lambda\left|\left|\sum_{k=1}^{n}\left((1-\lambda)^{-(n-k+1)} - e^{(n-k)\lambda}\right)e^{-(n-k)\tau A}\beta_k\right|\right|_{\mathcal{V}} \\ &= \lambda\left|\left|\sum_{\ell=0}^{n-1}\left((1-\lambda)^{-(\ell+1)} - e^{\ell\lambda}\right)e^{-\ell\tau A}\beta_k\right|\right|_{\mathcal{V}} \\ &\leq \lambda\sum_{\ell=0}^{n-1}\left((1-\lambda)^{-(\ell+1)} - e^{\ell\lambda}\right)\left|\left|e^{-\ell\tau A}\beta_k\right|\right|_{\mathcal{V}} \text{ as } (1-\lambda)^{-(\ell+1)} - e^{\ell\lambda} \geq 0 \\ &\leq \frac{1}{2}\lambda||\mathbf{1}||_{\mathcal{V}}\sum_{\ell=0}^{n-1}\left((1-\lambda)^{-(\ell+1)} - e^{\ell\lambda}\right) \text{ as } ||e^{-\ell\tau A}|| \leq 1 \text{ and } ||\beta_k||_{\mathcal{V}} \leq \frac{1}{2}||\mathbf{1}||_{\mathcal{V}} \\ &= \frac{1}{2}\lambda||\mathbf{1}||_{\mathcal{V}}\left(\frac{(1-\lambda)^{-n} - 1}{1-(1-\lambda)} - \frac{e^{n\lambda} - 1}{e^{\lambda} - 1}\right) \\ &= \frac{1}{2}||\mathbf{1}||_{\mathcal{V}}\left((1-\lambda)^{-n} - e^{n\lambda}\right) + \mathcal{O}(\tau) \text{ as } \lambda/(e^{\lambda} - 1) = 1 + \mathcal{O}(\tau) \\ &= \mathcal{O}(\tau) \end{aligned}$$

as desired. ∎

Following [13] we define the piecewise constant function $z_\tau : [0, \infty) \to \mathcal{V}$

$$z_\tau(s) := \begin{cases} e^{\tau B}\beta_1^{[\tau]}, & 0 \leq s \leq \tau \\ e^{k\tau B}\beta_k^{[\tau]}, & (k-1)\tau < s \leq k\tau \text{ for } k \in \mathbb{N} \end{cases}$$

and the function

$$\gamma_\tau(s) := e^{-sB} z_\tau(s) = \begin{cases} e^{(\tau-s)B}\beta_1^{[\tau]}, & 0 \le s \le \tau \\ e^{(k\tau-s)B}\beta_k^{[\tau]}, & (k-1)\tau < s \le k\tau \ \text{ for } \ k \in \mathbb{N} \end{cases}$$

following the bookkeeping notation of [13] of using the superscript $[\tau]$ to keep track of the time-step governing u_n and β_n. Next, we have weak convergence of z, up to a subsequence, as in [13].

Theorem 2.16. *For any sequence $\tau_n^{(0)} \downarrow 0$ with $\tau_n^{(0)} < \varepsilon$ for all n, there exists a function $z : [0,\infty) \to \mathcal{V}$ and a subsequence τ_n such that z_{τ_n} converges weakly to z in L^2_{loc}. It follows that:*

(A) *$\gamma_{\tau_n} \rightharpoonup \gamma$ in L^2_{loc}, where $\gamma(s) = e^{-sB} z$.*
(B) *For all $t \ge 0$,*

$$\int_0^t z_{\tau_n}(s)\, ds \to \int_0^t z(s)\, ds.$$

(C) *Passing to a further subsequence of τ_n, we have strong convergence of the Cesàro sums, that is, for all bounded $T \subseteq [0,\infty)$*

$$\frac{1}{N}\sum_{n=1}^{N} z_{\tau_n} \to z \quad \text{and} \quad \frac{1}{N}\sum_{n=1}^{N} \gamma_{\tau_n} \to \gamma \quad \text{in } \ L^2(T;\mathcal{V})$$

as $N \to \infty$.

Proof. Follows as in [13, Proposition 5.2] and [13, Corollary 5.3] *mutatis mutandis*. ∎

We thus infer convergence of the SDIE iterates as in [13]. Taking τ to zero along the sequence τ_n, we can define for all $t \ge 0$:

$$\hat{u}(t) := \lim_{n\to\infty, m=\lceil t/\tau_n \rceil} u_m^{[\tau_n]}. \tag{2.24}$$

By the above discussion, we can rewrite this as:

$$\begin{aligned} \hat{u}(t) &= e^{-tB}u_0 + B^{-1}(I - e^{-tB})f - \frac{1}{2\varepsilon}B^{-1}(I - e^{-tB})\mathbf{1} + \lim_{n\to\infty}\frac{\tau_n}{\varepsilon}\sum_{k=1}^{m} e^{-(m-k)\tau_n B}\beta_k^{[\tau_n]} \\ &= e^{-tB}u_0 + B^{-1}(I - e^{-tB})f - \frac{1}{2\varepsilon}B^{-1}(I - e^{-tB})\mathbf{1} + \frac{1}{\varepsilon}\lim_{n\to\infty} e^{-m\tau_n B}\int_0^{m\tau_n} z_{\tau_n}(s)\, ds. \end{aligned}$$

Next, note that $m\tau_n = \tau_n\lceil t/\tau_n \rceil =: t + \eta_n$ where $\eta_n \in [0, \tau_n)$. Therefore

$$\begin{aligned} \lim_{n\to\infty} e^{-m\tau_n B}\int_0^{m\tau_n} z_{\tau_n}(s)\, ds &= \lim_{n\to\infty} e^{-\eta_n B}e^{-tB}\int_0^t z_{\tau_n}(s)\, ds + e^{-\eta_n B}e^{-tB}\int_t^{t+\eta_n} z_{\tau_n}(s)\, ds \\ &= \lim_{n\to\infty} e^{-tB}\int_0^t z_{\tau_n}(s)\, ds + e^{-tB}\int_t^{t+\eta_n} z_{\tau_n}(s)\, ds \ \text{ as } \ e^{-\eta_n B} = I + \mathcal{O}(\tau_n) \\ &= \lim_{n\to\infty} e^{-tB}\int_0^t z_{\tau_n}(s)\, ds \ \text{ as } \ z_{\tau_n}(s) \ \text{ is bounded on } \ [t, t + \max_{n'}\eta_{n'}] \ \text{ uniformly in } \ n \\ &= e^{-tB}\int_0^t z(s)\, ds \ \text{ by Theorem 2.16(B).} \end{aligned}$$

So we have that

$$\hat{u}(t) = e^{-tB}u_0 + B^{-1}(I - e^{-tB})f - \frac{1}{2\varepsilon}B^{-1}(I - e^{-tB})\mathbf{1} + \frac{1}{\varepsilon}\int_0^t e^{-(t-s)B}\gamma(s)\,ds. \tag{2.25}$$

Note the similarity between (2.25) and the explicit form for ACE solutions (2.12). Thus, to prove that $\hat{u}$ is a solution to (2.8) it suffices to show that:

(a) $\hat{u}(t) \in \mathcal{V}_{[0,1]}$ for all $t \geq 0$,
(b) $\hat{u} \in H^1_{loc}([0,\infty); \mathcal{V}) \cap C^0([0,\infty); \mathcal{V})$, and
(c) $\gamma(t) \in \mathcal{B}(\hat{u}(t))$ for a.e. $t \geq 0$.

These results follow as in [13]. Item (a) follows immediately from the fact that for all n, $u_m^{[\tau_n]} \in \mathcal{V}_{[0,1]}$.

Towards (b), note that each term in (2.25) except for the integral is $C^\infty([0,\infty); \mathcal{V})$, and that $\int_0^t z(s)\,ds$ is continuous since z is locally bounded as a weak limit of locally uniformly bounded functions. Hence $\hat{u}$ is continuous. By (a), $\hat{u}$ is bounded so is locally L^2. Finally, it is easy to check that $\hat{u}$ has weak derivative

$$\frac{d\hat{u}}{dt} = -Be^{-tB}u_0 + e^{-tB}\left(f - \frac{1}{2\varepsilon}\mathbf{1}\right) + \frac{1}{\varepsilon}e^{-tB}z(t) - \frac{1}{\varepsilon}Be^{-tB}\int_0^t z(s)\,ds.$$

This is locally L^2 since (for T a bounded interval) B and e^{-tB} are bounded operators from $L^2(T; \mathcal{V})$ to $L^2(T; \mathcal{V})$, z is a weak limit of locally L^2 functions so is locally L^2, and $\int_0^t z(s)\,ds$ is continuous so is locally bounded.

Towards (c), by Theorem 2.16(C) and [13, p. 25] *mutatis mutandis* there is a sequence $N_k \to \infty$, independent of t, with

$$\gamma(t) = \lim_{k\to\infty} \frac{1}{N_k}\sum_{n=1}^{N_k} \beta_m^{[\tau_n]}$$

for a.e. $t \geq 0$. Then, at each such t, $\gamma(t) \in \mathcal{B}(\hat{u}(t))$ follows from $u_m^{[\tau_n]} \to \hat{u}(t)$ and $\beta_m^{[\tau_n]} \in \mathcal{B}(u_m^{[\tau_n]})$ as in [13, p. 25]. Hence we can infer the following convergence theorem.

Theorem 2.17 (Cf. [13, Theorem 5.4]). *For any given $u_0 \in \mathcal{V}_{[0,1]}$, $\varepsilon > 0$ (with $\varepsilon^{-1} \notin \sigma(A)$) and $\tau_n \downarrow 0$, there exists a subsequence τ_n' of τ_n with $\tau_n' < \varepsilon$ for all n, along which the SDIE iterates $(u_m^{[\tau_n']}, \beta_m^{[\tau_n']})$ given by (2.14) with initial state u_0 converge to the ACE solution with initial condition u_0 in the following sense: For each $t \geq 0$, as $n \to \infty$ and $m = \lceil t/\tau_n' \rceil$, $u_m^{[\tau_n']} \to \hat{u}(t)$, and there is a sequence $N_k \to \infty$ such that for almost every $t \geq 0$, $\frac{1}{N_k}\sum_{n=1}^{N_k} \beta_m^{[\tau_n']} \to \gamma(t)$, where $(\hat{u}, \gamma)$ is the solution to (2.8) with $\hat{u}(0) = u_0$.*

Corollary 2.18. *Let $u_0 \in \mathcal{V}_{[0,1]}$, $\varepsilon > 0$, $\varepsilon^{-1} \notin \sigma(A)$, and $\tau_n \downarrow 0$ with $\tau_n < \varepsilon$ for all n. Then for each $t \geq 0$, as $n \to \infty$, $u_{\lceil t/\tau_n \rceil}^{[\tau_n]} \to \hat{u}(t)$.*

Proof. Let $x_n : t \mapsto u_{\lceil t/\tau_n \rceil}^{[\tau_n]}$ and let τ_{n_k} be any subsequence of τ_n. Then by the theorem there is a subsubsequence $\tau_{n_{k_l}}$ such that $x_{n_{k_l}} \to \tilde{u}$ pointwise where $\tilde{u}$ is a solution to (2.8) with initial condition $\tilde{u}(0) = u_0$. By Theorem 2.7(c) such solutions are unique, so $\tilde{u} = \hat{u}$. Thus there exists x (in particular, $x = \hat{u}$) such that every subsequence of x_n has a convergent subsubsequence with limit x. It follows by a standard fact of topological spaces that $x_n \to \hat{u}$ pointwise as $n \to \infty$.[6] ∎

Finally, we follow [13] to use Theorem 2.17 to deduce that the Ginzburg-Landau energy monotonically decreases along the ACE trajectories by considering the Lyapunov functional H defined in (2.19). We also deduce well-posedness of the ACE.

Proposition 2.19 (Cf. [13, Proposition 5.6]). *Let $H_\tau(u) := \frac{1}{2\tau}H(u)$. Then for $u \in \mathcal{V}_{[0,1]}$*

$$H_\tau(u) = GL_{\varepsilon,\mu,\tilde{f}}(u) - \frac{1}{2}\langle \tilde{f}, M\tilde{f}\rangle_{\mathcal{V}} + \frac{1}{2}\tau\langle u, Q_\tau(u - 2A^{-1}f)\rangle_{\mathcal{V}},$$

[6] Suppose $x_n \not\to x$. Then there exists U which is open in the topology of pointwise convergence such that $x \in U$ and infinitely many $x_n \notin U$. Choose x_{n_k} such that for all k, $x_{n_k} \notin U$. This subsequence has no further subsubsequence converging to x.

where $Q_\tau := \tau^{-2}(I - \tau A - e^{-\tau A})$. *Hence* $H_\tau + \frac{1}{2}\langle \tilde{f}, M\tilde{f}\rangle_{\mathcal{V}} \to GL_{\varepsilon,\mu,\tilde{f}}$ *uniformly on* $\mathcal{V}_{[0,1]}$ as $\tau \to 0$, *and furthermore if* $u_\tau \to u$ *in* $\mathcal{V}_{[0,1]}$ *then* $H_\tau(u_\tau) + \frac{1}{2}\langle \tilde{f}, M\tilde{f}\rangle_{\mathcal{V}} \to GL_{\varepsilon,\mu,\tilde{f}}(u)$.

