Theory and Applications of Differential Equation Methods for Graph-based Learning

Budd, J.M.

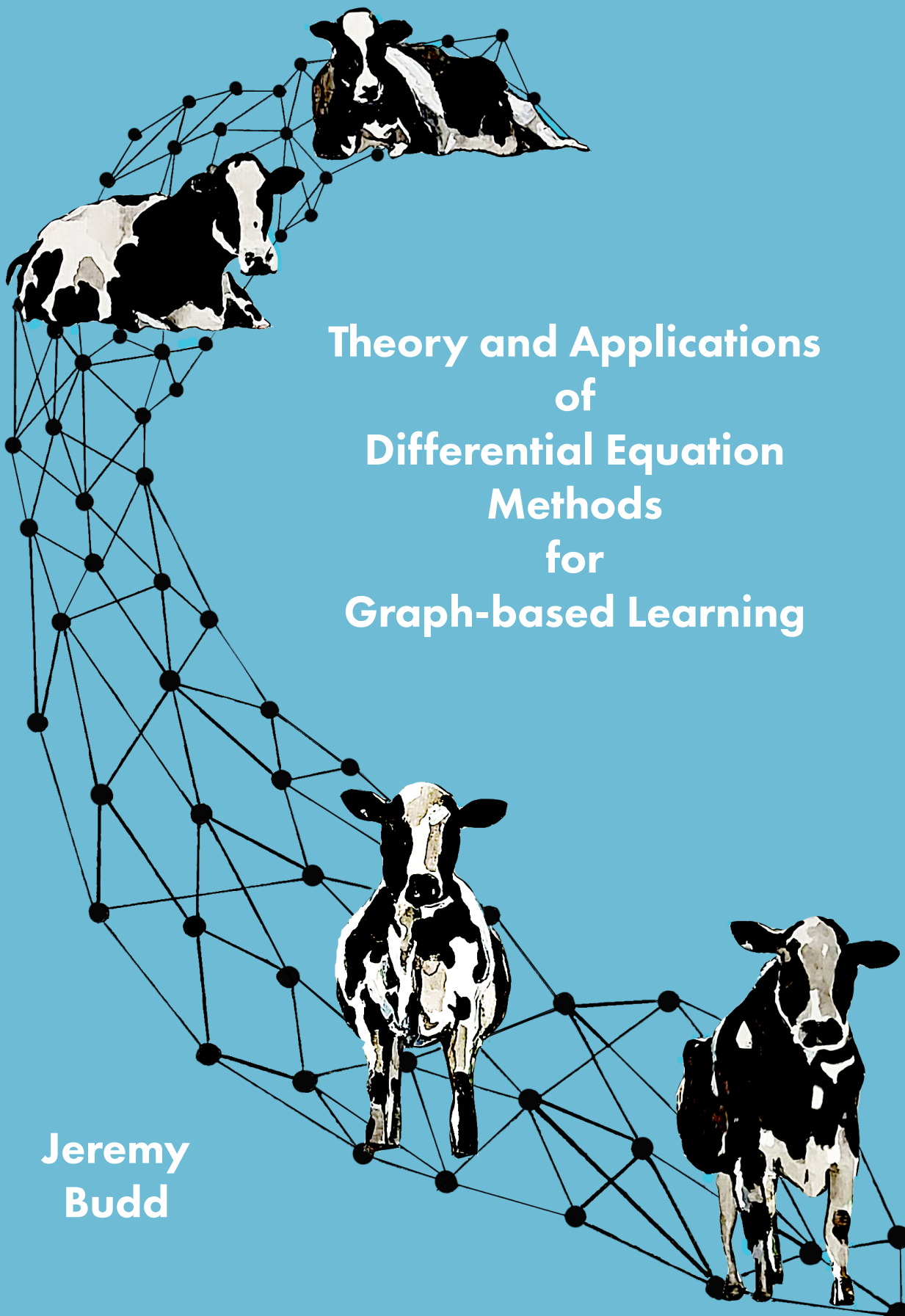
DOI
10.4233/uuid:b8e0648c-d38e-4f95-bcd7-b99a943cb2d1

Publication date
2022

Document Version
Final published version

Citation (APA)
Budd, J. M. (2022). *Theory and Applications of Differential Equation Methods for Graph-based Learning*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:b8e0648c-d38e-4f95-bcd7-b99a943cb2d1

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



Theory and Applications of Differential Equation Methods for Graph-based Learning

Jeremy Budd

Theory and Applications of Differential Equation Methods for Graph-based Learning

Theory and Applications of Differential Equation Methods for Graph-based Learning

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof.dr.ir. T.H.J.J. van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op dinsdag 25 Januari 2022 om 15:00 uur

door

Jeremy Michael BUDD

Master of Mathematics,
University of Cambridge, Verenigd Koninkrijk
geboren te Bristol, Verenigd Koninkrijk.

Dit proefschrift is goedgekeurd door de

promotor: Dr. J.L.A. Dubbeldam
copromotor: Dr. Y. van Gennip

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Dr. J.L.A. Dubbeldam,	Technische Universiteit Delft, promotor
Dr. Y. van Gennip,	Technische Universiteit Delft, copromotor
Onafhankelijke leden:	
Prof. dr. J.M.A.M. van Neerven	Technische Universiteit Delft
Prof. dr. H.M. Schuttelaars	Technische Universiteit Delft
Prof. dr. C.-B. Schönlieb	University of Cambridge, Verenigd Koninkrijk
Prof. dr. F.K.O. Hoffmann	Rheinische Friedrich-Wilhelms-Universität Bonn, Duitsland
Prof. dr. ir. A.W. Heemink	Technische Universiteit Delft, reservelid

Dit project heeft financiering ontvangen van Horizon 2020 onderzoek en innovatie van de Europese Unie programma in het kader van Marie Skłodowska-Curie subsidieovereenkomst nr. 777826.

Horizon 2020
European Union Funding
for Research & Innovation

Keywords: Graph dynamics, Allen–Cahn equation, Ginzburg–Landau functional, Merriman–Bence–Osher scheme, double-obstacle potential, mass conservation, fidelity constraint, mean curvature flow, image segmentation, joint reconstruction-segmentation.

Printed by: Proefschriften Printen

Front & Back: Bryony Budd

ISBN 978-90-832173-4-5

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

Some mathematicians are birds, others are frogs. Birds fly high in the air and survey broad vistas of mathematics out to the far horizon. They delight in concepts that unify our thinking and bring together diverse problems from different parts of the landscape. Frogs live in the mud below and see only the flowers that grow nearby. They delight in the details of particular objects, and they solve problems one at a time.

Freeman Dyson, *Birds and Frogs*

Contents

Summary

A large number of modern learning problems involve working with highly interrelated and interconnected data. Graph-based learning is an emerging technique for approaching such problems, by representing this data as a graph (a.k.a. a network). That is, the points of data are represented by the vertices of the graph, and then the edges linking these vertices represent the relationships between the points of data. This provides a unified perspective for thinking about all sorts of interrelated data: the vertices could represent pixels in an image or people in a social network, and the underlying framework would be the same.

Graph-based learning is a very mathematically rich field, and so in this thesis we shall be focusing on just one strand of this technique: the use of "PDEs on graphs" to solve learning problems. The topic of what exactly that means will be addressed in chapter 2, in which we will present our framework for analysis on graphs. The contributions of this thesis to this strand of graph-based learning are fourfold, two theoretical and two more applications-driven.

Our first, and foremost, contribution is the rigorous link we prove between Allen–Cahn (AC) flow and the Merriman–Bence–Osher (MBO) diffusion-thresholding scheme on graphs. These are two diffusion-based flows which have frequently been used in applications, broadly interchangeably, motivated by their known link in the continuum. Our key theorem is that for a specific choice of potential for the AC flow, the MBO scheme is a special case of what we call a semi-discrete implicit Euler (SDIE) scheme for AC flow. We furthermore show that this link is robust to the inclusion of further constraints, namely mass conservation and fidelity forcing. We also prove a number of other key results. First, we prove that the AC flow with this potential (which is not a differentiable function) has existence, uniqueness, and Lipschitz regularity of its solutions. Second, we solve the SDIE scheme, which in the mass-conserving case involved extensive use of tools from convex optimisation. We find that the SDIE scheme is in general a diffusion-thresholding scheme, but with the "hard" step-function thresholding of the MBO scheme relaxed to a "soft" piecewise linear thresholding as the time step gets smaller. Finally, we prove that as the time step tends to zero the SDIE scheme sequence converges to a trajectory of AC flow. This justifies thinking of it as a scheme for AC flow and also allows us to prove further results about this flow. In particular, we prove that the flow monotonically decreases the graph Ginzburg–Landau energy and we prove (in all except the mass-conserving case, in which the question remains open) that the flow is well-posed.

Our second contribution concerns the use of these methods for the task of image segmentation, i.e. the task of splitting an image into its component features. As indicated above, an image can be encoded as a finite graph defined on the set of its pixels, with the edge weight between two pixels a function of their similar-

ity. Given this graph, repeated iterations of the fidelity-forced MBO (or in general SDIE) scheme produce a series of binary segmentations of the image. However, in practice it is computationally unfeasible to compute such iterations exactly, due to the large size of the matrices involved in the graph diffusion. Our key contribution in this area is the investigation of two ideas to overcome this obstacle, which refine previous techniques. The first idea is to use the Nyström decomposition alongside a QR factorisation method to approximate the leading eigenvalues and eigenvectors of the graph Laplacian. The second idea is to use a Strang formula method to use this approximate eigendecomposition to compute the graph diffusion. We perform numerical experiments on a toy image to quantify the accuracy, speed, and reliability of these methods. Finally, we deploy this algorithm to segment a standard reference image from the literature. Our refinements lead to a substantially improved segmentation of this image compared to previous work.

Our third contribution is the incorporation of these graph-based segmentation methods into a technique for joint reconstruction-segmentation. Reconstruction-segmentation is the task of segmenting an image given indirect, noisy, and/or damaged observations of that image. Classically this task was performed sequentially: first reconstructing the image from the observations, and then segmenting the reconstructed image. A more recent technique in this area instead performs the reconstruction and segmentation jointly, leading to better results. However, previous work on this has typically made use of relatively simple segmentation methods. We devise a novel framework for joint reconstruction-segmentation on graphs, incorporating the graph segmentation technique described in the previous paragraph. We then test this for a denoising-segmentation task on an artificially noised version of the standard reference image. This work lays the foundation for an ongoing project, for which we discuss future steps.

Finally, we will consider a third flow, mean curvature flow (MCF), which is well-known to be related to AC flow and the MBO scheme in the continuum setting. This raises an open question: in what manner is MCF also related to AC flow and the MBO scheme on a graph? Before this question can be answered, we must first ask an even more basic question: how can we define MCF on a graph? Our key contribution is showing that a previous definition of graph MCF has a fatal flaw which scuppers any chance of it being related to diffusion-based flows like AC flow and the MBO scheme. We furthermore propose a new definition of graph MCF which avoids the flaw, and show that this newly defined MCF perfectly resembles the MBO scheme up to $\mathcal{O}(\tau^2)$ terms (for τ the time step in the MBO scheme/the MCF).

Samenvatting

Voor vele moderne problemen in machinaal leren is het nodig om met onderling sterk gerelateerde en hoog-dimensionale data te werken. Leren op grafen, waarbij de data gemodelleerd wordt als een graaf (d.w.z. een netwerk), is een opkomende techniek die voor zulke problemen gebruikt kan worden. De knopen van de graaf stellen hierbij de datapunten voor en de bogen die deze knopen verbinden de onderlinge verbanden tussen de datapunten. Dit geeft ons een verenigd perspectief om over verschillende soorten data en hun verbanden na te denken: zo is het onderliggende raamwerk hetzelfde, of de knopen nu pixels van een afbeelding voorstellen of personen in een sociaal netwerk.

Leren op grafen is een discipline die rijk is aan wiskunde. Daarom ligt de focus in dit proefschrift op slechts één deelgebied ervan: het gebruik van "partiële differentiaalvergelijkingen op grafen" om problemen in machinaal leren op te lossen. Wat dat precies betekent, komt aan bod als we in hoofdstuk 2 ons raamwerk presenteren voor analyse op grafen. Dit proefschrift bevat vier bijdragen aan dit deelgebied, twee theoretische en twee geïnspireerd door toepassingen.

Onze eerste en belangrijkste bijdrage is het rigoreuze verband dat we bewijzen tussen de Allen–Cahn (AC) stroming en het Merriman–Bence–Osher (MBO) diffusie-drempelwaardeproces op grafen. Dit zijn beide op diffusie gebaseerde stromingen die regelmatig, min of meer uitwisselbaar, gebruikt zijn in toepassingen vanwege het al bekende verband tussen beide stromingen in hun continuüm formulering. Ons voornaamste resultaat stelt dat het MBO-proces een speciaal geval is van wat wij een semi-discreet impliciet Euler (SDIE) proces voor AC-stroming noemen, mits er in de AC-stroming een specifieke potentiaal gebruikt wordt. Verder tonen we ook aan dat dit verband standhoudt, als er verdere beperkingen aan de stromingen worden opgelegd, zoals massabehoud of getrouwheid aan vooraf gegeven informatie. We bewijzen ook enkele andere belangrijke resultaten. Ten eerste bewijzen we dat de AC-stroming met de specifieke potentiaal (die niet differentieerbaar is) een unieke, Lipschitz-continue oplossing heeft. Ten tweede geven we oplossingen voor het SDIE-proces, waarbij we in het massabehoudende geval intensief gebruik maken van convexe optimalisatie. We ontdekken dat het algemene SDIE-proces een diffusie-drempelwaardeproces is, waarbij de "harde" stapfunctie-drempelwaarde van het MBO-proces afgezwakt wordt naar een "zachte" stuksgewijs lineaire drempelwaarde als de tijdstap kleiner wordt. Tenslotte bewijzen we dat het SDIE-proces convergeert naar een traject van de AC-stroming. Dit rechtvaardigt de opvatting dat het SDIE-proces een proces voor AC-stroming is en staat ons ook toe om nog meer resultaten te bewijzen voor de AC-stroming. Zo bewijzen we de monotone afname van de Ginzburg–Landau-energie op grafen langs trajecten van deze stroming en we bewijzen (behalve in het massabehoudende geval, waar dit nog een open vraag is) dat de stroming een correct gesteld probleem is.

Onze tweede bijdrage heeft te maken met het gebruik van deze methodes voor beeldsegmentatie, d.w.z. het opdelen van een afbeelding in verschillende delen die relevante structuren bevatten. Zoals hierboven aangegeven, kan een afbeelding voorgesteld worden door een eindige graaf gedefinieerd op de verzameling van pixels, waarbij het gewicht van de bogen een functie is van de gelijkenis tussen de corresponderende pixels. Gegeven deze graaf, produceren herhaalde iteraties van het MBO- (of in het algemeen, het SDIE-) proces met getrouwheid aan vooraf gegeven informatie een reeks binaire segmentaties van de afbeelding. In de praktijk is het, vanwege de grootte van de matrices die nodig zijn voor het diffusieproces op de graaf, computationeel echter niet haalbaar om zulke iteraties exact te berekenen. Onze hoofdbijdrage op dit gebied bestaat uit twee aanpassingen van bestaande technieken om dit obstakel te overwinnen. De eerste aanpassing is het gebruik van de Nyström-decompositie samen met een QR-decompositie om de eerste eigenwaarden en bijbehorende eigenvectoren te berekenen van de Laplace-matrix van de graaf. De tweede aanpassing is het gebruik van een formule van Strang om met behulp van deze benaderende eigendecompositie de diffusie op de graaf te berekenen. We voeren numerieke experimenten uit op een *toy model* (speelgoedmodel) om de accuraatheid, snelheid en betrouwbaarheid van deze methodes to kunnen kwantificeren. Tenslotte passen we dit algoritme toe om een standaard referentie-afbeelding uit de literatuur te segmenteren. Vergeleken met eerder werk, geven onze aanpassingen een aanzienlijk betere segmentatie van deze afbeelding.

Onze derde bijdrage is het inpassen van deze op grafen gebaseerde segmentatiemethodes in een gezamenlijke reconstructie-segmentatietechniek. De reconstructie-segmentatietaak bestaat uit het segmenteren van een afbeelding gegeven, mogelijk indirecte, observaties van de afbeelding die ruis kunnen bevatten en/of beschadigd kunnen zijn. Klassieke methodes voeren de twee onderdelen van deze taak na elkaar uit: eerst reconstructie, dan segmentatie van de gereconstrueerde afbeelding. Een recentere techniek op dit gebied voert daarentegen de reconstructie en segmentatie tegelijkertijd uit, wat tot betere resultaten leidt. Echter, eerder werk op dit gebied maakte doorgaans gebruik van relatief eenvoudige segmentatiemethodes. Wij bedenken een nieuw raamwerk voor gezamenlijke reconstructie-segmentatie op grafen, dat gebruik maakt van de segmentatietechniek op grafen die we in de vorige paragraaf beschreven. Dit testen wij dan op een kunstmatig van ruis voorziene versie van de standaard referentie-afbeelding die zowel van ruis ontdaan als gesegmenteerd dient te worden. Dit werk legt een fundament voor een lopend project, waarvoor we toekomstige stappen bespreken.