Proof. Expanding and collecting terms in (2.5), we find that for $u \in \mathcal{V}_{[0,1]}$

$$GL_{\varepsilon,\mu,\tilde{f}}(u) = \frac{1}{2\varepsilon}\langle u, \mathbf{1} - u\rangle_{\mathcal{V}} + \frac{1}{2}\langle u, Au - 2f\rangle_{\mathcal{V}} + \frac{1}{2}\langle \tilde{f}, M\tilde{f}\rangle_{\mathcal{V}}.$$

Then by (2.19) and recalling that $\lambda := \tau/\varepsilon$

$$\begin{aligned} H_\tau(u) &= \frac{1}{2\varepsilon}\langle u, \mathbf{1} - u\rangle_{\mathcal{V}} + \frac{1}{2\tau}\langle u, (I - e^{-\tau A})u - 2A^{-1}(I - e^{-\tau A})f\rangle_{\mathcal{V}} \\ &= \frac{1}{2\varepsilon}\langle u, \mathbf{1} - u\rangle_{\mathcal{V}} + \frac{1}{2\tau}\langle u, (\tau A + \tau^2 Q_\tau)u - 2A^{-1}(\tau A + \tau^2 Q_\tau)f\rangle_{\mathcal{V}} \\ &= \frac{1}{2\varepsilon}\langle u, \mathbf{1} - u\rangle_{\mathcal{V}} + \frac{1}{2}\langle u, Au - 2f\rangle_{\mathcal{V}} + \frac{1}{2}\tau\langle u, Q_\tau(u - 2A^{-1}f)\rangle_{\mathcal{V}}. \end{aligned}$$

To show the uniform convergence, note that $||u||_{\mathcal{V}}$ and $||u - 2A^{-1}f||_{\mathcal{V}}$ are uniformly bounded in u for $u \in \mathcal{V}_{[0,1]}$. Thus it suffices to prove that $||Q_\tau||$ is uniformly bounded in τ. But Q_τ is self-adjoint, and if ξ_k is an eigenvalue of A then Q_τ has corresponding eigenvalue $\tau^{-2}(1 - \tau\xi_k - e^{-\tau\xi_k}) \in [-\frac{1}{2}\xi_k^2, 0]$, so $||Q_\tau|| \le \frac{1}{2}||A||^2$. Finally, it suffices to show that $H_\tau(u_\tau) - H_\tau(u) \to 0$, since

$$H_\tau(u_\tau) + \frac{1}{2}\langle \tilde{f}, M\tilde{f}\rangle_{\mathcal{V}} - GL_{\varepsilon,\mu,\tilde{f}}(u) = H_\tau(u_\tau) - H_\tau(u) + H_\tau(u) + \frac{1}{2}\langle \tilde{f}, M\tilde{f}\rangle_{\mathcal{V}} - GL_{\varepsilon,\mu,\tilde{f}}(u).$$

Then by the above expression for H_τ

$$H_\tau(u_\tau) - H_\tau(u) = \frac{1}{2}\left\langle u_\tau - u, \frac{1}{\varepsilon}(1 - u_\tau - u) + (A + \tau Q_\tau)(u_\tau + u) - 2(I + \tau Q_\tau A^{-1})f\right\rangle_{\mathcal{V}} \to 0$$

since the right-hand entry in the inner product is bounded uniformly in τ. ∎

Theorem 2.20 (Cf. [13, Theorem 5.7, Remark 5.8]). *Suppose and* $\varepsilon^{-1} \notin \sigma(A)$. *Then the ACE trajectory* u *defined by Definition* 2.6 *has* $GL_{\varepsilon,\mu,\tilde{f}}(u(t))$ *monotonically decreasing in* t. *More precisely: for all* $t > s \ge 0$,

$$GL_{\varepsilon,\mu,\tilde{f}}(u(s)) - GL_{\varepsilon,\mu,\tilde{f}}(u(t)) \ge \frac{1}{2(t-s)}\,||u(s) - u(t)||_{\mathcal{V}}^2\,. \tag{2.26}$$

Furthermore, this entails an explicit $C^{0,1/2}$ *condition for* u

$$||u(s) - u(t)||_{\mathcal{V}} \le \sqrt{|t-s|}\sqrt{2GL_{\varepsilon,\mu,\tilde{f}}(u(0))}. \tag{2.27}$$

Proof. The proof is identical to that in [13, Theorem 5.7] and [13, Remark 5.8]. ∎

Theorem 2.21 (Cf. [13, Theorem 3.11]). *Let* $u_0, v_0 \in \mathcal{V}_{[0,1]}$ *define ACE trajectories* u, v *by Definition* 2.6, *and suppose* $\varepsilon^{-1} \notin \sigma(A)$. *Then, if* $\xi_1 := \min\sigma(A)$, *then*

$$||u(t) - v(t)||_{\mathcal{V}} \le e^{-\xi_1 t}e^{t/\varepsilon}||u_0 - v_0||_{\mathcal{V}}. \tag{2.28}$$

Proof. Fix $t \ge 0$ and let $m := \lceil t/\tau_n \rceil$. By Corollary 2.18, we take $\tau_n \downarrow 0$ such that $u_m^{[\tau_n]} \to u(t)$ and $v_m^{[\tau_n]} \to v(t)$ as $n \to \infty$. Then by (2.18):

$$||u_m^{[\tau_n]} - v_m^{[\tau_n]}||_{\mathcal{V}} \le e^{-m\xi_1\tau_n}(1 - \tau_n/\varepsilon)^{-m}||u_0 - v_0||_{\mathcal{V}}$$

and taking $n \to \infty$ gives (2.28). ∎

3 | THE SDIE SCHEME AS A CLASSIFICATION ALGORITHM

As was noted in the introduction, trajectories of the ACE and of the MBO scheme on graphs can be deployed as classification algorithms, as was originally done in work by Bertozzi and coauthors in [6, 32]. In the above, we have shown that the SDIE scheme (2.14) introduced in [13] generalizes the MBO scheme into a family of schemes, all of the same computational cost as the MBO scheme, and which as $\tau \downarrow 0$ become increasingly more accurate approximations of the ACE trajectories. In the remainder of this paper, we will investigate whether the use of these schemes can significantly improve on the use of the MBO scheme to segment the "two cows" images from [6, 32]. We will also discuss other potential improvements to the method of [32].

3.1 | Groundwork

In this section, we will describe the framework for applying graph dynamics to classification problems, following [6, 32].

The individuals that we seek to classify we will denote as a set V, upon which we have some information $x : V \to \mathbb{R}^q$. For example, in image segmentation V is the pixels of the image, and x is the grayscale/RGB/etc values at each pixel. Furthermore, we have *labeled reference data* which we shall denote as $Z \subseteq V$ and binary reference labels $\tilde{f}$ supported on Z. Supported on Z we have our fidelity parameter $\mu \in \mathcal{V}_{[0,\infty)} \setminus \{\mathbf{0}\}$, and we recall the notation $M := diag(\mu)$ and $f := M\tilde{f}$ (recall that the operator diag sends a vector to the diagonal matrix with that vector as diagonal, and also *vice versa*).

To build our graph, we first construct *feature vectors* $z : V \to \mathbb{R}^\ell$. The philosophy behind these is that we want vertices which are "similar" (and hence should be similarly labeled) to have feature vectors that are "close together." What this means in practice will depend on the application, for example, [41] incorporates texture into the features and [6, 15] give other options.

Next, we construct the weights on the graph by deciding on the edge set E (eg, $E = V^2 \setminus \{(i,i) \mid i \in V\}$) and for each $ij \in E$ computing $\omega_{ij} := \Omega(z_i, z_j)$ (and for $ij \notin E$, $\omega_{ij} := 0$). There are a number of standard choices for the similarity function Ω, see for example [6, 12, 43, 45]. The similarity function we will use in this paper is the Gaussian function:

$$\Omega(z, w) := e^{-\frac{||z-w||^2}{\ell\sigma^2}}.$$

Finally, from these weights we compute the graph Laplacian so that we can employ the graph ODEs discussed in the previous sections. In particular, we compute the *normalized* (a.k.a. random walk) graph Laplacian, that is, we will henceforth take $r = 1$ and so $\Delta = I - D^{-1}\omega$. We will also consider the *symmetric normalized* Laplacian $\Delta_s := I - D^{-1/2}\omega D^{-1/2}$, though this does not fit into the above framework (extending the above theory to the symmetric normalized Laplacian case is a topic for future research). This normalization matters because, as discussed in [6], the segmentation properties of diffusion-based graph dynamics are linked to the segmentation properties of the eigenvectors of the corresponding Laplacian. As shown in [6, Figure 2.1], normalization vastly improves these segmentation properties. As that figure looked at the symmetric normalized Laplacian, we include Figure 2 to show the difference between the symmetric normalized and the random walk Laplacian.

3.2 | The basic classification algorithm

For some time step $0 < \tau \leq \varepsilon$ note that

$$S_\tau u = e^{-\tau A}u + b,$$

where $b := A^{-1}(I - e^{-\tau A})f$ is independent of u.

1. **Input:** Vector $x : V \to \mathbb{R}^q$, reference data Z, and labels f supported on Z.
2. Convert x into feature vectors $z : V \to \mathbb{R}^\ell$.
3. Build a weight matrix $\omega = (\omega_{ij})$ on V^2 via $\omega_{ij} := \Omega(z_i, z_j)$.
4. Compute A and therefore b.

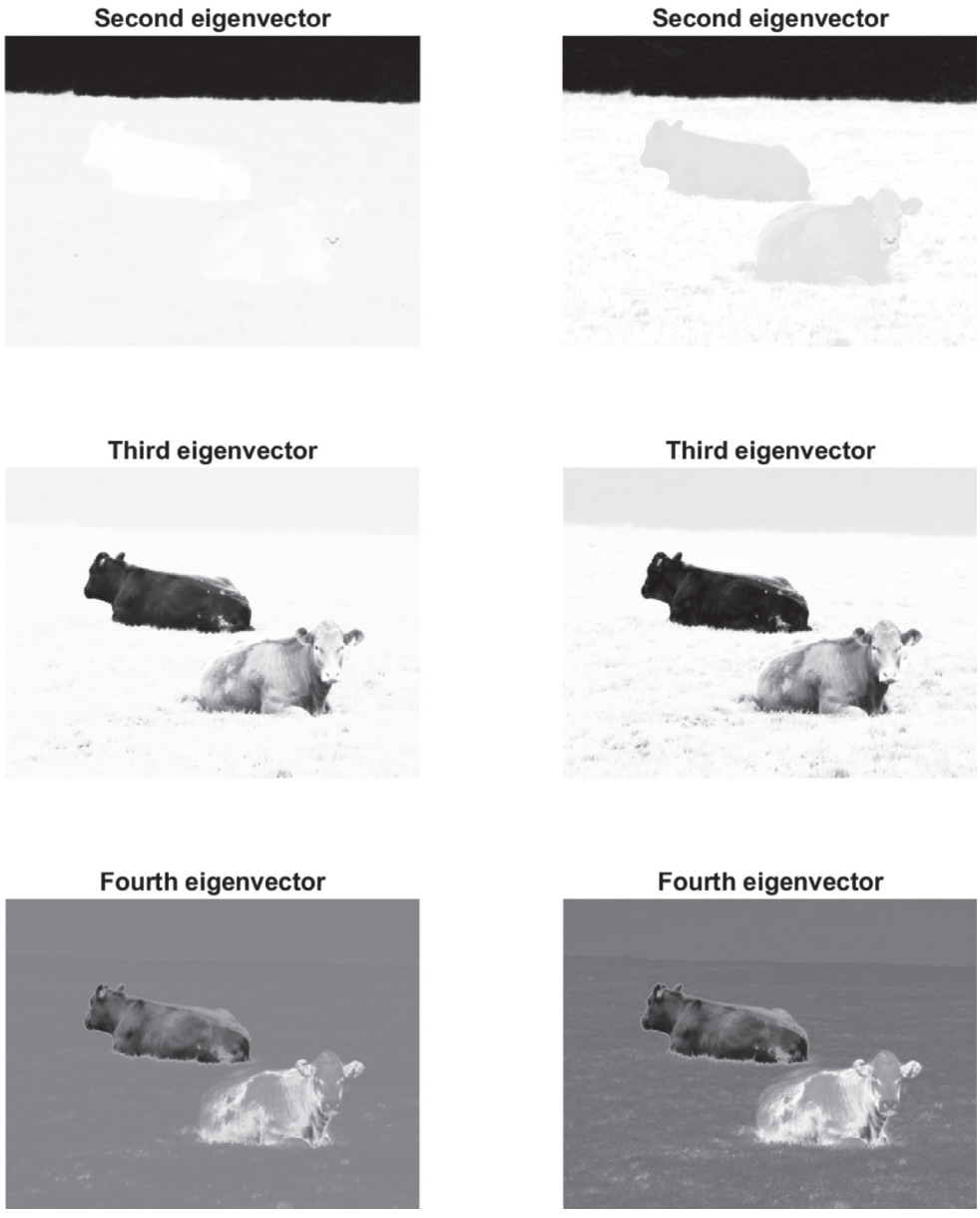


FIGURE 2 Second, third, and fourth eigenvectors of the random walk Laplacian (left) and symmetric normalized Laplacian (right) for the graph built on one of the "two cows" images from Section 4, computed using Algorithm 1

5. From some initial condition u_0, compute the SDIE sequence u_n until it meets a stopping condition at some $n = N$.
6. **Output:** u_N thresholded to lie in $\mathcal{V}_{\{0,1\}}$, that is, we output $\Theta(u_N)$ where Θ is as in (1.1).

Unfortunately, as written this algorithm cannot be feasibly run. The chief obstacle is that in many applications A is too large to store in memory, yet we need to quickly compute $e^{-\tau A}u$, potentially a large number of times. We also need to compute b accurately . Moreover, in general A does not have low numerical rank, so it cannot be well approximated by a low-rank matrix. In the rest of this section we describe our modifications to this basic algorithm that make it computationally efficient.

3.3 | Matrix compression and approximate SVDs

We will need to compress Δ into something we can store in memory. Following [6, 32], we employ the Nyström extension [21, 36]. We choose $K \ll |V|$ to be the rank to which we will compress Δ, and choose nonempty Nyström interpolation sets $X_1 \subseteq V \setminus Z$ and $X_2 \subseteq Z$ at random such that $|X|{=}K$ where $X := X_1 \cup X_2$. We write $|X_1| =: K_1$ and $|X_2| =: K_2$. Then using the function $ij \mapsto \omega_{ij}$ we compute $\omega_{VX} := \omega(V, X)$ (ie, $\omega_{VX} := (\omega_{ij})_{i\in V, j\in X}$) and $\omega_{XX} := \omega(X, X)$ and then the Nyström extension

is the approximation:

$$\omega \approx \omega_{VX}\omega_{XX}^{-1}\omega_{VX}^{T}.$$

Note that this avoids having to compute the full matrix ω which in many applications is too large to store in memory. We next compute an approximation for the degree vector d and degree matrix $D \coloneqq diag(d)$ of our graph

$$d \approx \hat{d} \coloneqq \omega_{VX}\omega_{XX}^{-1}\omega_{VX}^{T}\mathbf{1}, \quad D \approx \hat{D} \coloneqq diag\left(\hat{d}\right).$$

We thus approximately normalize ω

$$\tilde{\omega} \coloneqq D^{-1/2}\omega D^{-1/2} \approx \hat{D}^{-1/2}\omega_{VX}\omega_{XX}^{-1}\omega_{VX}^{T}\hat{D}^{-1/2} = \tilde{\omega}_{VX}\omega_{XX}^{-1}\tilde{\omega}_{VX}^{T},$$

where $\tilde{\omega}_{VX} \coloneqq \hat{D}^{-1/2}\omega_{VX}$.

Following [6, 32], we next compute an approximate eigendecomposition of $\tilde{\omega}$. We here diverge from the method of [6, 32]. The method used in those papers requires taking the matrix square root of ω_{XX}, but unless ω_{XX} is the zero matrix it will not be positive semi-definite.[7] While this clearly does not prevent the method of [6, 32] from working in practice, it is a potential source of error and we found it conceptually troubling. We here present an improved method, adapted from the method from [3] for computing a SVD from an adaptive cross approximation (ACA) (see [3] for a definition of ACA).

First, we compute the thin QR decomposition (see [25, Theorem 5.2.2])

$$\tilde{\omega}_{VX} = QR,$$

where $Q \in \mathbb{R}^{|V|\times K}$ is orthonormal and $R \in \mathbb{R}^{K\times K}$ is upper triangular. Next, we compute the eigendecomposition

$$R\omega_{XX}^{-1}R^{T} = \Phi\Sigma\Phi^{T},$$

where $\Phi \in \mathbb{R}^{K\times K}$ is orthogonal and $\Sigma \in \mathbb{R}^{K\times K}$ is diagonal. It follows that $\tilde{\omega}$ has approximate eigendecomposition:

$$\tilde{\omega} \approx Q\Phi\Sigma\Phi^{T}Q^{T} = U_s\Sigma U_s^{T},$$

where $U_s \coloneqq Q\Phi$ is orthonormal. This gives an approximate eigendecomposition of the symmetric normalized Laplacian

$$\Delta_s = I - \tilde{\omega} \approx U_s(I_K - \Sigma)U_s^{T} = U_s\Lambda U_s^{T},$$

where I_K is the $K \times K$ identity matrix and $\Lambda \coloneqq I_K - \Sigma$, and so we get an approximate SVD of the random walk Laplacian

$$\Delta = D^{-1/2}\Delta_s D^{1/2} \approx U_1\Lambda U_2^{T},$$

where $U_1 \coloneqq \hat{D}^{-1/2}U_s$ and $U_2 \coloneqq \hat{D}^{1/2}U_s$. As in [3], it is easy to see that this approach is $\mathcal{O}(K|V|)$ in space and $\mathcal{O}(K^2|V| + K^3)$ in time. We summarize this all as Algorithm 1.