Ten slotte besteden we aandacht aan een derde stroming, namelijk stroming volgens de gemiddelde kromming (GK-stroming), waarvan het in de continuümcontext bekend is, dat deze gerelateerd is aan AC-stroming en het MBO-proces. Dit roept een nog altijd open vraag op: hoe is GK-stroming gerelateerd aan AC-stroming en het MBO-proces op een graaf? Voordat deze vraag beantwoord kan worden, moeten we eerst een fundamentelere vraag stellen: hoe kunnen we GK-stroming op een graaf definiëren? Onze voornaamste bijdrage is het aantonen dat een eerdere definitie van GK-stroming op een graaf een fatale fout bevat, die elke hoop dat het verband heeft met op diffusie gebaseerde stromingen zoals AC-stroming en het

MBO-proces torpedeert. Verder stellen we een nieuwe definitie van GK-stroming op grafen voor die deze fout vermijdt en we laten zien dat deze nieuwe GK-stroming op $\mathcal{O}(\tau^2)$-termen na overeenkomt met het MBO-proces (waar τ de tijdstap is in het MBO-process/de GK-stroming).

1

Introduction

The story so far: In the beginning the Universe was created.
This has made a lot of people very angry and been widely regarded as a bad move.

Douglas Adams, *The Restaurant at the End of the Universe*

1

To understand a piece of information, one needs to consider it in context, to consider how it forms part of a larger web of interrelated information. A pixel by itself tells you very little, but when multiple pixels come together they form an image, and when multiple images come together they can tell a story. A natural mathematical way to represent such information is as a *graph*, as a set of vertices linked by edges, where the vertices encode individual pieces of information (e.g., pixels) and the edges encode the relationships between those pieces of information. This idea is the starting point for the emerging technique of *graph-based learning*, which uses such graphical representations to solve learning problems, such as clustering and classification. One strand of this technique is to solve such problems by putting a PDE-like flow onto the graph, which will induce a *phase separation* of the signal from the background.

This thesis will investigate some of the theoretical underpinnings of this strand, as well as exploring and developing its use in applications. We will consider a pair of flows, (namely the Allen–Cahn (AC) flow and the Merriman–Bence–Osher (MBO) scheme) which have been used interchangeably in graph-based learning, developing the theory of these flows and showing that they can be rigorously linked together. Furthermore, we will show that this link is robust to the addition of important application relevant constraints, namely mass conservation and fidelity forcing. Next, incorporating ideas from this theory, we will consider the application of these methods to the specific learning problem of image segmentation, in which we shall improve upon past approaches. Taking this thread further, we will consider the task of *reconstruction-segmentation*, for which a powerful technique is *joint reconstruction-segmentation*, which performs the reconstruction and segmentation simultaneously. We will develop a novel method which incorporates (to the authors' knowledge, for the first time) joint reconstruction-segmentation within the framework of graph-based learning. Finally, we will consider how *mean curvature flow* can be defined on a graph, and indicate how this definition appears to be linked to the two flows which have been the focus of our work.

1.1. A bird's eye view of graph-based learning

To motivate the technique of graph-based learning, let us describe a generic discrete clustering/classification problem. Let V be a set of individual entities (these could be pixels in an image, or whole images, or words, or people, etc.) which bear some relations to each other. We want to find some function u on V which sends each $i \in V$ to an appropriate *class*. This function is often called a *labelling function*. Examples of such problems include: Which pixels in an image belong to a cow (see chapter 5)? Which images in a data set are pictures of dogs vs. pictures of cats [31]? Which words in a set of texts belong to which topics [2]? Or, which people in a social network belong to certain social groups [37]? To aid us, in classification problems we will have a (potentially very small) subset Z of V, the elements of which will have already been labelled by a labelling function f.

The perspective of graph-based learning is to encode V as the vertex set of a graph, and encode the relationships between elements of V as edges of that graph, weighted according to the strength of those relationships. Then to solve a

classification task, the idea is to use this graph structure to *propagate* the labels on the *a priori* labelled set Z to the entirety of V. We shall now describe three of the major strands of graph-based learning approaches to such problems, in broadly chronological order.

The earliest work in this area is the method of *spectral clustering*. This was popularised as a method for machine learning by Shi and Malik [33] and Ng, Jordan, and Weiss [30] in the early 2000s, but dates back much earlier, e.g. work by Donath and Hoffmann [15] in the early 70s. Spectral clustering works by first constructing the first k eigenvectors $(\xi^\ell)_{\ell=1}^k$ of the graph Laplacian Δ[1] of the graph generated on our data, and then for each $i \in V$ associates to i the vector $(\xi_i^1, ..., \xi_i^k)$ (i.e. the i^{th} component in each of the k eigenvectors). Next, one deploys a standard clustering method (e.g., k-means [25]) to cluster the vertices based on these vectors. In effect, one uses the graph structure to perform a dimensionality reduction of the data before performing a clustering. For an overview and analysis of this method, see the "tutorial" by von Luxburg [27]. This technique of solving clustering problems via graph-based embeddings has received considerable attention, for recent work see e.g. García Trillos, Hoffmann, and Hosseini [36].

Next, some of the original pioneering work in graph-based learning for classification problems was the *Laplace learning* technique of Zhu *et al.* [39], also from the early 2000s. This is a label propagation method, which extends the labels harmonically via the graph Laplacian. That is, it finds a labelling function u solving:

$$\Delta u = 0 \text{ on } V \setminus Z, \qquad u = f \text{ on } Z.$$

This technique has seen wide application, e.g. in [5, 38]. It has also proved to be highly fertile, with attempts to solve an issue at very low label rates (see El Alaoui *et al.* [3]) leading to the exploration of using graph p-Laplacians (e.g. [34]), ∞-Laplacians (e.g. [23]), weighted Laplacians (e.g. [10]), and most recently the so-called *Poisson learning* [11].

Last, but by no means least, is the method of *PDEs on graphs*, which shall be the focus of this thesis. Pioneering works in this strand were those of Ta, Elmoataz, and Lézoray [35] in 2011, using morphological PDEs such as the Eikonal equation; Bertozzi and Flenner [8] in 2012, using the AC flow; and Merkurjev, Kostić, and Bertozzi [29] in 2013, using the MBO threshold dynamics scheme. These methods were extended to multi-class classification in Desquesnes, Elmoataz, and Lézoray [14] and Garcia-Cardona *et al.* [20], and also received extensive theoretical study in e.g. [6, 18, 21, 26]. Applications of these methods can be found in e.g. [9, 22, 24, 32]. Our main interest will be the interchangeable use of AC flow and the MBO scheme in the work by Bertozzi and co-authors. This was motivated heuristically by the well-known links in the continuum between these flows via mean curvature flow (MCF) (definitions of MCF on a graph were proposed in [17] and [21], see chapter 7 for details), but was not rigorously supported. One of the key results of this thesis is to provide that rigorous support.

Note 1. *A further, more theoretical strand of research in this area concerns the*

[1] We will define the graph Laplacian and other graph analysis concepts in chapter 2.

continuum limits *of these graphs, linking together the discrete and continuum perspectives. A major topic in this strand is the study of the consistency of the above methods in the large-data limit. As this work lies outside of the scope and consideration of this thesis, we shall here simply refer the reader to e.g.* [13, 16, 19, 28].

1.2. Chapter outline

In chapter 2, we will introduce the framework for analysis on graphs within which the rest of this work will reside. We will in this work restrict ourselves to a certain subset of graphs (namely finite, undirected, simple, connected graphs with non-negative edge weights), and we will briefly discuss in the chapter the significance of these assumptions for our framework.

In chapter 3, we will define in a graph setting the PDE-like flows we shall be considering, namely AC flow and the MBO scheme. In particular, we shall define and investigate the properties of graph AC flow with the *double-obstacle potential*, and for both AC flow and the MBO scheme we shall also consider either mass conservation or fidelity forcing constraints. We will then introduce our key original contribution, a *semi-discrete implicit Euler (SDIE)* scheme for AC flow. Finally, we will investigate the properties of this double-obstacle AC flow. In particular, we shall prove conditions under which the flow "freezes", prove uniqueness of solutions, and state a number of other key properties (including existence and Lipschitz regularity of solutions) which we shall prove in chapter 4.