3.3.1 | Numerical assessment of the matrix compression method

We consider the accuracy of our Nyström-QR approach for the compression of the symmetric normalized Laplacian[8] Δ_s built on the simple image in Figure 3, containing $|V| = 6400$ pixels, which is small enough for us to compute the true value of Δ_s to high accuracy. For $K \in \{50, 100, \cdots, 500\}$, we compare the rank K approximation $U_s\Lambda U_s^{T}$ with the true Δ_s in terms of the relative Frobenius distance, that is, $||U_s\Lambda U_s^{T} - \Delta_s||_F/||\Delta_s||_F$. Moreover, we compare these

[7] It is easy to see that nonzero ω_{XX} has negative eigenvalues, as it has zero trace.

[8] The case of the random walk Laplacian is similar (due to the similarity of the matrices) except for a small additional error incurred by the approximation of D.

Algorithm 1. QR decomposition-based Nyström method for computing an approximate SVD of Δ, inspired by [3].

function NYSTRÖMQR($ij \mapsto \omega_{ij}, V, Z, K$) ▷ Computes U_1, Λ, and U_2; $\Delta \approx U_1 \Lambda U_2^T$ is an approximate SVD of rank K
  $X_1 = \texttt{random\_subset}(V \setminus Z, K_1)$ ▷ A random subset of $V \setminus Z$ of size K_1
  $X_2 = \texttt{random\_subset}(Z, K_2)$ ▷ A random subset of Z of size K_2, note that $K_1 + K_2 = K$
  $X = X_1 \cup X_2$
  $\omega_{XX} = \omega(X, X)$
  $\omega_{VX} = \omega(V, X)$
  $\hat{d} = \omega_{VX} \left(\omega_{XX}^{-1} \left(\omega_{VX}^T \mathbf{1} \right) \right)$
  $\tilde{\omega}_{VX} = \hat{d}^{-1/2} \odot \omega_{VX}$ ▷ Applying $\odot$ columnwise, that is, $(\tilde{\omega}_{VX})_{ij} = \hat{d}_i^{-1/2} (\omega_{VX})_{ij}$
  $[Q, R] = \texttt{thin\_qr}(\tilde{\omega}_{VX})$ ▷ Computes thin QR decomposition $\tilde{\omega}_{VX} = QR$
  $S = R \omega_{XX}^{-1} R^T$
  $S = (S + S^T)/2$ ▷ Corrects symmetry-breaking computational errors
  $[\Phi, \Sigma] = \texttt{eig}(S)$ ▷ Computes eigendecomposition $S = \Phi \Sigma \Phi^T$
  $\Lambda = I_K - \Sigma$
  $U_s = Q\Phi$ ▷ Note that $\Delta_s \approx U_s \Lambda U_s^T$, so to return the decomposition of Δ_s terminate here
  $U_1 = \hat{d}^{-1/2} \odot U_s$ ▷ I.e. $(U_1)_{ij} = \hat{d}_i^{-1/2} (U_s)_{ij}$
  $U_2 = \hat{d}^{1/2} \odot U_s$ ▷ I.e. $(U_2)_{ij} = \hat{d}_i^{1/2} (U_s)_{ij}$
  return U_1, Λ, U_2
end function

FIGURE 3 The 80×80 image on which the Laplacian Δ_s is constructed to test the low-rank approximations

errors to the errors incurred by other low-rank approximations of Δ_s, namely the Nyström method used in [6, 32], the Halko-Martinsson-Tropp (HMT) method[9] [26], and the optimal rank K approximation of Δ_s with respect to $\|\cdot\|_F$, which (by the Eckart-Young theorem [19]; see also [25, Theorem 2.4.8]) is obtained by setting all but the K leading singular values of Δ_s to 0. In addition to the methods' accuracy, we measure their complexity via the runtimes of MATLAB R2019a implementations of them on the set-up as in Section 4.2.

We report the relative Frobenius distance in Figure 4. As the Nyström (and HMT) methods are randomized, we repeat the experiments 1000 times and plot the mean error in the far-left figure and the standard deviations in the center-right figure. To expose the difference between the methods for $K \geq 100$, we plot the percentage increase of the other errors above the SVD error (ie, $[\text{error}/\text{error}_{\text{SVD}} - 1] \times 100\%$) and show this percentage in the center-left figure. In the far-right figure, we compare the complexity of the algorithms in terms of their average runtime. The SVD timing is constant in K as we always computed a full SVD and kept the largest K singular values.

We observe that the Nyström-QR method outperforms the Nyström method from [6, 32]: it is faster, more accurate, and is stable for small K. In terms of accuracy, the Nyström-QR error is only about 0.012% larger than the optimal low-rank error. Notably, this additional error is (almost) constant, indicating that the Nyström-QR method and the SVD converge at a similar rate. The Nyström-QR randomness has hardly any effect on the error; the standard deviation of the relative

[9]The HMT results serve only to give an additional benchmark for the Nyström methods: HMT requires matrix-vector-products with Δ_s, which was infeasible for us in applications. However, as we were finalizing this paper we were made aware of the recent work of [5], which may make computing the HMT-SVD of Δ_s feasible.

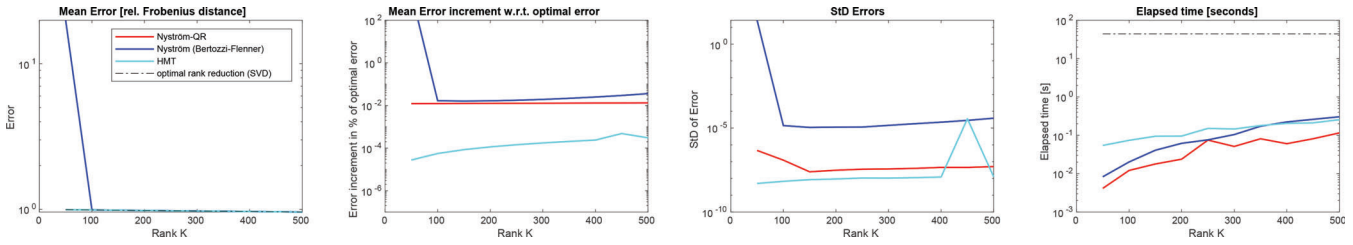


FIGURE 4 Error of the low-rank approximations of Δ_s in terms of their relative Frobenius distance to the true Δ_s (far-left), percentage increase above the optimal error reached with the SVD (center-left), and standard deviations (StD) of the errors (center-right). Timings of the methods in seconds (far-right). In the far-left figure the red and cyan lines are both plotted but cannot be distinguished from each other by the eye

error ranges from 8.87×10^{-7} ($K=50$) to 5.00×10^{-8} ($K=500$). By contrast, for the Nyström method from [6, 32] we see much more random variation. The bump in the HMT error at $K=450$ is an outlier that is present (though not always for the same K) in many repeats of this experiment.

3.4 | Implementing the SDIE scheme: a Strang formula method

To compute the iterates of our SDIE scheme, we will need to compute an approximation for $\mathcal{S}_\tau u_n$. In [32], at each iteration $\mathcal{S}_\tau u_n$ was approximated via a semi-implicit Euler method, which therefore incurred a linear (in the time step of the Euler method, that is, δt in the below notation) error in both the $e^{-\tau A}u_n$ and b terms (plus a spectrum truncation error). In Appendix A1 we show that the method from [32] works by approximating a Lie product formula approximation (see [27, Theorem 2.11]) of $e^{-\tau A}$, therefore we propose as an improvement a scheme that directly employs the superior Strang formula[10] to approximate $e^{-\tau A}u_n$—with quadratic error (plus a spectrum truncation error). We also consider potential improvements of the accuracy of computing b: by expressing b as an integral and using quadrature methods[11]; by expressing b as a solution to the ODE from (2.1) with initial condition $\mathbf{0}$, and using the Euler method from [32] with a very small time step (or a higher-order ODE solver)[12]; or by computing the closed form solution for b directly using the Woodbury identity. We therefore improve on the accuracy of computing $\mathcal{S}_\tau u$ at low cost.

The Strang formula for matrix exponentials [39] is given, for $k>0$ a parameter and $\mathcal{O}$ relative to the limit $k\to\infty$, by

$$e^{X+Y}=(e^{\frac{1}{2}Y/k}e^{X/k}e^{\frac{1}{2}Y/k})^k+\mathcal{O}(k^{-2}).$$

Given $\Delta\approx U_1\Lambda U_2^T$ as in Algorithm 1 (the case for Δ_s is likewise) for any $u\in\mathcal{V}$ we compute (writing $\delta t:=\tau/k$)

$$\begin{aligned}e^{-\tau A}u&=\left(e^{-\frac{1}{2}\tau/kM}e^{-\tau/k\Delta}e^{-\frac{1}{2}\tau/kM}\right)^k u+\mathcal{O}(k^{-2})\\&=\left(e^{-\frac{1}{2}\delta tM}\left(I+U_1(e^{-\delta t\Lambda}-I_K)U_2^T\right)e^{-\frac{1}{2}\delta tM}\right)^k u+E+\mathcal{O}(\delta t^2)\\&=v_k+E+\mathcal{O}(\delta t^2),\end{aligned}\tag{3.1}$$

where E is a spectrum truncation error[13] and v_k is defined by $v_0:=u$, and v_r for $r\in\{1,\ldots,k\}$ is defined iteratively by

$$\begin{aligned}v_{r+1}&=e^{-\delta tM}v_r+e^{-\frac{1}{2}\delta tM}U_1(e^{-\delta t\Lambda}-I_K)U_2^Te^{-\frac{1}{2}\delta tM}v_r\\&=a_1(\delta t)\odot v_r+a_3(\delta t)\odot\left(U_1\left(a_2(\delta t)\odot\left(U_2^T\left(a_3(\delta t)\odot v_r\right)\right)\right)\right)\\&=:S(\delta t)v_r,\end{aligned}\tag{3.2}$$

[10] We owe the suggestion to use this formula to Arieh Iserles, who also suggested to us the Yoshida method that we consider below.
[11] We again thank Arieh Iserles for also making this suggestion.
[12] We can afford to do this for b, but not generally for the $\mathcal{S}_\tau u_n$, because b only needs to be computed once rather than at each n.
[13] Specifically, $E=(e^{-\frac{1}{2}\delta tM}e^{-\delta t\Delta}e^{-\frac{1}{2}\delta tM})^k u-(e^{-\frac{1}{2}\delta tM}(I+U_1(e^{-\delta t\Lambda}-I_K)U_2^T)e^{-\frac{1}{2}\delta tM})^k u$ is incurred by the spectrum truncation in the middle of the latter term.

where $\odot$ is the Hadamard (ie, elementwise) product, $a_1(\delta t) \coloneqq \exp(-\delta t \mu)$, $a_2(\delta t) \coloneqq \exp(-\delta\, t\, diag(\Lambda)) - \mathbf{1}_K$, and $a_3(\delta t) \coloneqq \exp(-\frac{1}{2}\delta t \mu)$ is the elementwise square root of $a_1(\delta t)$ (where exp is applied elementwise, and $\mathbf{1}_K$ is the vector of K ones). In Figure 6, we verify on a simple image that this method has quadratic error (plus a spectrum truncation error) and outperforms the [32] Euler method. Furthermore (3.2) is as fast as (A2) (ie, a step of the [32] Euler method). This is because by defining $\tilde{a}_1 \coloneqq \mathbf{1} - \delta t \mu$ and $\tilde{a}_2 \coloneqq (\mathbf{1}_K + \delta\, t\, diag(\Lambda))^{-1}$ (applying the reciprocation elementwise), we can rewrite (A2) as

$$v_{r+1} = U_1 \left(\tilde{a}_2 \odot \left(U_2^T \left(\tilde{a}_1 \odot v_r + \mu \delta t f \right) \right) \right)$$

and so both (3.2) and (A2) involve two $\mathcal{O}(NK)$ matrix multiplications and the vectors in (3.2) and (A2) and are at most N-dimensional, hence the Hadamard products in (3.2) and (A2) are all at most $\mathcal{O}(N)$ and so are not rate-limiting.

At the cost of extra $\mathcal{O}(NK)$ matrix multiplications, one can employ the method of Yoshida [44] to increase the order of the (non-spectrum-truncation) error by 2. If we set $\alpha_0 \coloneqq -\sqrt[3]{2}/(2 - \sqrt[3]{2})$ and $\alpha_1 \coloneqq 1/(2 - \sqrt[3]{2})$ then we can define the map

$$Y(\delta t) \coloneqq S(\alpha_1 \delta t) \circ S(\alpha_0 \delta t) \circ S(\alpha_1 \delta t)$$

which gives $Y^k(\delta t)u = e^{-\tau A}u + \mathcal{O}(\delta t^4)$ plus a spectrum truncation error.[14] However, as can be seen in Figure 6C,D, the spectrum truncation error can make negligible any gain from using the Yoshida method over the Strang formula.

It remains to compute an approximation for $b \coloneqq A^{-1}(I - e^{-\tau A})f$. It is easy to show that b can be rewritten as the integral

$$b = \int_0^\tau e^{-tA} f \, dt$$

which we can approximate via a quadrature, for example, applying respectively the trapezium, midpoint, or Simpson's rules we get

$$b = \frac{1}{2}\tau(I + e^{-\tau A})f + \mathcal{O}(\tau^3) = \tau e^{-\frac{1}{2}\tau A} f + \mathcal{O}(\tau^3) = \frac{1}{6}\tau(I + 4e^{-\frac{1}{2}\tau A} + e^{-\tau A})f + \mathcal{O}(\tau^5) \tag{3.3}$$

any of which we can approximate efficiently via the above methods. Furthermore, as we only need to compute b once, we can take a large value, k_b, for k in those methods. As is standard for quadrature methods, the accuracy can often be improved by subdividing the interval. For example, using Simpson's rule and subdividing into intervals of length $h \coloneqq \tau/(2m)$ we get

$$b = \frac{1}{3}h \left(I + 2\sum_{j=1}^{m-1} e^{-2jhA} + 4\sum_{j=1}^{m} e^{-(2j-1)hA} + e^{-\tau A} \right) f + \mathcal{O}(\tau h^4)$$

which if $k_b = 2m$, that is, the Simpson subdivision equals the Strang/Yoshida step number, can be approximated efficiently by

$$b = \frac{1}{3}h \left(f + 2\sum_{j=1}^{m-1} v_{2j} + 4\sum_{j=1}^{m} v_{2j-1} + v_{2m} \right) + E_1 + E_2 + \mathcal{O}(\tau h^4),$$

where $v_r \coloneqq S^r(h)f$ with $E_1 = \mathcal{O}(\tau^2 h)$ or $v_r \coloneqq Y^r(h)f$ with $E_1 = \mathcal{O}(\tau^2 h^3)$, and E_2 is the spectrum truncation error. Finally, we can also let MATLAB compute its preferred quadrature using the built-in `integrate` function, using either the Strang formula or Yoshida method to compute the integrand. However, we found this to be very slow.