In chapter 4, we will present the key theoretical results of this thesis. Our key result will be that the SDIE scheme for double-obstacle AC flow, including under either the mass conservation or fidelity forcing constraints, is equivalent to a variational scheme which has the MBO scheme as a special case. We explicitly characterise the solutions to this SDIE scheme, showing that in general they correspond to a piecewise linear relaxation of the step-function thresholding in the MBO scheme. We exhibit a Lyapunov functional for the SDIE scheme, and use this to investigate the scheme's long-time behaviour. Finally, we will show that as the time step tends to zero, the SDIE trajectories converge to trajectories of AC flow. We then use this convergence result to prove the properties of AC flow which were only stated in 3.

In chapter 5, we will consider the use of this SDIE scheme as a method for image segmentation. We will begin by discussing how to represent an image as a finite graph. Next, we describe the basic algorithm for image segmentation using the SDIE scheme. However, this algorithm is computationally unfeasible, due to the very large size of the matrices involved. We describe two ideas to get around this obstacle, which refine previously used techniques: first, we use a Nyström-QR method based on Bebendorf and Kunis [7] to approximate the leading eigenvalues and eigenvectors of the graph Laplacian, and second we use a Strang formula method to use this approximate decomposition to compute the graph diffusion. We perform numerical experiments on a toy image to quantify the accuracy, speed, and reliability of these methods. Finally, we deploy this algorithm to segment the "two cows" image that was also segmented in [8] and [29], as well as two related examples. We will discover that whilst for these examples the SDIE scheme had

best performance in its MBO special case, our other refinements will lead to a substantially improved segmentation compared to [8, 29].

In chapter 6, we will consider the task of reconstruction-segmentation, i.e. the task of both reconstructing an image from indirect, noisy, and/or damaged observations, and also segmenting that image. Our interest will be focused on the powerful technique of *joint reconstruction-segmentation* (which has recently seen increasing attention [1, 12]), which performs this task by performing the reconstruction and segmentation simultaneously. Previous work on this technique has made use of relatively simple segmentation methods. We will devise a novel framework for incorporating the graph PDE-based segmentation method of chapter 5 within a joint reconstruction-segmentation technique. We will then demonstrate this technique for a denoising-segmentation task on an artificially noised version of the "two cows" image. Finally, as this work lays the foundation for ongoing work, we will discuss a number of directions for future research.

Finally, in chapter 7 we will consider a question raised by Van Gennip *et al.* [21], namely that of the relationship between AC flow, the MBO scheme, and *mean curvature flow* (MCF) on graphs. We will first review the relationship these flows have in the continuum, and review (and slightly extend) the promising Γ-convergence results which suggest such a link in the graph context. Next, we will show that the definition of graph MCF offered by [21] has a key flaw, and that because of that flaw the MCF cannot resemble diffusion-based flows on a general graph. Finally, we will propose a new definition of graph MCF which avoids the flaw, and show that this flow perfectly resembles the MBO scheme up to $\mathcal{O}(\tau^2)$ terms (for τ the time step in the MCF and the MBO scheme).

1.3. Key contributions of this thesis

1. We defined a graph AC flow with the double-obstacle potential, including either mass conservation or fidelity forcing constraints, and proved various desirable properties of this flow, including existence and uniqueness of solutions, and monotonic decrease of the Ginzburg–Landau energy. (Chapter 3)

2. We introduced the SDIE scheme for graph double-obstacle AC flow, proved that this scheme has the MBO scheme as a special case, and proved that this scheme in general has solutions corresponding to a diffusion for a time step followed by a piecewise linear thresholding. We also show that as the time step tends to zero the SDIE trajectories converge to trajectories of the AC flow. Hence, this SDIE scheme "interpolates" between the double-obstacle AC flow and the MBO scheme. (Chapter 4)

3. We investigated numerically the virtues of the Nyström-QR method recommended by Alfke *et al.* [4] for approximately eigendecomposing the graph Laplacian, and compared it to the previous Nyström method used by e.g. [8, 29]. (Chapter 5)

4. We introduced, and investigated numerically, a novel Strang formula method

for using the eigendecomposition of the graph Laplacian to compute fidelity-forced graph diffusion. (Chapter 5)

5. We incorporated contributions 3 and 4 into an image segmentation algorithm using the SDIE scheme, which in the MBO special case outperformed earlier graph-based segmentation algorithms on a standard test image from the literature. (Chapter 5)

6. We introduced a novel graph-based framework for performing joint reconstruction-segmentation using the graph SDIE-based segmentation algorithm from contribution 5. (Chapter 6)

7. We refined the Van Gennip *et al.* [21] definition of graph mean curvature flow to avoid a key flaw, and showed that this new graph mean curvature flow formally resembles the graph MBO scheme. (Chapter 7)

Bibliography

[1] Jonas Adler et al. *Task adapted reconstruction for inverse problems*. 2018. arXiv: 1809.00948 [cs.CV].

[2] Charu C Aggarwal and ChengXiang Zhai. "A survey of text classification algorithms". In: *Mining text data*. Springer, 2012, pp. 163–222.

[3] Ahmed El Alaoui et al. "Asymptotic behavior of ℓ_p-based Laplacian regularization in semi-supervised learning". In: *29th Annual Conference on Learning Theory*. Ed. by Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir. Vol. 49. Proceedings of Machine Learning Research. Columbia University, New York, New York, USA: PMLR, 2016, pp. 879–906. URL: http://proceedings.mlr.press/v49/elalaoui16.html.

[4] Dominik Alfke et al. "NFFT Meets Krylov Methods: Fast Matrix-Vector Products for the Graph Laplacian of Fully Connected Networks". In: *Frontiers in Applied Mathematics and Statistics* 4 (2018), p. 61. ISSN: 2297-4687. DOI: 10.3389/fams.2018.00061. URL: https://www.frontiersin.org/article/10.3389/fams.2018.00061.

[5] Rie Kubota Ando and Tong Zhang. "Learning on Graph with Laplacian Regularization". In: *Proceedings of the 19th International Conference on Neural Information Processing Systems*. NIPS'06. Canada: MIT Press, 2006, pp. 25–32.

[6] Egil Bae and E. Merkurjev. "Convex Variational Methods on Graphs for Multiclass Segmentation of High-Dimensional Data and Point Clouds". In: *Journal of Mathematical Imaging and Vision* 58 (2017), pp. 468–493.

[7] M. Bebendorf and S. Kunis. "Recompression techniques for adaptive cross approximation". In: *Journal of Integral Equations and Applications* 21.3 (2009), pp. 331–357. DOI: 10.1216/JIE-2009-21-3-331.

[8] Andrea Bertozzi and Arjuna Flenner. "Diffuse Interface Models on Graphs for Classification of High Dimensional Data". In: *Multiscale Modeling Simulation* 10 (July 2012), pp. 1090–1118. DOI: 10.1137/11083109X.

[9] Luca Calatroni et al. "Graph Clustering, Variational Image Segmentation Methods and Hough Transform Scale Detection for Object Measurement in Images". In: *Journal of Mathematical Imaging and Vision* 57 (Feb. 2017), pp. 269–291. DOI: 10.1007/s10851-016-0678-0.

[10] J. Calder and D. Slepčev. "Properly-Weighted Graph Laplacian for Semi-supervised Learning". In: *Applied Mathematics & Optimization* 82.3 (2020), pp. 1111–1159. DOI: 10.1007/s00245-019-09637-3.

[11] J. Calder et al. "Poisson Learning: Graph Based Semi-Supervised Learning At Very Low Label Rates". English. In: *Proceedings of the International Conference on Machine Learning*. 2020, pp. 8588–8598.

[12] Veronica Corona et al. "Enhancing joint reconstruction and segmentation with non-convex Bregman iteration". In: *Inverse Problems* 35.5 (2019), p. 055001. DOI: 10.1088/1361-6420/ab0b77.

[13] Riccardo Cristoferi and Matthew Thorpe. "Large Data Limit for a Phase Transition Model with the p-Laplacian on Point Clouds". In: *European Journal of Applied Mathematics* 31.2 (Nov. 2018), pp. 185–231. ISSN: 0956-7925.

[14] Xavier Desquesnes, A. Elmoataz, and O. Lézoray. "Eikonal Equation Adaptation on Weighted Graphs: Fast Geometric Diffusion Process for Local and Non-local Image and Data Processing". In: *Journal of Mathematical Imaging and Vision* 46 (2013), pp. 238–257. DOI: 10.1007/s10851-012-0380-9.

[15] William Donath and Alan Hoffman. "Algorithms for partitioning of graphs and computer logic based on eigenvectors of connections matrices". In: *IBM Technical Disclosure Bulletin* (1972).

[16] Matthew M. Dunlop et al. "Large data and zero noise limits of graph-based semi-supervised learning algorithms". In: *Applied and Computational Harmonic Analysis* 49.2 (2020), pp. 655–697. ISSN: 1063-5203. DOI: 10.1016/j.acha.2019.03.005. URL: https://www.sciencedirect.com/science/article/pii/S1063520318301398.

[17] Abdallah El Chakik, Abderrahim Elmoataz, and Xavier Desquesnes. "Mean curvature Flows on Graphs for Image and Manifold Restoration and Enhancement". In: *Signal Processing* 105 (Dec. 2014), pp. 449–463. DOI: 10.1016/j.sigpro.2014.04.029.

[18] A. Elmoataz, F. Lozes, and M. Toutain. "Nonlocal PDEs on Graphs: From Tug-of-War Games to Unified Interpolation on Images and Point Clouds". In: *Journal of Mathematical Imaging and Vision* 57 (2016), pp. 381–401.

[19] Nicolás García Trillos et al. "Error Estimates for Spectral Convergence of the Graph Laplacian on Random Geometric Graphs Toward the Laplace–Beltrami Operator". In: *Foundations of Computational Mathematics* 20 (Jan. 2020), pp. 827–887. DOI: 10.1007/s10208-019-09436-w.

[20] Cristina Garcia-Cardona et al. "Multiclass Data Segmentation Using Diffuse Interface Methods on Graphs". In: vol. 36. 8. 2014, pp. 1600–1613. DOI: 10.1109/TPAMI.2014.2300478.

[21] Y. van Gennip et al. "Mean Curvature, Threshold Dynamics, and Phase Field Theory on Finite Graphs". In: *Milan Journal of Mathematics* 82 (2014), pp. 3–65.

[22] Geoffrey Iyer, Jocelyn Chanussot, and Andrea L. Bertozzi. "A Graph-Based Approach for Data Fusion and Segmentation of Multimodal Images". In: *IEEE Transactions on Geoscience and Remote Sensing* 59.5 (2021), pp. 4419–4429. DOI: 10.1109/TGRS.2020.2971395.

[23] Rasmus Kyng et al. "Algorithms for Lipschitz Learning on Graphs". In: *Proceedings of The 28th Conference on Learning Theory*. Ed. by Peter Grünwald, Elad Hazan, and Satyen Kale. Vol. 40. Proceedings of Machine Learning Research. Paris, France: PMLR, 2015, pp. 1190–1223. URL: http://proceedings.mlr.press/v40/Kyng15.html.

[24] Hao Li et al. "PDEs on graphs for semi-supervised learning applied to first-person activity recognition in body-worn video". In: *Discrete Continuous Dynamical Systems* 41.9 (2021), pp. 4351–4373.

[25] S. Lloyd. "Least squares quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137. DOI: 10.1109/TIT.1982.1056489.

[26] Xiyang Luo and Andrea Bertozzi. "Convergence of the Graph Allen–Cahn Scheme". In: *Journal of Statistical Physics* 167 (May 2017), pp. 934–958. DOI: 10.1007/s10955-017-1772-4.

[27] Ulrike von Luxburg. "A tutorial on spectral clustering". In: *Statistics and Computing* 28.4 (2007), pp. 395–416. DOI: 10.1007/s11222-007-9033-z.

[28] Ulrike von Luxburg, Mikhail Belkin, and Olivier Bousquet. "Consistency of spectral clustering". In: *The Annals of Statistics* 36.2 (2008), pp. 555–586. DOI: 10.1214/009053607000000640.

[29] Ekaterina Merkurjev, Tijana Kostić, and Andrea Bertozzi. "An MBO Scheme on Graphs for Classification and Image Processing". In: *SIAM Journal on Imaging Sciences* 6 (Oct. 2013), pp. 1903–1910. DOI: 10.1137/120886935.

[30] Andrew Ng, Michael Jordan, and Yair Weiss. "On Spectral Clustering: Analysis and an algorithm". In: *Advances in Neural Information Processing Systems*. Ed. by T. Dietterich, S. Becker, and Z. Ghahramani. Vol. 14. MIT Press, 2002. URL: https://proceedings.neurips.cc/paper/2001/file/801272ee79cfde7fa5960571fee36b9b-Paper.pdf.

[31] Omkar M Parkhi et al. "Cats and dogs". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 3498–3505. DOI: 10.1109/CVPR.2012.6248092.

[32] Yiling Qiao et al. "Uncertainty quantification for semi-supervised multi-class classification in image processing and ego-motion analysis of body-worn videos". In: *Electronic Imaging* 2019.11 (2019), pp. 264-1-264–7. ISSN: 2470-1173. DOI: doi:10.2352/ISSN.2470-1173.2019.11.IPAS-264. URL: https://www.ingentaconnect.com/content/ist/ei/2019/00002019/00000011/art00015.

[33] Jianbo Shi and J. Malik. "Normalized cuts and image segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.8 (2000), pp. 888–905. DOI: 10.1109/34.868688.

[34] D. Slepčev and M. Thorpe. "Analysis of p-Laplacian Regularization in Semi-Supervised Learning". In: *SIAM Journal on Mathematical Analysis* 51.3 (2019), pp. 2085–2120. DOI: 10.1137/17M115222X.

[35] Vinh-Thong Ta, Abderrahim Elmoataz, and Olivier Lezoray. "Nonlocal PDEs-Based Morphology on Weighted Graphs for Image and Data Processing". In: *IEEE Transactions on Image Processing* 20.6 (2011), pp. 1504–1516. DOI: 10.1109/TIP.2010.2101610.

[36] Nicolás García Trillos, Franca Hoffmann, and Bamdad Hosseini. "Geometric structure of graph Laplacian embeddings". In: *Journal of Machine Learning Research* 22.63 (2021), pp. 1–55. URL: http://jmlr.org/papers/v22/19-683.html.

[37] Yves Van Gennip et al. "Community detection using spectral clustering on sparse geosocial data". In: *SIAM Journal on Applied Mathematics* 73.1 (2013), pp. 67–83.

[38] Xiaojin Zhu. "Semi-Supervised Learning with Graphs". PhD thesis. Carnegie Mellon University, 2005.

[39] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions". In: *IN ICML*. 2003, pp. 912–919.

2

Groundwork

Groundwork chapters don't get fancy quotes.

Jeremy Budd

In this chapter, we lay out the framework for analysis on graphs which we shall use for the remainder of this thesis, following the one presented by Van Gennip, Guillen, Osting, and Bertozzi in [8] *(which in turn was synthesising previous work, see the references within* [8]*). We will also briefly discuss the impact of the assumptions we impose on our graphs.*

Parts of this chapter have been published in *SIAM J. Math. Anal.* 52 (2020) [3], *Eur. J. Appl. Math.* (2021) [4], and *GAMM Mitteilungen* 44 (2021) [5].