Another method to compute b is to solve an ODE. We note that, by (2.2), b is the fidelity forced diffusion of $\mathbf{0}$ at time τ, that is,

$$\frac{dv}{dt}(t) = -\Delta v(t) - Mv(t) + f, \quad v(0) = \mathbf{0},$$

[14] This method can be extended to give higher-order formulae of any even order, but consideration of those formulae is beyond the scope of this paper.

has $v(\tau) = b$. Hence we can approximate b by solving

$$\frac{d\hat{v}}{dt}(t) = -U_1\Lambda(U_2^T\hat{v}(t)) - \mu \odot \hat{v}(t) + f, \quad \hat{v}(0) = \mathbf{0},$$

via the semi-implicit Euler method from [32]. Since we only need to compute b once we can choose a small time step, that is, a time step of τ/k_b for k_b large, for this Euler method. One could also choose a higher-order ODE solver for this same reason, however as [32] notes this ODE is stiff, which we found causes standard solvers such as `ode45` (ie, Dormand-Prince-(4, 5) [18]) to be inaccurate, and we ran into the issue of the MATLAB stiff solvers requiring matrices too large to fit in memory.

Finally, we can try to compute the formula for b directly. By the Strang formula or Yoshida method, we can efficiently compute $g := f - e^{-\tau A}f$. It remains to compute $b = A^{-1}g$. Given our approximation $\Delta \approx U_1\Lambda U_2^T$, by the Woodbury identity [42]

$$A^{-1} = (\Delta + M)^{-1} \approx (I - U_1\Sigma U_2^T + M)^{-1} = Y^{-1} - Y^{-1}U_1\left(-\Sigma^+ + U_2^TY^{-1}U_1\right)^{-1}U_2^TY^{-1}.$$

where $Y := I + M$, superscript $+$ denotes the pseudoinverse, and recall that $\Sigma = I_K - \Lambda$. Then

$$b = y \odot (g - U_1x),$$

where $y := (1 + \mu)^{-1}$, reciprocation applied elementwise, and x is given by solving

$$\left(-\Sigma^+ + U_2^T(y \odot U_1)\right)x = U_2^T(y \odot g),$$

where we define $y \odot U_1$ as columnwise Hadamard multiplication, that is, $(y \odot U_1)_{ij} := y_i(U_1)_{ij}$. We compare the accuracy of these approximations of b in Table 1, and observe that no method is hands-down superior. Table 1 also indicates that the likely reason for methods like Simpson's rule not performing as well as expected is that the spectrum truncation error is dominating.

Given these ingredients, it is then straightforward to compute the SDIE scheme sequence via Algorithm 2.

3.4.1 | Numerical assessment of methods

In this section, we will build our graphs on the image in Figure 5, which is small enough for us to compute the true values of $e^{-\tau A}u$ (with A here given by $\Delta_s + M$) and b to high accuracy. First, in Figure 6 we investigate the accuracy of the Strang formula and Yoshida method vs the [32] Euler method. We take $|V|$=1600, $\tau = 0.5$, u a random vector given by MATLAB's `rand`(1600,1), and μ as the characteristic function of the left two quadrants of the image. We consider two cases: one where $K = |V|$ (ie, full-rank) and one where $K = \sqrt{|V|}$. We observe that the Strang formula and Yoshida method are more accurate than the Euler method in both cases, and that the Yoshida method is more accurate than the Strang formula, but only barely in the rank-reduced case. Furthermore, the log-log gradients of the Strang formula error and the Yoshida

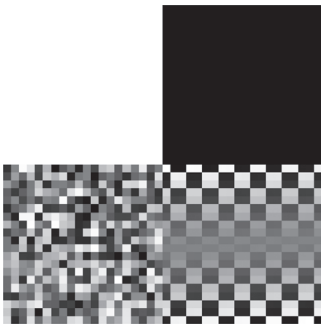


FIGURE 5 The 40×40 or 80×80 image that we build our graphs on, using feature vectors as described in Section 4.2

Algorithm 2. The SDIE scheme via a Strang formula method.

function SDIE($\mu, f, U_1, \Lambda, U_2, \tau, \varepsilon, u_0, k, k_b, K, \delta$) ▷ Computes the terminus of the SDIE scheme.
  if using the quadrature method **then**
    $F_1 : t \mapsto e^{-tA}f$ ▷ Computed using Strang formula or Yoshida method, with parameter k_b
    $b = \texttt{quadrature}(F_1, [0, \tau])$ ▷ Approximates $\int_0^\tau F_1(t)\, dt$ by some quadrature method
  else if using the ODE method **then**
    $F_2 : x \mapsto \left(-U_1\Lambda(U_2^T x) - \mu \odot x + f\right)$
    $\hat{v} = \texttt{ode\_solver}(F_2, [0, \tau], \mathbf{0})$ ▷ Solves $\hat{v}'(t) = F_2(\hat{v})$ on $[0, \tau]$ with $\hat{v}(0) = \mathbf{0}$
    $b = \hat{v}(\tau)$
  else if using the Woodbury identity method **then**
    $g = f - e^{-\tau A}f$ ▷ Computed using Strang formula or Yoshida method, with parameter k_b
    $y = (1 + \mu)^{-1}$
    $\Sigma = I_K - \Lambda$
    $(-\Sigma^+ + U_2^T(y \odot U_1))x = U_2^T(y \odot g)$ ▷ Solving the linear system for x
    $b = y \odot (g - U_1 x)$
  end if
  $a_1 = \exp(-\tau/k\mu)$
  $a_2 = \exp(-\tau/k\,\mathrm{diag}(\Lambda)) - \mathbf{1}_K$
  $a_3 = \mathrm{sqrt}(a_1)$
  $n = 0$
  while $||u_n - u_{n-1}||_2^2 / ||u_n||_2^2 \geq \delta$ **do**
    $v = u_n$
    for $r = 1$ to k **do**
      $v = a_1 \odot v + a_3 \odot \left(U_1 \left(a_2 \odot \left(U_2^T (a_3 \odot v)\right)\right)\right)$ ▷ Strang formula iteration
    end for
    $v = v + b$ ▷ Approximates $v = \mathcal{S}_\tau u_n$
    $V_1 = \{i \in V \mid v_i \in [\tau/2\varepsilon, 1 - \tau/2\varepsilon)\}$
    $V_2 = \{i \in V \mid v_i \geq 1 - \tau/2\varepsilon\}$
    $u_{n+1} = (1 - \tau/\varepsilon)^{-1}(v - \tau/2\varepsilon\mathbf{1}) \odot \chi_{V_1} + \chi_{V_2}$ ▷ Applies (2.16), the piecewise linear thresholding
    $n = n + 1$
  end while
  return u_n
end function

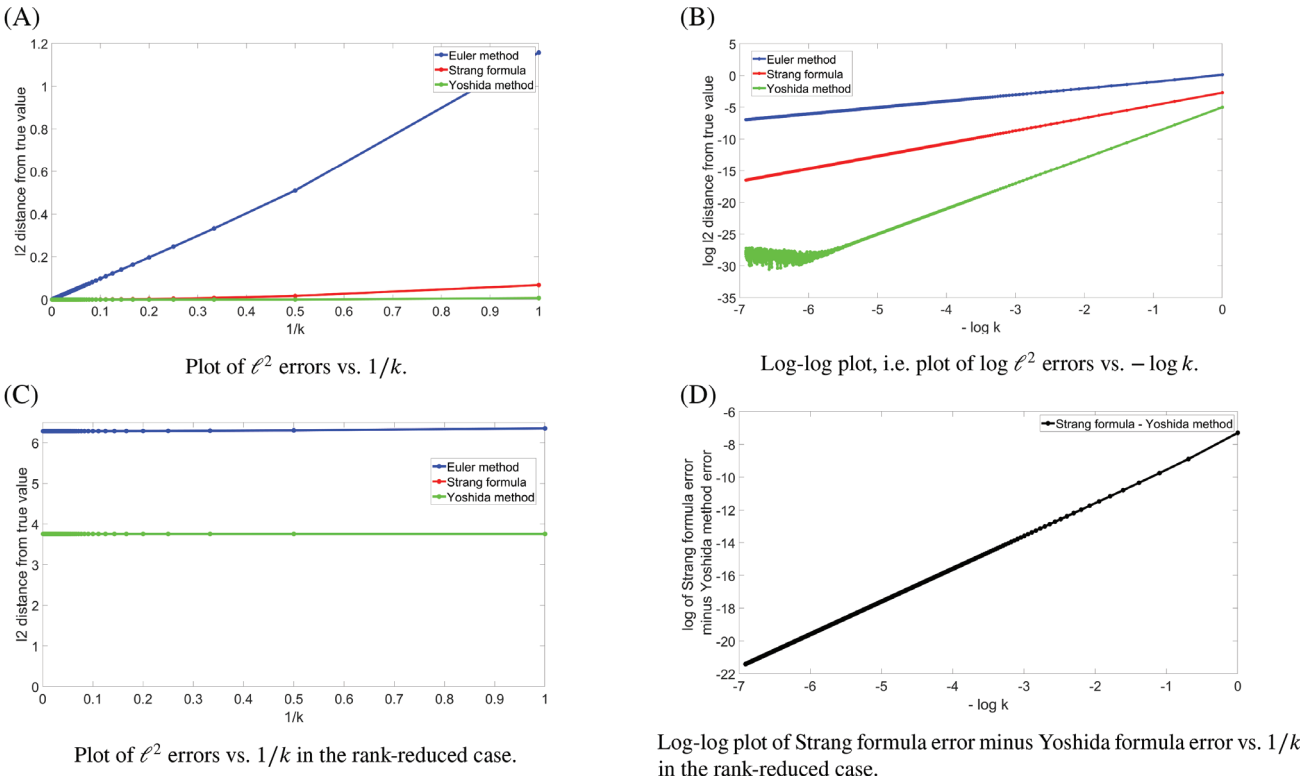
method error (excluding the outliers for small k values and the outliers caused by errors from reaching machine precision) in Figure 6(b) are respectively 2.000 and 3.997 (computed using `polyfit`), confirming that these methods achieve their theoretical orders of error in the full-rank case.

Next, in Table 1 we compare the accuracy of the different methods for computing b. We take Z as the left two quadrants of the image, $\mu = \chi_Z$, $\tilde{f}$ as equal to the image on Z, and $k_b = 1000$ in the Strang formula/Yoshida method approximations for $e^{-tA}f$ and in the [32] Euler scheme. We observe that the rank reduction plays a significant role in the errors incurred, and that no method is hands-down superior. In the "two cows" application (Example 4.1), we have observed that (interestingly) the [32] Euler method yields the best segmentation. A topic for future research can be whether this is generally true for large matrices.

4 | APPLICATIONS IN IMAGE PROCESSING

4.1 | Examples

We consider three examples, all using images of cows from the Microsoft Research Cambridge Object Recognition Image Database (available at `https://www.microsoft.com/en-us/research/project/image-`



Plot of ℓ^2 errors vs. $1/k$.

Log-log plot, i.e. plot of log ℓ^2 errors vs. $-\log k$.

Plot of ℓ^2 errors vs. $1/k$ in the rank-reduced case.

Log-log plot of Strang formula error minus Yoshida formula error vs. $1/k$ in the rank-reduced case.

FIGURE 6 Comparison of the ℓ^2 error from approximating $e^{-\tau A}u$ via the semi-implicit Euler method (blue), Strang formula (red), and Yoshida method (green) approximations for $e^{-\tau A}u$ on the graph built on the image in Figure 5. A,B, $K = |V|$. C,D, $K = \sqrt{|V|}$. The gradient of the line in (D), excluding outliers for small k, is 2.000. In (C), the green and red lines are both plotted but cannot be distinguished by the eye

`understanding/`[15]). Some of these images have been used before by [6, 32] to illustrate and test graph-based segmentation algorithms.

Example 4.1 (Two cows). We first introduce the two cows example familiar from [6, 32]. We take the image of two cows in the top left of Figure 7 as the reference data Z and the segmentation in the bottom left as the reference labels $\tilde{f}$, which separate the cows from the background. We apply the SDIE scheme to segment the image of two cows shown in the top right of Figure 7, aiming to separate the cows from the background, and compare to the ground truth in the bottom right. Both images are RGB images of size 480×640 pixels, that is, the reference data and the image are tensors of size $480 \times 640 \times 3$.

We will use Example 4.1 to illustrate the application of the SDIE scheme. Moreover, we will run several numerical experiments on this example. Namely, we will:

- study the influence of the parameters ϵ and τ, comparing non-MBO SDIE ($\tau < \epsilon$) and MBO SDIE ($\tau = \epsilon$);
- compare different normalizations of the graph Laplacian, that is, the symmetric vs random walk normalization;
- investigate the influence of the Nyström-QR approximation of the graph Laplacian in terms of the rank K; and
- quantify the inherent uncertainty in the computational strategy induced by the randomized Nyström approximation.

Example 4.2 (Grayscale). This example is the grayscale version of Example 4.1. Hence, we map the images in Figure 7 to grayscale using `rgb2gray`. We show the grayscale images in Figure 8. We use the same segmentation of the reference data image as in Example 4.1. The grayscale images are matrices of size 480×640.

The grayscale image is much harder to segment than the RGB image, as there is no clear color separation. With Example 4.2, we aim to illustrate the performance of the SDIE scheme in a harder segmentation task.

[15] Accessed 20 October 2020.

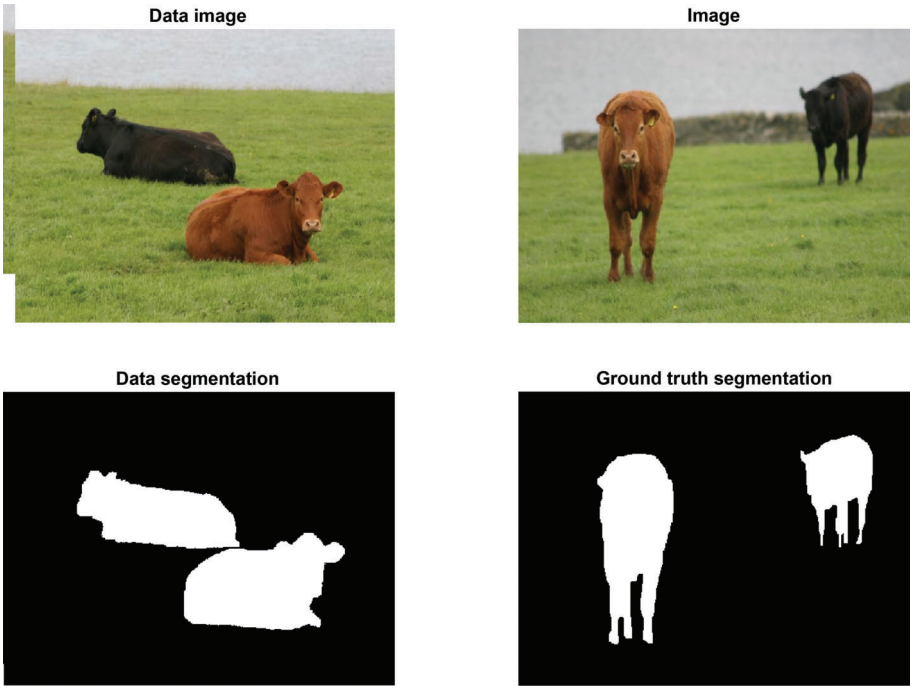


FIGURE 7 Two cows: the reference data image, the image to be segmented, the reference $\tilde{f}$, and the ground truth segmentation associated to Example 4.1. *Note.* During proofing, the authors noticed that a mistake during the hand-segmentation had caused 35 pixels in the bottom-left of the data segmentation to be mislabelled as "cow". The figures in this paper were produced with this slightly misclassified reference data. Test runs with this mistake fixed suggest that its impact was negligible. Both segmentations were drawn by hand by the authors

FIGURE 8 Two cows grayscale: the reference data image and the image to be segmented associated to Example 4.2. Note that the reference $\tilde{f}$ and the ground truth segmentation are identical to those in Figure 7

FIGURE 9 Many cows: the image to be segmented associated to Example 4.3. Note that the reference data image and labels are identical to those in Figure 7 (left)

Example 4.3 (Many cows). In this example, we have concatenated four images of cows that we aim to segment as a whole. We show the concatenated image in Figure 9. Again, we shall separate the cows from the background. As reference data, we use the reference data image and labels as in Example 4.1. Hence, the reference data is a tensor of size $480 \times 640 \times 3$. The image consists of approximately 1.23 megapixels. It is represented by a tensor of size $480 \times 2560 \times 3$.

With Example 4.3, we will illustrate the application of the SDIE scheme to large scale images, as well as the case where the image and reference data are of different sizes.

Note. In each of these examples we took as reference data a separate reference data image. However, our algorithm does not require this, and one could take a subset of the pixels of a single image to be the reference data, and thereby investigate the impact of the relative size of the reference data on the segmentation, which is beyond the scope of this paper but is explored for the [32] MBO segmentation algorithm and related methods in [37, Figure 4].

4.2 | Set-up

Algorithms
We here use the Nyström-QR method to compute the rank K approximation to the Laplacian, and we use the [32] semi-implicit Euler method (with time step τ/k_b) to compute b (as we found that in practice this worked best for the above examples).

Feature vectors
Let $\mathcal{N}(i)$ denote the 3×3 neighborhood of pixel $i\in V$ in the image (with replication padding at borders performed via `padarray`) and let $\mathcal{K}$ be a 3×3 Gaussian kernel with standard deviation 1 (computed via `fspecial('gaussian',3,1)`). Thus $x|_{\mathcal{N}(i)}$ can be viewed as a triple of 3×3 matrices $x^J|_{\mathcal{N}(i)}$ for $J\in\{R,G,B\}$ (ie, one in each of the R, G, and B channels). Then in each channel we define

$$z_i^R := 9\mathcal{K}\odot x^R|_{\mathcal{N}(i)}, \qquad z_i^G := 9\mathcal{K}\odot x^G|_{\mathcal{N}(i)}, \qquad z_i^B := 9\mathcal{K}\odot x^B|_{\mathcal{N}(i)},$$

and thus define $z_i := (z_i^R, z_i^G, z_i^B)\in\mathbb{R}^{3\times 3\times 3}$, which we reshaped (using `reshape`) so that $z\in\mathbb{R}^{|V|\times 27}$.

Interpolation sets
For the interpolation sets X_1, X_2 in Nyström we took $K_1 = K_2 = K/2$. That is, we took $K/2$ vertices from the reference data image and $K/2$ vertices from the image to be segmented, chosen at random using `randperm`. We experimented with choosing interpolation sets using ACA (see [3]), but this showed no improvement over choosing random sets, and ran much slower.

Initial condition
We took the initial condition, that is, u_0, to equal the reference $\tilde{f}$ on the reference data vertices and to equal 0.49 on the vertices of the image to be segmented (where $\tilde{f}$ labels "cow" with 1 and "not cow" with 0). We used 0.49 rather than the more natural 0.5 because the latter led to much more of the background (eg, the grass) getting labeled as "cow." This choice can be viewed as incorporating the slight extra a priori information that the image to be segmented has more non-cow than cow.

Note. It is not an improvement to initialize with the exact proportion of cow in the image (about 0.128), as in that case the thresholding is liable to send every u_i for $i\in V\setminus Z$ to zero. If one has access to that a priori information, one should use such an initial condition alongside a mass-conserving scheme. We investigate a semi-discrete scheme for mass-conserving Allen-Cahn (without fidelity forcing) in [14], which can be extended to the fidelity-forced case in a manner similar to that of this paper. What the 0.49 initial condition does is more modest. After the initial diffusion, some pixels will have a value very close to 0.5; by lowering the value of the initial condition, we lower the diffused values, introducing a bias towards classifying these borderline pixels as "not cow." As there is more non-cow than cow in the image, this bias improves the accuracy. It is unknown to the authors why this was necessary for us but was not necessary for [32] nor for [6].

Fidelity parameter
We followed [32] and took $\mu = \hat{\mu}\chi_Z$, for $\hat{\mu} > 0$ a parameter.

Computational set-up
All programming was done in MATLAB R2019a with relevant toolboxes the Computer Vision Toolbox Version 9.0, Image Processing Toolbox Version 10.4, and Signal Processing Toolbox Version 8.2. All reported runtimes are of implementations executed serially on a machine with an Intel® Core™ i7-9800X @ 3.80 GHz [16 cores] CPU and 32 GB RAM of memory.

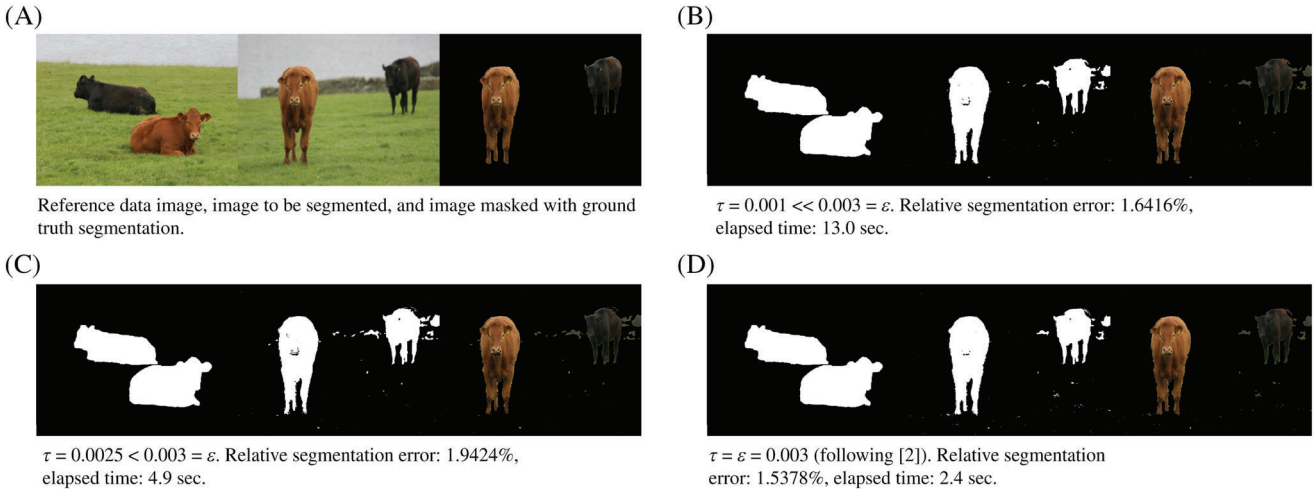


FIGURE 10 MBO SDIE and non-MBO SDIE segmentations for the *two cows* segmentation task. In the top left figure, we show the reference data image, the image to be segmented, and the image masked with the segmentation we consider the ground truth, see also Figure 7. The other figures (B-D) show the labels on the reference data, the segmentation returned by the respective algorithm, and the original images masked with the segmentation

4.3 | Two cows

We begin with some examples of segmentations obtained from the SDIE scheme. Based on these, we illustrate the progression of the algorithm and discuss the segmentation output qualitatively. We will give a quantitative analysis in Section 4.3.1. Note that we give here merely *typical* realizations of the random output of the algorithm—the output is random due to the random choice of interpolation sets in the Nyström approximation. We investigate the stochasticity of the algorithm in Section 4.3.2.

We consider three different cases: the MBO case $\tau = \varepsilon$ and two non-MBO cases, where $\tau \ll \varepsilon$, and $\tau < \varepsilon$. We show the resulting reconstructions from these methods in Figure 10. Moreover, we show the progression of the algorithms in Figure 12. The parameters not given in the captions of Figure 10 are $\hat{\mu} = 30$, $\sigma = 35$, $k_b = 1$, $k = 1$, $\delta = 10^{-10}$, and $K = 70$.

Note. The regime $\tau > \varepsilon$ is not of much interest since, by [13, Remark 4.8] *mutatis mutandis*, in this regime the SDIE scheme has nonunique solution for the update, of which one is just the MBO solution.

Comparing the results in Figure 10, we see roughly equivalent segmentations and segmentation errors. Indeed, the cows are generally nicely segmented in each of the cases. However, the segmentation also labels as "cow" a part of the wall in the background and small clumps of grass, while a small part of the left cow's snout is cut out. This may be because the reference data image does not contain these features and so the scheme is not able to handle them correctly.

In Figure 11 we compare the result of Figure 10(d) (our best segmentation) with the results of the analogous experiments in [6, 32]. We observe significant qualitative improvement. In particular, our method achieves a much more complete identification of the left cow's snout, complete identification of the left cow's eyes and ear tag, and a slightly more complete identification of the right cow's hind. However, our method does misclassify more of the grass than the methods in [6, 32] do. Note that as well as the change in method, the reference that we hand-drew (see Figure 7, bottom-left) is different from both the reference used in [6, Figure 4.6] and the reference [32, Figure 2(e)].

We measure the computational cost of the SDIE scheme through the measured runtime of the respective algorithm. We note from Figure 10 that the MBO scheme ($\tau = \varepsilon$) outperforms the non-MBO schemes ($\tau < \varepsilon$); the SDIE relaxation of the MBO scheme merely slows down the convergence of the algorithm, without improving the segmentation. This can especially be seen in Figure 12, where the SDIE scheme needs many more steps to satisfy the termination criterion. At least for this example, the non-MBO SDIE scheme is less efficient than the MBO scheme. Thus, in the following sections we focus on the MBO case.

FIGURE 11 Comparison of our segmentation (using the set-up in Figure 10D) with the analogous segmentations from the previous literature [6, 32], both reproduced with permission from SIAM and the authors. Note that unfortunately in reproduction the color balances and aspect ratios have become slightly inconsistent, but we can still make qualitative comparisons

4.3.1 | Errors and timings

We now quantify the influence, on the accuracy and computational cost of the segmentation, of the Nyström rank K, the number of discretization steps k_b and k in the Euler method and the Strang formula respectively, and the choice of normalization of the graph Laplacian. To this end, we segment the *two cows* image using the following parameters: $\varepsilon = \tau = 0.003$, $\hat{\mu} = 30$, $\sigma = 35$, and $\delta = 10^{-10}$. We take $K \in \{10, 25, 70, 100, 250\}$, $(k_b, k) \in \{(1, 1), (10, 5)\}$, and use the random walk Laplacian Δ and the symmetric normalized Laplacian Δ_s.

We plot total runtimes (ie, the time elapsed from the loading of the image to be segmented, the reference data image, and the reference, to the output of the segmentation) and relative segmentation errors in Figure 13. As our method has randomness from the Nyström extension, we repeat every experiment 100 times and show means and standard deviations. We make several observations. Starting with the runtimes, we indeed see that these are roughly linear in K, verifying numerically the expected complexity. The runtime also increases when increasing k_b and k. That is, increasing the accuracy of the Euler method and Strang formula does not lead to faster convergence, and also does not increase the accuracy of the overall segmentation. Finally, we see that the symmetric normalized Laplacian incurs consistently low relative segmentation error for small values of K. This property is of the utmost importance to scale up our algorithm for very large images. Interestingly, the segmentations using the symmetric normalized Laplacian seem to deteriorate for a large K. The random walk Laplacian has diametric properties in this regard: the segmentations are only reliably accurate when K is reasonably large.

4.3.2 | Uncertainty in the segmentation

Due to the randomized Nyström approximation, our approximation of the SDIE scheme is inherently stochastic. Therefore, the segmentations that the algorithm returns are realizations of random variables. We now briefly study these random variables, especially with regard to K. We show pointwise mean and standard deviations of the binary labels in each of the left two columns of the four subfigures of Figure 14. In the remaining figures, we weight the original *two cows* image with these means (varying continuously between label 1 for "cow" and label 0 for "not cow") and standard deviations. For these experiments we use the same parameter set-up as Section 4.3.1.

We make several observations. First, as K increases we see less variation. This is what we expect, as when $K = |V|$ the method is deterministic so has no variation. Second, the type of normalization of the Laplacian has an effect: the symmetric normalized Laplacian leads to less variation than the random walk Laplacian. Third, the parameters k_b and k appear to have no major effect within the range tested. Finally, looking at the figures with rather large K, we observe that the standard deviation of the labels is high in the areas of the figure in which there is indeed uncertainty in the segmentation, namely the boundaries of the cows and the parts of the wall with similar color to the dark cow. Determining the exact position of the boundary of a cow on a pixel-by-pixel level is indeed also difficult for a human observer. Moreover, the SDIE scheme usually confuses the wall in the background for a cow. Hence, a large standard deviation here reflects that the estimate is uncertain.