2.1. Framework for analysis on graphs

The framework for analysis on graphs is presented in Van Gennip *et al.* [8], we reproduce here those aspects needed for our discussion. Let $G = (V, E, \omega)$ be a finite, undirected, weighted, and connected graph with no multi-edges or self-loops. Let G have vertex set V, edge set $E \subseteq V^2$ (with $ij \in E$ if and only if $ji \in E$ for all $i, j \in V$), 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 $ij \in E$. On G we define the spaces ($X \subseteq \mathbb{R}$):

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

Since V is finite, we shall interchangeably view elements of $\mathcal{V}$ and $\mathcal{V}_X$ as functions and as real vectors. Next, we define the spaces of time-dependent vertex functions (where $T \subseteq \mathbb{R}$ an interval)

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

For a parameter $r \in [0, 1]$, and denoting $d_i := \sum_j \omega_{ij}$, which we refer to as the *degree* of vertex i, we define the following inner products on $\mathcal{V}$ and $\mathcal{E}$:

$$\langle u, v\rangle_{\mathcal{V}} := \sum_{i\in V} u_i v_i d_i^r, \qquad \langle \varphi, \phi\rangle_{\mathcal{E}} := \frac{1}{2}\sum_{i,j\in V} \varphi_{ij}\phi_{ij}\omega_{ij} \tag{2.1}$$

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

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

where $(\cdot,\cdot)_{L^2(T;\mathbb{R})}$ is the standard continuum L^2 inner product. These inner products induce norms $||\cdot||_{\mathcal{V}}$, $||\cdot||_{\mathcal{E}}$, and $||\cdot||_{t\in T}$ in the usual way. We also define for $u \in \mathcal{V}$ the norm $||u||_\infty := \max_{i\in V} |u_i|$, and for $u \in \mathcal{V}_{t\in T}$ the norm $||u||_{\infty,t\in T} := \operatorname{ess\,sup}_{t\in T} ||u(t)||_\infty$ where ess sup denotes the essential supremum. We next define the L^2 and L^∞ spaces:

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

which we will consider as normed spaces with norms $||\cdot||_{t\in T}$ and $||\cdot||_{\infty,t\in T}$ respectively. Finally, 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 generalised time derivative $du/dt \in L^2(T;\mathcal{V})$ such that

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

where $C_c^\infty(T;\mathcal{V})$ denotes the infinitely differentiable and compactly supported elements of $\mathcal{V}_{t\in T}$. We link this to the familiar continuum H^1.

Proposition 2.1.1. *$u \in H^1(T;\mathcal{V})$ if and only if $u_i \in H^1(T;\mathbb{R})$ for each $i \in V$.*

Proof. Note that $(du/dt)_i = du_i/dt$, so u and $du/dt \in L^2(T;\mathcal{V})$ if and only if $\forall i \in V$, u_i and $du_i/dt \in L^2(T;\mathbb{R})$. Next, $(u, d\varphi/dt)_{t\in T} = -(du/dt, \varphi)_{t\in T}$ if and only if

$$\sum_{i\in V} d_i^r (u_i, d\varphi_i/dt)_{L^2(T;\mathbb{R})} = -\sum_{i\in V} d_i^r (du_i/dt, \varphi_i)_{L^2(T;\mathbb{R})}.$$

It follows that $\forall \varphi \in C_c^\infty(T;\mathcal{V})$ $(u, d\varphi/dt)_{t\in T} = -(du/dt, \varphi)_{t\in T}$ if and only if

$$\forall i \in V\ \forall \phi \in C_c^\infty(T;\mathbb{R})\ \ (u_i, d\phi/dt)_{L^2(T;\mathbb{R})} = -(du_i/dt, \phi)_{L^2(T;\mathbb{R})}$$

and therefore $\forall i \in V$ $u_i \in H^1(T;\mathbb{R})$. □

We define the following inner product on $H^1(T;\mathcal{V})$:

$$(u,v)_{H^1(T;\mathcal{V})} := (u,v)_{t\in T} + \left(\frac{du}{dt}, \frac{dv}{dt}\right)_{t\in T} = \sum_{i\in V} d_i^r (u_i, v_i)_{H^1(T;\mathbb{R})}.$$

We also define the local H^1 space

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

and we likewise define $L^2_{loc}(T;\mathcal{V})$. Furthermore, we define the Hölder spaces $C^{k,\alpha}(T;\mathcal{V})$ by, for $u \in \mathcal{V}_{t\in T}$, $u \in C^{k,\alpha}(T;\mathcal{V})$ if and only if for all $i \in V$, $u_i \in C^{k,\alpha}(T;\mathbb{R})$.

Next, we introduce the graph variants of the vector calculus gradient and Laplacian operators:

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

where the graph Laplacian Δ is positive semi-definite, unlike the negative semi-definite continuum Laplacian, and is self-adjoint with respect to $\mathcal{V}$. As shown in [8], 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 $D := \mathrm{diag}(d)$ (i.e. $D_{ii} := d_i$, and $D_{ij} := 0$ otherwise) to be the *degree matrix*. Then writing ω for the matrix of weights ω_{ij} we get

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

Our choice of r dictates which graph Laplacian we use. For $r = 0$ we have $\Delta = D - \omega$, which is the standard *unnormalised Laplacian* which we will sometimes denote Δ_u. For $r = 1$ we have $\Delta = I - D^{-1}\omega$, which is called the *random walk Laplacian*. Note that the *symmetric normalised Laplacian* $\Delta_s := I - D^{-1/2}\omega D^{-1/2}$ used in [2, 12] is not covered by our scheme. Finally, we note the following spectral properties of Δ (for more details see [8, Lemma 2.5]):

a. The smallest eigenvalue γ_0 of Δ is zero.

b. When G is connected, this eigenvalue has multiplicity one and has corresponding eigenvector $\xi_0 \propto \mathbf{1}$, the vector of ones.

c. Hence for any eigenvector ξ of Δ, either $\xi \propto \mathbf{1}$ with $\Delta\xi = \mathbf{0}$, or $\xi \perp \mathbf{1}$ and ξ has a strictly positive eigenvalue.

d. The spectral radius $\rho(\Delta)$ of Δ is bounded above by $2\max_{i\in V} d_i^{1-r}$.

Given the graph Laplacian, 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 the diffusion equation

$$\frac{dv}{dt} = -\Delta v,\ v(0) = u.$$

Note that $e^{-t\Delta}$ commutes with Δ for all $t \in \mathbb{R}$.

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

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

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

Proposition 2.1.2. *If $u \in H^1(T;\mathcal{V})$ and T bounded below, then $e^{-t\Delta}u \in H^1(T;\mathcal{V})$ with*

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

Proof. Let $T = (a,b)$ with $a > -\infty$. Now, $e^{-t\Delta}$ has eigenvalues $e^{-\lambda_k t}$, for $\lambda_k \geq 0$ the eigenvalues of Δ, and $e^{-t\Delta}$ is self-adjoint so $||e^{-t\Delta}|| = \rho(e^{-t\Delta}) \leq \max\{1, e^{-a||\Delta||}\}$ for $t \in T$. So $e^{-t\Delta}$ is a uniformly bounded operator for $t \in T$ and therefore $||e^{-t\Delta}u||_{t\in T} < \infty$ and (since $\Delta u, du/dt \in L^2(T;\mathcal{V})$) $||e^{-t\Delta}\frac{du}{dt} - e^{-t\Delta}\Delta u||_{t\in T} < \infty$. Then for $\varphi \in C_c^\infty(T;\mathcal{V})$

$$\begin{aligned}\left(e^{-t\Delta}\frac{du}{dt} - e^{-t\Delta}\Delta u, \varphi\right)_{t\in T} &= \left(\frac{du}{dt}, e^{-t\Delta}\varphi\right)_{t\in T} - \left(u, e^{-t\Delta}\Delta\varphi\right)_{t\in T}\\ &= -\left(u, \frac{d}{dt}\left(e^{-t\Delta}\varphi\right) + e^{-t\Delta}\Delta\varphi\right)_{t\in T}\\ &= -\left(u, e^{-t\Delta}\frac{d\varphi}{dt}\right)_{t\in T} = -\left(e^{-t\Delta}u, \frac{d\varphi}{dt}\right)_{t\in T}\end{aligned}$$

so $e^{-t\Delta}u$ has the desired generalised derivative. □

Finally, when considering variational problems of the form

$$\operatorname*{argmin}_{x \in X} f(x)$$

we will write $f \simeq g$ and say the functionals are *equivalent* when $g(x) = af(x) + b$ for a, b independent of x (or constant in x on X) and $a > 0$. This ensures that f and g have the same minimisers.

2.2. A note on our assumptions on the graph

As stated above, in this thesis we will assume that our graph G is finite, simple, connected, undirected, and positively weighted. In this subsection, we will briefly discuss the consequences of relaxing these conditions.

The case of G an infinite graph is a substantial divergence from our framework, affecting a large number of definitions and results. To detail these effects would take us well beyond the scope of this thesis; as an example, see e.g. [10] for the subtleties of defining Δ in the infinite case.