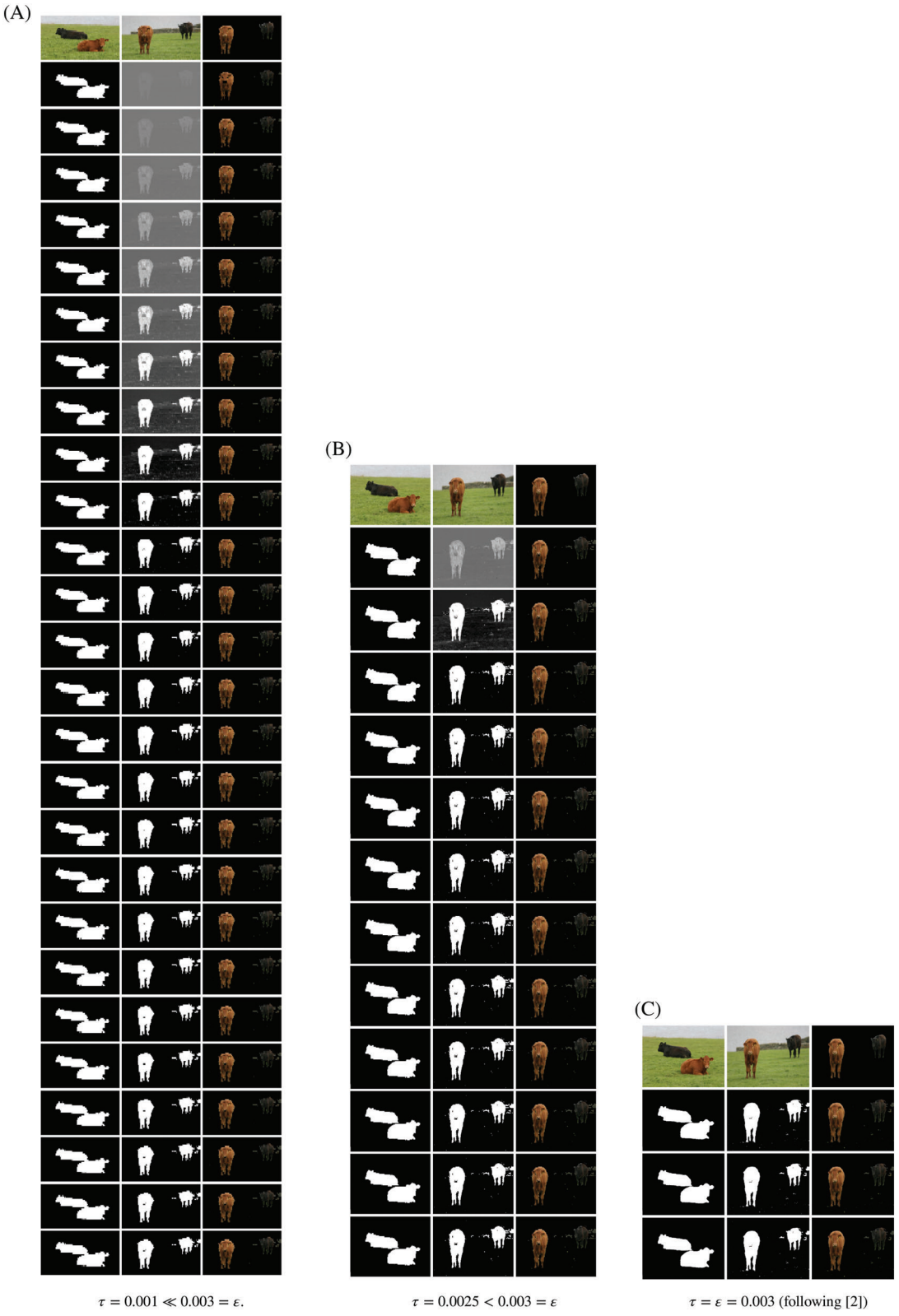


FIGURE 12 Progression of MBO and non-MBO SDIE for the *two cows* example. In each subfigure: the first row shows the reference data, image, and ground truth, as in Figure 10. The middle rows, showing the reshaped label vector u_n and the image masked by the thresholded label, each represent one iteration of the considered algorithm, to be read from top to bottom. The last row gives the state returned by the scheme, that is, the state satisfying the termination criterion, which correspond to the subfigures in Figure 10. For layout reasons, we have squashed the figure in (A)

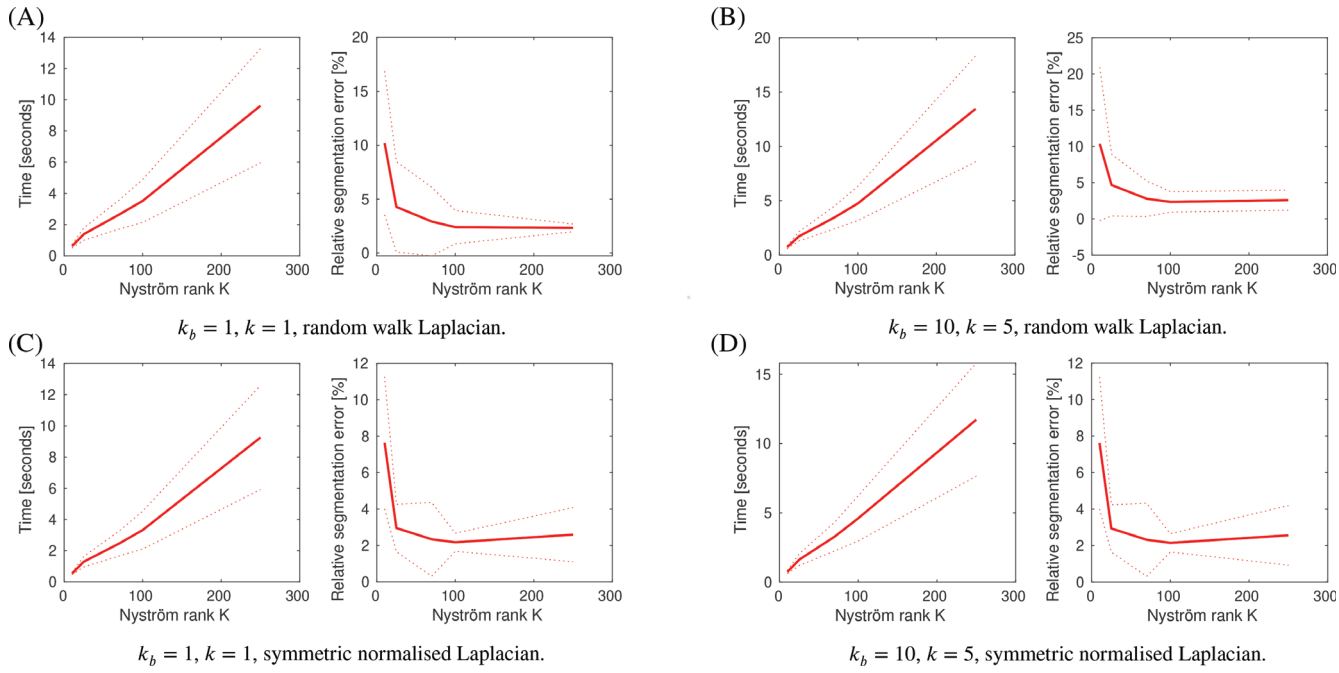


FIGURE 13 Error and timing of 100 independent segmentations of the two cows image (Example 4.1) with the MBO SDIE scheme. The solid lines represent the means averaged over 100 runs, the dotted lines show means ±standard deviations

This is of course not a rigorous Bayesian uncertainty quantification, as for instance is given in [7, 37] for graph-based learning. However the use of stochastic algorithms for inference tasks, and the use of their output as a method of uncertainty quantification, has for instance been motivated by [31].

4.4 | Grayscale

We now move on to Example 4.2, the *grayscale* problem. We will especially use this example to study the influence of the parameters $\hat{\mu}$ and σ. The parameter $\hat{\mu} > 0$ determines the strength of the fidelity term in the ACE. From a statistical point of view, we can understand a choice of $\hat{\mu}$ as an assumption on the statistical precision (ie, the inverse of the variance of the noise) of the reference $\tilde{f}$ (see [7, Section 3.3] for details). Thus, a small $\hat{\mu}$ should lead to a stronger regularization coming from the Ginzburg-Landau functional, and a large $\hat{\mu}$ leads to more adherence to the reference. The parameter $\sigma > 0$ is the "standard deviation" in the Gaussian kernel Ω used to build the weight matrix ω. For our methods we must not choose too small a σ, as otherwise the weight matrix becomes sparse (up to some precision), and so the Nyström submatrix ω_{XX} has a high probability of being ill-conditioned. In such a case this sparsity can be exploited using Rayleigh-Chebyshev [2] methods as in [32], or MATLAB sparse matrix algorithms as in [6], but this lies beyond the scope of this paper. If σ is too large then the graph structure no longer reflects the features of the image.

In the following, we set $\varepsilon = \tau = 0.00024$, $k_b = 10$, $k = 5$, and $\delta = 10^{-10}$. To get reliable results we choose a rather large K, $K = 200$, and therefore (by the discussion in Section 4.3.2) we use the random walk Laplacian. We will qualitatively study single realizations of the inherently stochastic SDIE algorithm. We vary $\hat{\mu} \in \{50, 100, 150, 200\}$ and $\sigma \in \{2, 35, 10^3, 10^4\}$. We show the best results from these tests in Figure 15. Moreover, we give a comprehensive overview of all tests and the progression of the SDIE scheme in Figure 16.

We observe that this segmentation problem is indeed considerably harder than the *two cows* problem, as we anticipated after stating Example 4.2. The difference in shade between the left cow and the wall is less visible than in Example 4.1, and the left cow's snout is less identifiable as part of the cow. Thus, the segmentation errors we incur are about three times larger than before. There is hardly any visible influence from changing σ in $\{35, 10^3\}$. However, $\sigma = 2$ and $\sigma = 10^4$ lead to significantly worse results. For $\sigma = 2$, the sparsity (up to some precision) of the matrix deteriorates the results and the method becomes unstable; for $\sigma = 10^4$, the weight matrix does not sufficient distinguish pixels of different shade, which is why the method labels everything as "not cow."

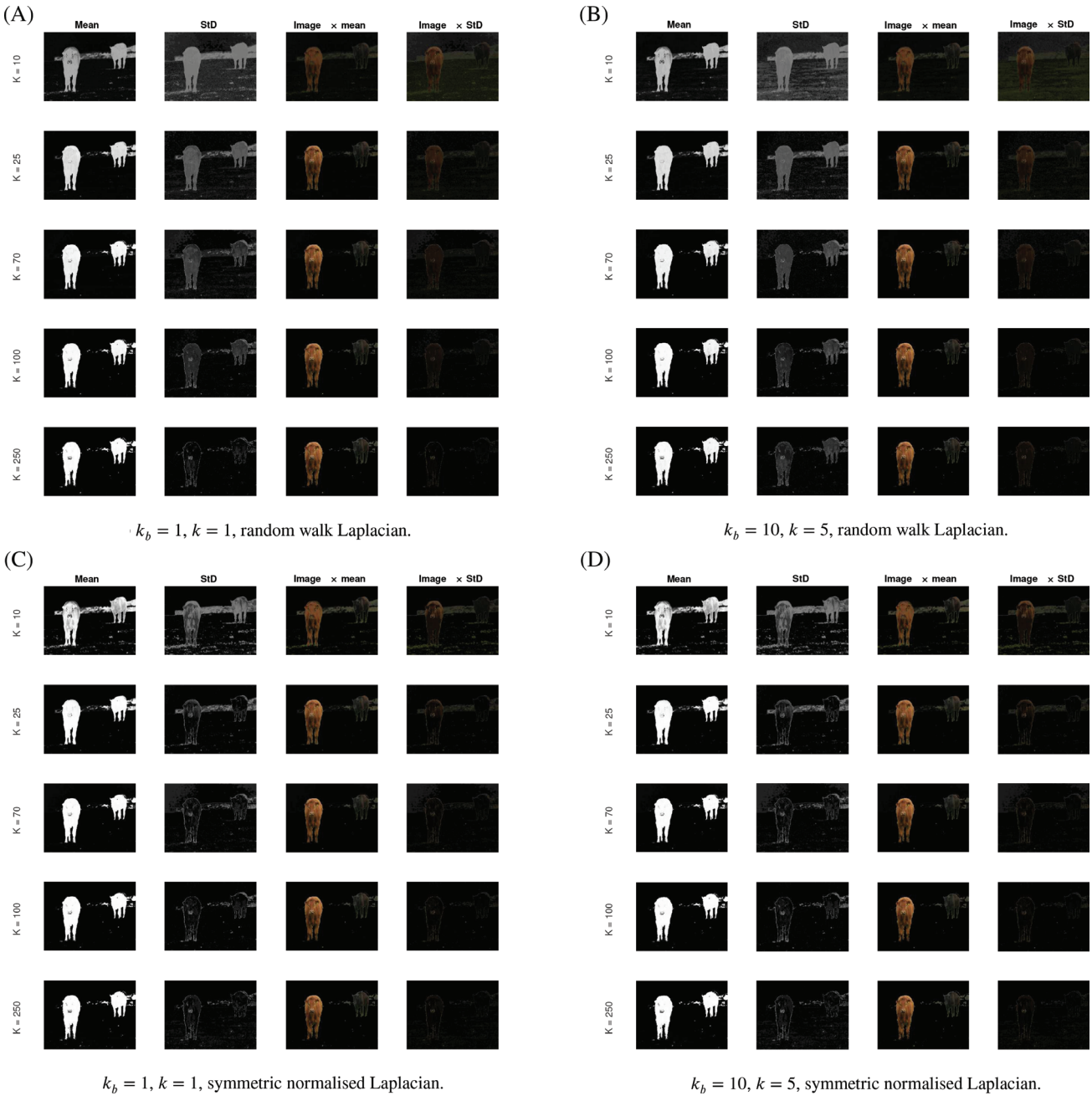


FIGURE 14 Mean, standard deviation, and images weighted by mean and standard deviation of 100 independent segmentations of Example 4.1 with the MBO SDIE scheme, with set-up as in Section 4.3.1

Given σ in $\{35, 10^3\}$, the strength of the fidelity term $\hat{\mu}$ has a significant impact on the result, as well. Indeed, for $\hat{\mu} = 50$ the algorithm does not find any segmentation. For $\hat{\mu} \geq 100$, we get more practical segmentations. Interestingly, for $\hat{\mu} = 150$ and $\hat{\mu} = 200$ we get almost all of the left cow, but misclassify most of the wall in the background; for $\hat{\mu} = 100$, we miss a large part of the left cow, but classify more accurately the wall. The interpretation of $\hat{\mu}$ as the statistical precision of the reference explains this effect well. For $\hat{\mu} = 100$, we assume that the reference is less precise, leading us (due to the smoothing effect of the Ginzburg-Landau regularization) to classify accurately most of the wall. With $\hat{\mu} \geq 150$, we assume that the reference is more precise, leading us to misclassify the wall (due to its similarity to the cows in the reference data image) but classify accurately more of the cows. At $\hat{\mu} = 200$, this effect even leads to a larger total segmentation error. The runtime varied throughout the experiments: the standard seems to be between 6 and 7 seconds. By doing an additional

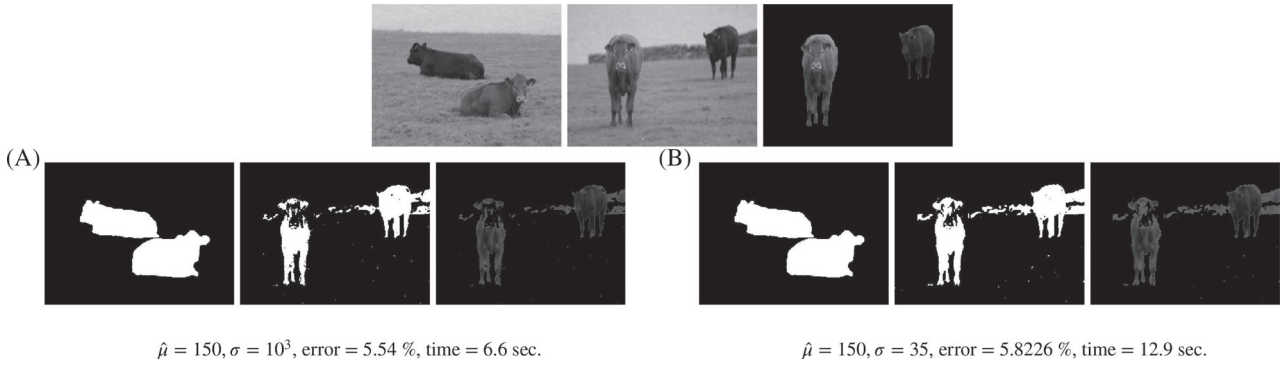


FIGURE 15 MBO SDIE segmentations for the *grayscale* segmentation task. In the centered top figure, we show the reference data image, the image to be segmented, and the image masked with the ground truth segmentation, cf. Figure 7. The other figures show the reference $\tilde{f}$, the segmentation returned by the algorithm, and the original images masked with the segmentation

step in the iteration, the times for $\sigma = 2$ as well as $(\hat{\mu}, \sigma) = (50, 35)$ are significantly increased. We also see a larger runtime in the case of $(\hat{\mu}, \sigma) = (150, 35)$.

4.5 | Many cows

We finally study the *many cows* example, that is, Example 4.3. The main differences to the earlier examples are the larger size of the image that is to be segmented and the variety within it. We first comment on the size. The image is given by a $480 \times 2560 \times 3$ tensor, which is a manageable size. The graph Laplacian, however, is a dense matrix with 1.536×10^6 rows and columns. A matrix of this size requires 17.6 TB of memory to be constructed in MATLABR2019a. This image is much more difficult to segment than the previous examples, in which the cows in the image to be segmented are very similar to the cows in the reference data. Here, we have concatenated images of cows that look very different, for example, cows with a white blaze on their nose.

As the *two cows* image is part of the *many cows* image, we first test the algorithmic set-up that was successful at segmenting the former. We show the result (and remind the reader of the set-up) in Figure 17(a). The segmentation obtained in this experiment is rather mediocre—even the *two cows* part is only coarsely reconstructed. We present two more attempts at segmenting the *many cows* image in Figure 17: we choose $\varepsilon = \tau = 0.00025$, a slightly larger Nyström rank $K = 100$, and vary $(k_b, k, \hat{\mu}) \in \{(1, 1{,}500), (10, 10{,}400)\}$. In both cases, we obtain a considerably better segmentation of the image. In the case where $\hat{\mu} = 500$, we see a good, but slightly noisy segmentation of the brown and black parts of the cows. In the case where $\hat{\mu} = 400$, we reduce the noise in the segmentation, but then also misclassify some parts of the cows. The blaze (and surrounding fur) is not recognized as "cow" in any of the segmentations, likely because the blaze is not represented in the reference data image. The influence of the set-up on the runtimes is now much more pronounced. For the given segmentations, however, all the runtimes are at most a factor of eight larger than the smallest runtimes in the previous examples, despite the larger image size.

5 | CONCLUSION

Extending the set-up and results in [13], we defined the continuous-in-time graph ACE with fidelity forcing as in [6] and a SDIE time discretization scheme for this equation. We proved well-posedness of the ACE and showed that solutions of the SDIE scheme converge to the solution of the ACE. Moreover, we proved that the graph MBO scheme with fidelity forcing [32] is a special case of an SDIE scheme.

We provided an algorithm to solve the SDIE scheme, which—besides the obvious extension from the MBO scheme to the SDIE scheme—differs in two key places from the existing algorithm for graph MBO with fidelity forcing: it implements the Nyström extension via a QR decomposition and it replaces the Euler discretization of the diffusion step with a computation based on the Strang formula for matrix exponentials. The former of these changes appears to have been a quite significant improvement: in experiments the Nyström-QR method proved to be faster, more accurate, and more

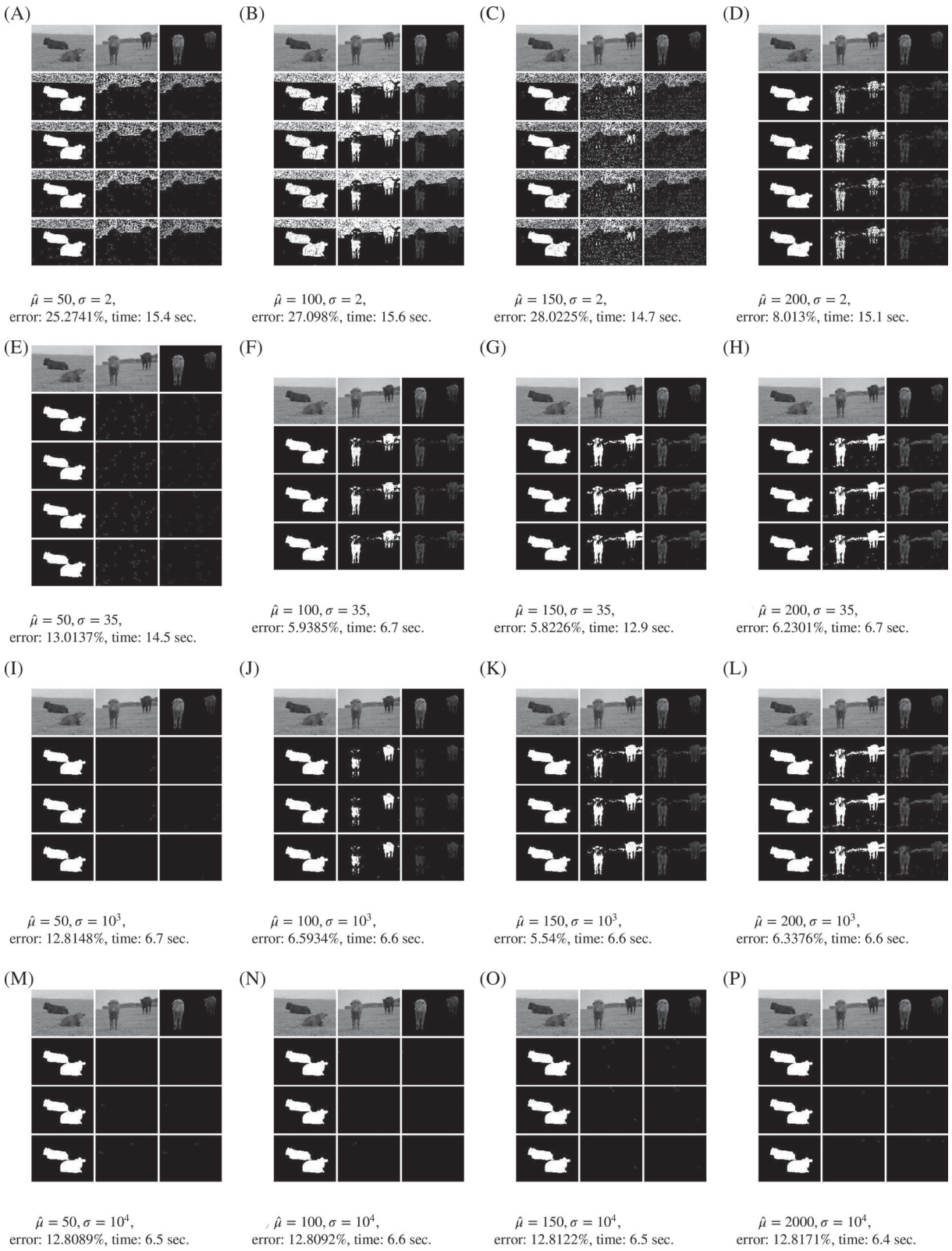


FIGURE 16 Progression of the MBO SDIE scheme for the *grayscale* segmentation task. In each subfigure: The first row shows the reference data, the image to be segmented, and the ground truth segmentation. The middle rows, showing the reshaped label vector u_n and the image masked by the label, each represent one iteration of the considered algorithm, to be read from top to bottom. The last row gives the state returned by the scheme, that is, the state satisfying the termination criterion

(A)

Segmentation with parameters $\varepsilon = \tau = 0.003, K = 70, k_b = 1, k = 1, \hat{\mu} = 30, \sigma = 35$, and the symmetric normalised Laplacian. Elapsed time for segmentation: 9.1 sec.

(B)

Segmentation with parameters $\varepsilon = \tau = 0.00025, K = 100, k_b = 1, k = 1, \hat{\mu} = 500, \sigma = 35$, and the symmetric normalised Laplacian. Elapsed time for segmentation: 14.0 sec.

(C)

Segmentation with parameters $\varepsilon = \tau = 0.00025, K = 100, k_b = 10, k = 10, \hat{\mu} = 400, \sigma = 35$, and the symmetric normalised Laplacian. Elapsed time for segmentation: 19.1 sec.

FIGURE 17 Segmentations of the MBO scheme for the *many cows* segmentation task. In each subfigure: the top row shows the reference data image and twice the image that is to be segmented, and the bottom row shows the reference $\tilde{f}$, the segmentation returned by the respective algorithm, and the original image masked with the segmentation

stable than the Nyström method used in previous literature [6, 32], and it is less conceptually troubling than that method, as it does not involve taking the square root of a non-positive-semi-definite matrix.

We applied this algorithm to a number of image segmentation examples concerning images of cows from the Microsoft Research Cambridge Object Recognition Image Database. We found that while the SDIE scheme yielded no improvement over the MBO scheme (and took longer to run in the non-MBO case) the other improvements that we made led to a substantial qualitative improvement over the segmentations of the corresponding examples in [6, 32]. We furthermore investigated empirically various properties of this numerical method and the role of different parameters. In particular:

- We found that the symmetric normalized Laplacian incurred consistently low segmentation error when approximated to a low rank, while the random walk Laplacian was more reliably accurate at higher ranks (where "higher" is still less than 0.1% of the full rank). Thus for applications that require scalability, and thus very low-rank approximations, we recommend using the symmetric normalized Laplacian.
- We investigated empirically the uncertainty inherited from the randomization in the Nyström extension. We found that the rank reduction and the normalization of the graph Laplacian had the most influence on the uncertainty, and we furthermore observed that at higher ranks the segmentations had high variance at those pixels which are genuinely difficult to classify, for example, the boundaries of the cows.
- We noted that the fidelity parameter μ corresponds to ascribing a statistical precision to the reference data. We observed that when the reference data were not fully informative, as in Examples 4.2 and 4.3, it was important to tune this parameter to get an accurate segmentation.

To conclude, we give a number of directions we hope to explore in future work, in no particular order.

We seek to investigate the use of the very recent method of [5] combined with methods such as the HMT method [26] or the (even more recent) generalized Nyström method [35] to further increase the accuracy of the low-rank approximations

to the graph Laplacian. Unfortunately, we became aware of these works too near to finalizing this paper to use those methods here.

We will seek to extend both our theoretical and numerical framework to multiclass graph-based classification, as considered for example in [22, 28, 37]. The groundwork for this extension was laid by the authors in [14, Section 6].

This work dovetails with currently ongoing work of the authors with Carola-Bibiane Schönlieb and Simone Parisotto exploring graph-based joint reconstruction and segmentation with the SDIE scheme. In practice, we often do not observe images directly, but must reconstruct an image from what we observe. For example, when the image is noisy, blurred, has regions missing, or we observe only a transform of the image, such as the output of a CT or MRI scanner. "Joint reconstruction and segmentation" (see [1, 17]) is the task of simultaneously reconstructing and segmenting an image, which can reduce computational time and improve the quality of both the reconstruction and the segmentation.

We will also seek to develop an SDIE classification algorithm that is scalable towards larger data sets containing not just one, but hundreds or thousands of reference data images. This can be attempted via data subsampling: rather than evolving the SDIE scheme with regard to the full reference data set, we use only a data subset that is randomly replaced by a different subset after some time interval has passed. Data subsampling is the fundamental principle of various algorithms used in machine learning, such as stochastic gradient descent [38]. A subsampling strategy that is applicable in continuous-in-time algorithms has recently been proposed and analyzed by [30].

ACKNOWLEDGEMENTS

We thank Andrea Bertozzi, Arjuna Flenner, and Arieh Iserles for very helpful comments. This work dovetails with ongoing work of the authors joint with Carola-Bibiane Schönlieb and Simone Parisotto, whom we therefore also thank for many indirect contributions. The work of Jeremy Budd and Yves van Gennip was supported by the European Union's Horizon 2020 research and innovation program under Marie Skłodowska-Curie grant 777826. The work of the Jonas Latz was supported by the EPSRC grant EP/S026045/1.

REFERENCES

[1] J. Adler, S. Lunz, O. Verdier, C.-B. Schönlieb, and O. Öktem, *Task adapted reconstruction for inverse problems*, 2018, arXiv e-prints: arXiv:1809.00948 [cs.CV].

[2] C. Anderson, A Rayleigh-Chebyshev procedure for finding the smallest eigenvalues and associated eigenvectors of large sparse Hermitian matrices, J. Comput. Phys. **229** (2010), 7477–7487.

[3] M. Bebendorf and S. Kunis, Recompression techniques for adaptive cross approximation, J. Integral Equ. Appl. **21** (2009), 331–357. https://doi.org/10.1216/JIE-2009-21-3-331.

[4] J. Bence, B. Merriman, and S. Osher, Diffusion generated motion by mean curvature, CAM report, 92-18, Department of Mathematics, University of California, Los Angeles, CA, 1992.

[5] K. Bergermann, M. Stoll, and T. Volkmer, *Semi-supervised learning for multilayer graphs using diffuse interface methods and fast matrix vector products*, 2020, arXiv e-prints:arXiv:2007.05239 [cs.LG].

[6] A. L. Bertozzi and A. Flenner, Diffuse interface models on graphs for analysis of high dimensional data, Multiscale Model. Simul. **10** (2012), 1090–1118. https://doi.org/10.1137/11083109X.

[7] A. L. Bertozzi, X. Luo, A. M. Stuart, and K. Zygalakis, Uncertainty quantification in graph-based classification of high dimensional data, SIAM/ASA J. Uncertain. Quant. **6** (2018), 568–595. https://doi.org/10.1137/17M1134214.

[8] J. F. Blowey and C. M. Elliott, The Cahn-Hilliard gradient theory for phase separation with non-smooth free energy, Part I: Mathematical analysis, Eur. J. Appl. Math. **3** (1991), 233–279.

[9] J. F. Blowey and C. M. Elliott, The Cahn-Hilliard gradient theory for phase separation with non-smooth free energy, part II: Numerical analysis, Eur. J. Appl. Math. **3** (1992), 147–179.

[10] J. F. Blowey and C. M. Elliott, *Curvature dependent phase boundary motion and parabolic double obstacle problems*, in *Degenerate diffusions*, The IMA Volumes in Mathematics and its Applications, Vol **47**, W. M. Ni, L. A. Peletier, and J. L. Vazquez, Eds., Springer, New York, NY, 1993, 19–60.

[11] J. Bosch, S. Klamt, and M. Stoll, Generalizing diffuse interface methods on graphs: Non-smooth potentials and hypergraphs, SIAM J. Appl. Math. **78** (2018), 1350–1377.

[12] A. Buades, B. Coll, and J. M. Morel, A review of image denoising algorithms, with a new one, Multiscale Model. Simul. **4** (2005), 490–530.

[13] J. Budd and Y. van Gennip, Graph Merriman–Bence–Osher as a semidiscrete implicit Euler scheme for graph Allen–Cahn flow, SIAM J. Math. Anal. **52** (2020), 4101–4139. https://doi.org/10.1137/19M1277394.

[14] J. Budd and Y. van Gennip, *Mass-conserving diffusion-based dynamics on graphs*, 2020, arXiv e-prints: arXiv:2005.13072 [math.AP].

[15] L. Calatroni, Y. van Gennip, C.-B. Schönlieb, H. M. Rowland, and A. Flenner, Graph clustering, variational image segmentation methods and Hough transform scale detection for object measurement in images, J. Math. Imaging. Vis. **57** (2017), 269–291.