If G is non-simple, it must have multi-edges or self-loops. Multi-edges are essentially harmless for our framework, as they behave exactly like a single edge with weight equal to the sum of the weights of the multi-edges. If G has self-loops, then let G' be the simple subgraph of G without those self-loops. Then, as shown in [1], the unnormalised Laplacian Δ_u on G (defined as in [1, (1)]) and the unnormalised Laplacian Δ'_u on G' (defined as the $r = 0$ case of (2.1)) are related by

$$\Delta_u = \Delta'_u + M$$

where M is a diagonal matrix with diagonal entries $M_{ii} := \omega_{ii}$. Therefore, diffusion on G corresponds to the ODE

$$\frac{dv}{dt} = -\Delta_u v = -\Delta'_u v - Mv.$$

This can be observed to be a special case of fidelity forced diffusion on G', as will be defined in chapter 3. Finally, the degree matrices D and D' on G and G' are related by $D = D' + M$, so it follows that the normalised Laplacians $\Delta := D^{-r}\Delta_u$ and $\Delta' := D'^{-r}\Delta'_u$ are related by

$$\Delta = (I + MD'^{-1})^{-r}\Delta' + (D' + M)^{-r}M =: M_1\Delta' + M_2$$

where M_1 and M_2 are diagonal matrices, so diffusion with a normalised Laplacian on G corresponds to a forced and rescaled diffusion on G'.

If G is disconnected, it is a simple matter to apply our framework to each connected component of G.

If G is directed, then there are a number of different approaches to defining the Laplacian on a directed graph. For example, in [13, p. 6] and [14], the unnormalised Laplacian is defined by $\Delta_u = D - A$ where A is the (directed) adjacency matrix and D is the diagonal matrix of *out*-degrees. An alternative approach, found

in [15], is as follows: given a directed graph $G = (V, E)$, define vertex sets $\mathcal{H}, \mathcal{A} \subseteq V$ where $\mathcal{H}$ are the vertices with positive out-degrees d_i^{out} and $\mathcal{A}$ are the vertices with positive in-degrees d_i^{in} (note that $\mathcal{H} \cap \mathcal{A}$ need not be empty). Then, define the map $T : \mathcal{V}|_{\mathcal{A}} \to \mathcal{V}|_{\mathcal{H}}$ given by, for all $i \in \mathcal{H}$,

$$(Tu)_i = \sum_{j \in \mathcal{A}} \frac{\omega_{ij}}{\sqrt{d_i^{out} d_j^{in}}} u_j$$

with adjoint $T^* : \mathcal{V}|_{\mathcal{H}} \to \mathcal{V}|_{\mathcal{A}}$ given by, for all $j \in \mathcal{A}$,

$$(T^*u)_j = \sum_{i \in \mathcal{H}} \frac{\omega_{ij}}{\sqrt{d_i^{out} d_j^{in}}} u_i.$$

Next, extend T and T^* to $\mathcal{V}$ by setting $(Tu)_i = 0$ and $(T^*u)_j = 0$ for $i \notin \mathcal{H}$ and $j \notin \mathcal{A}$, and then for $\gamma \in [0, 1]$ define the Laplacian $\Delta_\gamma := I - \gamma T^*T - (1 - \gamma)TT^*$. A third approach can be found in [11, §2]. It is beyond the scope of this work to examine which of these definitions works best with our framework, and to what extent our framework can be extended to directed graphs.

Finally, if G is a signed graph (i.e., G has negative weights) then define $E^+ := \{ij \in E \mid \omega_{ij} > 0\}$ and $E^- := \{ij \in E \mid \omega_{ij} < 0\}$ and thus define the positively weighted graphs $G^+ := (V, E^+, \omega|_{E^+})$ and $G^- := (V, E^-, -\omega|_{E^-})$. It was shown in [6, (39)] that the unnormalised Laplacian Δ_u on G (defined as in [6, (36)]) can be decomposed as

$$\Delta_u = \Delta_u^+ + Q_u^-$$

where Δ_u^+ is the unnormalised Laplacian on G^+ and Q_u^- is the unnormalised *signless* Laplacian (see [7, 9] for details) on G^-, defined by

$$(Q_u^- v)_i := \sum_{j \in V \text{ s.t. } ij \in E^-} (-\omega_{ij})(v_i + v_j).$$

The authors of [6] then go on to define an AC flow and MBO scheme on G (see chapter 3), and apply this to a number of clustering problems. It is a topic for future research whether our framework can be extended to link AC flow and the MBO scheme on signed graphs.

Bibliography

[1] Behcet Acikmese. *Spectrum of Laplacians for Graphs with Self-Loops*. 2015. arXiv: 1505.08133 [math.OC].

[2] Andrea Bertozzi and Arjuna Flenner. "Diffuse Interface Models on Graphs for Classification of High Dimensional Data". In: *Multiscale Modeling Simulation* 10 (July 2012), pp. 1090–1118. DOI: 10.1137/11083109X.

[3] Jeremy Budd and Yves van Gennip. "Graph Merriman–Bence–Osher as a SemiDiscrete Implicit Euler Scheme for Graph Allen–Cahn Flow". In: *SIAM Journal on Mathematical Analysis* 52 (Jan. 2020), pp. 4101–4139. DOI: 10.1137/19M1277394.

[4] Jeremy Budd and Yves van Gennip. "Mass-conserving diffusion-based dynamics on graphs". In: *European Journal of Applied Mathematics* (Apr. 2021), pp. 1–49. DOI: 10.1017/S0956792521000061.

[5] Jeremy Budd, Yves van Gennip, and Jonas Latz. "Classification and image processing with a semi-discrete scheme for fidelity forced Allen–Cahn on graphs". English. In: *GAMM Mitteilungen* 44.1 (2021), pp. 1–43. ISSN: 0936-7195. DOI: 10.1002/gamm.202100004.

[6] Mihai Cucuringu, Andrea Pizzoferrato, and Yves van Gennip. In: *Communications in Mathematical Sciences* 19.1 (2021), pp. 73–109. DOI: 10.4310/CMS.2021.v19.n1.a4.

[7] Madhav Desai and Vasant Rao. "A characterization of the smallest eigenvalue of a graph". In: *Journal of Graph Theory* 18.2 (1994), pp. 181–194.

[8] Y. van Gennip et al. "Mean Curvature, Threshold Dynamics, and Phase Field Theory on Finite Graphs". In: *Milan Journal of Mathematics* 82 (2014), pp. 3–65.

[9] Willem H Haemers and Edward Spence. "Enumeration of cospectral graphs". In: *European Journal of Combinatorics* 25.2 (2004), pp. 199–211.

[10] Sebastian Haeseler et al. "Laplacians on infinite graphs: Dirichlet and Neumann boundary conditions". In: *Journal of Spectral Theory* 2 (Mar. 2011), pp. 397–432. DOI: 10.4171/JST/35.

[11] M. Hein, J. Audibert, and U. von Luxburg. "From Graphs to Manifolds - Weak and Strong Pointwise Consistency of Graph Laplacians". In: Student Paper Award. Max Planck Society. 2005, pp. 470–485.

[12] Ekaterina Merkurjev, Tijana Kostić, and Andrea Bertozzi. "An MBO Scheme on Graphs for Classification and Image Processing". In: *SIAM Journal on Imaging Sciences* 6 (Oct. 2013), pp. 1903–1910. DOI: 10.1137/120886935.

[13] Bojan Mohar. "The Laplacian spectrum of graphs". In: *Graph Theory, Combinatorics, and Applications*. Wiley, 1991, pp. 871–898.

[14] Ying Zhang, Zhiqiang Zhao, and Zhuo Feng. *A Unified Approach to Scalable Spectral Sparsification of Directed Graphs*. 2020. arXiv: 1812.04165 [cs.DS].

[15] Dengyong Zhou, Thomas Hofmann, and Bernhard Schölkopf. "Semi-supervised Learning on Directed Graphs". In: *Advances in Neural Information Processing Systems*. Ed. by L. Saul, Y. Weiss, and L. Bottou. Vol. 17. MIT Press, 2005. URL: https://proceedings.neurips.cc/paper/2004/file/ad47a008a2f806aa6eb1b53852cd8b37-Paper.pdf.

3

Graph Allen–Cahn flow and the graph MBO scheme

I know what [an analogy] is.
It's like a thought with another thought's hat on.