[16] T. Chan and L. Vese, Active contour without edges, IEEE Trans. Image Process. **10** (2001), 266–277.

[17] V. Corona, M. Benning, M. J. Ehrhardt, L. F. Gladden, R. Mair, A. Reci, A. J. Sederman, S. Reichelt, and C.-B. Schönlieb, Enhancing joint reconstruction and segmentation with non-convex Bregman iteration, Inverse Probl. **35** (2019), 055001. https://doi.org/10.1088/1361-6420/ab0b77.
[18] J. R. Dormand and P. J. Prince, A family of embedded Runge–Kutta formulae, J. Comput. Appl. Math. **6** (1980), 19–26.
[19] C. Eckart and G. Young, The approximation of one matrix by another of lower rank, Psychometrika **1** (1936), 211–218.
[20] S. Esedoḡlu and Y. H. R. Tsai, Threshold dynamics for the piecewise constant Mumford–Shah functional, J. Comput. Phys. **211** (2006), 367–384.
[21] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, Spectral grouping using the Nyström method, IEEE Trans. Pattern Anal. Mach. Intell. **26** (2004), 1–12.
[22] C. Garcia-Cardona, E. Merkurjev, A. L. Bertozzi, A. Flenner, and A. G. Percus, Multiclass data segmentation using diffuse interface methods on graphs, IEEE Trans. Pattern Anal. Mach. Intell. **36** (2014), 1600–1613.
[23] Y. van Gennip and A. L. Bertozzi, Γ-convergence of graph Ginzburg–Landau functionals, Adv. Differ. Equ. **17** (2012), 1115–1180.
[24] Y. van Gennip, N. Guillen, B. Osting, and A. L. Bertozzi, Mean curvature, threshold dynamics, and phase field theory on finite graphs, Milan J. Math. **82** (2014), 3–65.
[25] G. H. Golub and C. F. Van Loan, *Matrix computations*, Vol **4**, Johns Hopkins University Press, Baltimore, MD, 2013.
[26] N. Halko, P. G. Martinsson, and J. A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, SIAM Rev. **53** (2011), 217–288.
[27] B. C. Hall, *Lie groups, lie algebras, and representations: An elementary introduction*, in *Graduate Texts in Mathematics*, Vol **222**, Springer, New York, NY, 2015.
[28] M. Jacobs, E. Merkurjev, and S. Esedoḡlu, Auction dynamics: A volume constrained MBO scheme, J. Comput. Phys. **354** (2018), 288–310.
[29] R. V. Kohn and P. Sternberg, Local minimisers and singular perturbations, Proc. Roy. Soc. Edinburgh Sect. A **111** (1989), 69–84.
[30] J. Latz, *Analysis of stochastic gradient descent in continuous time*, 2020, arXiv e-prints: arXiv:2004.07177 [math.PR].
[31] S. Mandt, M. D. Hoffman, and D. M. Blei, Stochastic gradient descent as approximate Bayesian inference, J. Mach. Learn. Res. **18** (2017), 1–35.
[32] E. Merkurjev, T. Kostić, and A. L. Bertozzi, An MBO scheme on graphs for segmentation and image processing, SIAM J. Imaging Sci. **6** (2013), 1903–1930. https://doi.org/10.1137/120886935.
[33] L. Modica and S. Mortola, Un esempio di Γ^{-}-convergenza, Boll. Un. Mat. Ital. B **14** (1977), 285–299.
[34] D. Mumford and J. Shah, Optimal approximation by piecewise smooth functions and associated variational problems, Commun. Pure Appl. Math. **42** (1989), 577–685.
[35] Y. Nakatsukasa, *Fast and stable randomized low-rank matrix approximation*, 2020, arXiv e-prints: arXiv:2009.11392 [math.NA].
[36] E. J. Nyström, Über die Praktische Auflösung von Linearen Integralgleichungen mit Anwendungen auf Randwertaufgaben der Potentialtheorie, Comment. Phys. Math. **4** (1928), 1–52.
[37] Y. Qiao, C. Shi, C. Wang, H. Li, M. Haberland, X. Luo, A. M. Stuart, and A. L. Bertozzi, Uncertainty quantification for semi-supervised multi-class classification in image processing and ego-motion analysis of body-worn videos, Electron. Imaging Image Process. Algorithms Syst. **XVII** (2019), 264-1–264-7.
[38] H. Robbins and S. Monro, A stochastic approximation method, Ann. Math. Stat. **22** (1951), 400–407.
[39] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. **5** (1968), 506–517.
[40] G. Teschl, *Ordinary differential equations and dynamical systems*, in *Graduate Studies in Mathematics*, Vol **140**, American Mathematical Society, Providence, RI, 2012.
[41] M. Varma and A. Zisserman, A statistical approach to texture classification from single images, Int. J. Comput. Vis. **62** (2005), 61–81.
[42] M. A. Woodbury, *Inverting modified matrices*, Memorandum Report 42, Statistical Research Group, Princeton, NJ, 1950.
[43] L. Yaroslavsky, *Digital picture processing: An introduction*, Springer-Verlag, Berlin, Germany, 1985.
[44] H. Yoshida, Construction of higher order symplectic integrators, Phys. Lett. A **150** (1990), 262–268.
[45] L. Zelnik-Manor and P. Perona, Self-tuning spectral clustering, Adv. Neural Inf. Process. Syst. **17** (2004), 1601–1608.

How to cite this article: Budd J, van Gennip Y, Latz J. Classification and image processing with a semi-discrete scheme for fidelity forced Allen–Cahn on graphs. *GAMM-Mitteilungen*. 2021;44:e202100004. https://doi.org/10.1002/gamm.202100004

APPENDIX A. AN ANALYSIS OF THE METHOD FROM [32]

In [32], the authors approximated $S_\tau u$ by a semi-implicit Euler method for fidelity forced diffusion. That is, since $S_\tau u$ is defined to equal $v(\tau)$ where v is defined by

$$\frac{dv}{dt} = -\Delta v - M(v - \tilde{f}), \qquad v(0) = u,$$

the authors of [32] approximate trajectories of this ODE by a semi-implicit Euler method with time step $\delta t > 0$ (such that $\tau/\delta t \in \mathbb{N}$), that is, $v_0 \coloneqq u$ and

$$\frac{v_{k+1} - v_k}{\delta t} = -\Delta v_{k+1} - M(v_k - \tilde{f})$$

with solution (recalling that $f \coloneqq M\tilde{f}$)

$$\begin{aligned} v_{k+1} &= (I + \delta t\Delta)^{-1}\left(v_k - \delta t M(v_k - \tilde{f})\right) \\ &= (I + \delta t\Delta)^{-1}(I - \delta t M)\, v_k + \delta t(I + \delta t\Delta)^{-1} f \end{aligned} \tag{A1}$$

and then (A1) leads to the approximation $S_\tau u \approx v_{\tau/\delta t}$. To compute (A1), they use the Nyström decomposition to compute the leading eigenvectors and eigenvalues of Δ.

Note. In fact, in [32] the authors use Δ_s not Δ. It makes no difference to this analysis which Laplacian is used.

Given $\Delta \approx U_1 \Lambda U_2^T$, an approximate SVD of low rank, the authors approximate (A1) by

$$\begin{aligned} \hat{v}_{k+1} &= U_1(I_K + \delta t\Lambda)^{-1} U_2^T(\hat{v}_k - \delta t M(\hat{v}_k - \tilde{f})) \\ &= U_1(I_K + \delta t\Lambda)^{-1} U_2^T(I - \delta t M)\hat{v}_k + \delta t U_1(I_K + \delta t\Lambda)^{-1} U_2^T f. \end{aligned} \tag{A2}$$

Therefore, the final approximation for $S_\tau u$ in [32] is computed by setting $\hat{v}_0 = u$ and $S_\tau u \approx \hat{v}_{\tau/\delta t}$.

A.1 Analysis

Both (A1) and (A2) are of the form

$$v_{k+1} = \mathcal{A} v_k + g.$$

By induction, this has kth term

$$v_k = \mathcal{A}^k v_0 + \sum_{r=0}^{k-1} \mathcal{A}^r g = \mathcal{A}^k v_0 + (\mathcal{A} - I)^{-1}(\mathcal{A}^k - I)g. \tag{A3}$$

Thus taking $k = \tau/\delta t$ and $v_0 = u$, we get successive approximations

$$\begin{aligned} S_\tau u \approx{}& \left[(I + \delta t\Delta)^{-1}(I - \delta t M)\right]^k u \\ &+ \left[(I + \delta t\Delta)^{-1}(I - \delta t M) - I\right]^{-1} \left(\left[(I + \delta t\Delta)^{-1}(I - \delta t M)\right]^k - I\right) \delta t(I + \delta t\Delta)^{-1} f \end{aligned} \tag{A4}$$

$$\begin{aligned} \approx{}& \left[U_1(I_K + \delta t\Lambda)^{-1} U_2^T(I - \delta t M)\right]^k u \\ &+ \left[U_1(I_K + \delta t\Lambda)^{-1} U_2^T(I - \delta t M) - I\right]^{-1} \left(\left[U_1(I_K + \delta t\Lambda)^{-1} U_2^T(I - \delta t M)\right]^k - I\right) \delta t U_1(I_K + \delta t\Lambda)^{-1} U_2^T f. \end{aligned} \tag{A5}$$

To see what these approximations are doing, note that

$$I + \delta t X = e^{\delta t X} + \mathcal{O}(\delta t^2)$$

and note the Lie product formula [27, Theorem 2.11]

$$e^{(X+Y)/k} = e^{X/k} e^{Y/k} + \mathcal{O}(k^{-2}) \quad \text{and therefore} \quad e^{X+Y} = \left(e^{X/k} e^{Y/k}\right)^k + \mathcal{O}(k^{-1}).$$

Then, since $k\delta t = \tau$,

$$\left[(I + \delta t\Delta)^{-1}(I - \delta t M)\right]^k = \left[e^{-\delta t\Delta} e^{-\delta t M} + \mathcal{O}(\delta t^2)\right]^k$$

$$
\begin{aligned}
&= \left[e^{-\delta t\Delta} e^{-\delta t M}\right]^k + \mathcal{O}(k\delta t^2) \\
&= \left(e^{-\tau(\Delta+M)} + \mathcal{O}(k\delta t^2)\right) + \mathcal{O}(k\delta t^2) \\
&= e^{-\tau A} + \mathcal{O}(\delta t)
\end{aligned} \tag{A6}
$$

and the second term in (A4) becomes

$$
\begin{aligned}
&\left[I - (I+\delta t\Delta)^{-1}(I-\delta t M)\right]^{-1} \left(I - e^{-\tau A} + \mathcal{O}(\delta t)\right) \delta t (I+\delta t\Delta)^{-1} \\
&\quad = (\Delta + M)^{-1}(I+\delta t\Delta)\left(I - e^{-\tau A} + \mathcal{O}(\delta t)\right)(I+\delta t\Delta)^{-1} \\
&\quad = A^{-1}(I - e^{-\tau A}) + E + \mathcal{O}(\delta t)
\end{aligned}
$$

where (writing $[X, Y] \coloneqq XY - YX$ for the commutator of X and Y)

$$
\begin{aligned}
E &\coloneqq A^{-1}(I+\delta t\Delta)\left[I - e^{-\tau A}, (I+\delta t\Delta)^{-1}\right] \\
&= -A^{-1}(I+\delta t\Delta)\left[e^{-\tau A}, (I+\delta t\Delta)^{-1}\right] \\
&= A^{-1}\left[e^{-\tau A}, (I+\delta t\Delta)\right](I+\delta t\Delta)^{-1} \\
&= \delta t A^{-1}\left[e^{-\tau A}, \Delta\right](I+\delta t\Delta)^{-1} = \mathcal{O}(\delta t)
\end{aligned}
$$

is the commutation error. Hence the overall error for (A4) is $\mathcal{O}(\delta t)$. The error for (A5) is similar but also contains an extra error from the spectrum truncation.

We can also relate this Euler method to a modified quadrature rule. It is easy to see from (2.2) that

$$
S_\tau u = e^{-\tau A} u + \int_0^\tau e^{-tA} f \, dt.
$$

We understand the Euler approximation for the $e^{-\tau A}u$ term by (A6). By (A3), we can write the Euler approximation for the integral term as

$$
\int_0^\tau e^{-tA} f \, dt \approx \delta t \sum_{r=0}^{k-1} \left[(I+\delta t\Delta)^{-1}(I-\delta t M)\right]^r (I+\delta t\Delta)^{-1} f \tag{A7}
$$

$$
\approx \delta t \sum_{r=0}^{k-1} \left[U_1 (I_K + \delta t\Lambda)^{-1} U_2^T (I - \delta t M)\right]^r U_1 (I_K + \delta t\Lambda)^{-1} U_2^T f. \tag{A8}
$$

We note that, since $M = diag(\mu)$ (and assuming that $\delta t ||\mu||_\infty < 1$),

$$
(I - \delta t M)^{-1} = diag\left((\mathbf{1} - \delta t\mu)^{-1}\right)
$$

applying the reciprocation elementwise. Therefore, we can rewrite (A7) as

$$
\begin{aligned}
b \coloneqq \int_0^\tau e^{-tA} f \, dt &\approx \delta t \sum_{r=0}^{k-1} \left[(I+\delta t\Delta)^{-1}(I-\delta t M)\right]^{r+1} \left((\mathbf{1} - \delta t\mu)^{-1} \odot f\right) \\
&= \delta t \sum_{r=1}^{k} \left(e^{-r\delta t A}\left((\mathbf{1} - \delta t\mu)^{-1} \odot f\right) + \mathcal{O}(r\delta t^2)\right) \qquad \text{by (A6) and relabeling } r \\
&= \left(\delta t \sum_{r=1}^{k} e^{-r\delta t A}\left((\mathbf{1} - \delta t\mu)^{-1} \odot f\right)\right) + \mathcal{O}(\tau^2 \delta t)
\end{aligned}
$$

recalling that $k\delta t = \tau$. This can be seen to be a quadrature by the right-hand rule of the integral

$$
\int_0^\tau e^{-tA}\left((\mathbf{1} - \delta t\mu)^{-1} \odot f\right) \, dt.
$$

Likewise, we can rewrite (A8) as

$$\int_0^\tau e^{-tA} f \, dt \approx \delta t \sum_{r=1}^{k} \left[U_1 (I_K + \delta t \Lambda)^{-1} U_2^T (I - \mu \delta t P_Z) \right]^r \left((\mathbf{1} - \delta t \mu)^{-1} \odot f \right)$$

where going from (A7) to (A8) has incurred an extra error from the spectrum truncation alongside the quadrature and Lie product formula errors.

A.2 Conclusions from analysis

The key takeaway from these calculations (besides the verification that we have the usual $\mathcal{O}(\delta t)$ Euler error) concerns (A6). That equation shows that the Euler approximation for the $e^{-\tau A}$ term is in fact an approximation of a Lie product formula approximation for $e^{-\tau A}$. This motivates our method of cutting out the middleman and using a matrix exponential formula directly, and furthermore motivates our replacement of the linear error Lie product formula with the quadratic error Strang formula.

We have also shown how the Euler method approximation for b can be written as a form of quadrature, motivating our investigation of other quadrature methods as potential improvements for computing b.