"Urban Matrimony and the Sandwich Arts", *Community*

In the continuum setting, Allen–Cahn (AC) flow and the Merriman–Bence–Osher (MBO) scheme are known to be linked via their mutual connection to mean curvature flow (MCF) (see chapter 7 for a discussion of graph MCF) and this link is known to be robust to the inclusion of mass conservation and fidelity forcing constraints (see section 3.1 for details). In this chapter, we develop important theory for the graph AC flow. First, we define the graph AC flow and MBO scheme, and define mass-conserving and fidelity forced variants. Next, we define our semi-discrete implicit Euler (SDIE) scheme *for AC flow, which will be the key ingredient of chapter 4's proof of a rigorous link between graph AC flow and the MBO scheme. We observe that the variational forms of the MBO and SDIE schemes suggest using the* double-obstacle potential *as the potential in AC flow. The bulk of this chapter then examines the properties of this double-obstacle AC flow, including its two constrained variants. We shall exhibit weak forms and explicit integral forms, prove conditions under which the flow "freezes", and prove uniqueness of solutions. Furthermore, we shall state the existence and Lipschitz regularity of solutions, monotonic decrease of the Ginzburg–Landau energy along solutions, and in the ordinary and fidelity forced cases, well-posedness of the*

Parts of this chapter have been published in *SIAM J. Math. Anal.* 52 (2020) [12], *Eur. J. Appl. Math.* (2021) [13], and *GAMM Mitteilungen* 44 (2021) [14].

ODE. These properties will all be proved in chapter 4, using the properties of the SDIE scheme.

3.1. Background

In the continuum, the links between AC flow and the MBO scheme have been well-studied. The key result is that both AC flow and the MBO scheme converge to mean curvature flow as their respective parameters tend to zero (see e.g. Bronsard and Kohn [11] for details on the convergence of AC flow to mean curvature flow, and Evans [21] for details on the convergence of the MBO scheme). Furthermore, this link is robust to the addition of additional constraints, in particular a mass conservation (a.k.a. volume preservation) constraint. A mass-constrained MBO scheme was first introduced in Ruuth and Wetton [31], and recently Laux and Swartz [25] showed that this scheme converges (up to a subsequence) to the weak formulation of mass-constrained mean curvature flow defined in Mugnai *et al.* [28]. Mass-conserving dynamics of the Ginzburg–Landau functional date back to [15, 16] and the development of the Cahn–Hilliard equation. In the 1990s, Rubinstein and Sternberg [30] devised a mass-conserving variant of the AC flow as an alternative to the Cahn–Hilliard equation, which more recently Chen, Hilhorst, and Logak [18] have rigorously proved has mass-conserving mean curvature flow as its phase field limit. We will use Rubinstein and Sternberg's equation as the basis for our mass-conserving graph AC flow.

Turning to the graph context, graph AC flow and MBO schemes (with fidelity forcing) have received much attention in the last decade as algorithms for image processing and semi-supervised learning, stemming from pioneering work by Bertozzi and Flenner [5] using graph AC flow, and Merkurjev, Kostić, and Bertozzi [27] using a graph MBO scheme. Following this, Bae and Merkurjev [2] studied the effect of mass conservation constraints on these algorithms. The use of these methods was based on an implicit assumption that the continuum connections between these processes extend to their graph counterparts. In chapter 4 we will demonstrate rigorously that these graph flows are indeed linked, though our route will not go via mean curvature flow.

On the theoretical side, Van Gennip, Guillen, Osting, and Bertozzi [22] defined a framework for analysis on graphs, and defined a graph AC flow and MBO scheme (and mean curvature flow, which we will consider in chapter 7) within that framework. They then proved rigorous results about these flows individually, particularly about the conditions under which these flows "pin" or "freeze". More recently Van Gennip [23] studied a graph analogue of the Ohta–Kawasaki functional for pattern formation, and devised a mass-conserving modified graph MBO scheme as a method for minimising this functional with a mass conservation constraint.

3.2. Definitions of AC flow and the MBO scheme

We shall begin by defining our two key processes, in three settings. First we shall define them in their ordinary form. Then, we shall introduce two different types of extra dynamics: mass-conservation, and fidelity forcing.

3.2.1. The ordinary case

Definition 3.2.1 (Graph MBO scheme)**.** *The* graph Merriman–Bence–Osher (MBO) scheme *is a scheme that creates a series of* $u_n \in \mathcal{V}_{\{0,1\}}$ *by the following two-step iteration: for* $u_n \in \mathcal{V}_{\{0,1\}}$*, and* $\tau > 0$ *the* time step*, define* u_{n+1} *via*

1. $v_n := e^{-\tau\Delta} u_n$*, i.e. 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* $v \in \mathcal{V}$*,*

$$(\Theta(v))_i := \begin{cases} 1, & \text{if } v_i \geq 1/2, \\ 0, & \text{if } v_i < 1/2. \end{cases} \tag{3.1}$$

In [22, Proposition 4.6] *it was shown that this scheme can be expressed variationally, where* $u_n = \chi_{S_n}$*, by*

$$u_{n+1} \in \operatorname*{argmin}_{u \in \mathcal{V}_{[0,1]}} \left\langle \mathbf{1} - 2e^{-\tau\Delta} u_n, u \right\rangle_{\mathcal{V}} \tag{3.2}$$

which we can rewrite with the equivalent functional:[1]

$$u_{n+1} \in \operatorname*{argmin}_{u \in \mathcal{V}_{[0,1]}} \frac{1}{2\tau} \langle \mathbf{1} - u, u \rangle_{\mathcal{V}} + \frac{\left|\left| u - e^{-\tau\Delta} u_n \right|\right|_{\mathcal{V}}^2}{2\tau}. \tag{3.3}$$

Note 2. *Note that* (3.3) *has a form resembling a* discrete solution [1, Definition 2.0.2] *(cf. the study of minimising movements) of a gradient flow. That is, it resembles a sequence arising from an Euler scheme for a gradient flow. This motivates our link to AC flow.*

Definition 3.2.2 (Graph Allen–Cahn flow)**.** Graph Allen–Cahn (AC) flow *is the* $\langle \cdot, \cdot \rangle_{\mathcal{V}}$ *gradient flow of the* graph Ginzburg–Landau functional*, which we shall define as*:

$$\mathrm{GL}_\varepsilon(u) := \frac{1}{2} ||\nabla u||_\varepsilon^2 + \frac{1}{\varepsilon} \langle W \circ u, \mathbf{1} \rangle_{\mathcal{V}} \tag{3.4}$$

where W *is a double-well potential with wells at 0 and 1 (i.e.* $W : \mathbb{R} \to [0, \infty]$ *has* $W(0) = W(1) = 0$ *and is strictly positive at all other values) and* $\varepsilon > 0$ *is a parameter. This definition slightly differs from that in Van Gennip* et al. [22]*: we have replaced their* $\sum_{i \in V} W(u_i)$ *with* $\langle W \circ u, \mathbf{1} \rangle_{\mathcal{V}}$*, which we have found plays better with the Hilbert space structure and enables the link we derive with the MBO scheme. The AC flow is then given, for* $W : \mathbb{R} \to [0, \infty)$ *differentiable, by the ODE*:

$$\frac{du}{dt} = -\Delta u - \frac{1}{\varepsilon} W' \circ u = -\nabla_{\mathcal{V}} \mathrm{GL}_\varepsilon(u) \tag{3.5}$$

where $\nabla_{\mathcal{V}}$ *is the Hilbert space gradient on* $\mathcal{V}$*.*

[1] One can check that $\left\langle \mathbf{1} - 2e^{-\tau\Delta} u_n, u \right\rangle_{\mathcal{V}} = \langle u, \mathbf{1} - u \rangle_{\mathcal{V}} + \langle u - e^{-\tau\Delta} u_n, u - e^{-\tau\Delta} u_n \rangle_{\mathcal{V}} - \langle e^{-\tau\Delta} u_n, e^{-\tau\Delta} u_n \rangle_{\mathcal{V}}$. Then suppress the constant (in u) term $\langle e^{-\tau\Delta} u_n, e^{-\tau\Delta} u_n \rangle_{\mathcal{V}}$ and divide by 2τ.

3.2.2. The fidelity forced case

Following [20, 27], we first define fidelity forced diffusion.

Definition 3.2.3 (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:*

$$\frac{du}{dt}(t) = -\Delta u(t) - M(u(t) - \tilde{f}) =: -Au(t) + M\tilde{f}, \qquad u(0) = u_0, \tag{3.6}$$

where $M := \mathrm{diag}(\mu)$ for $\mu \in \mathcal{V}_{[0,\infty)}$ the fidelity parameter*, $A := \Delta + M$, and $\tilde{f} \in \mathcal{V}_{[0,1]}$ is the* reference*. We define $Z := \mathrm{supp}(\mu)$, which we call the* reference data*. The effect of these added terms is that they enforce fidelity to the reference on the reference data. Note that μ_i paramaterises the strength of the fidelity to the reference at vertex i. 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 3. *That μ can be non-constant on Z has practical relevance, for example if the confidence in the accuracy of the reference were higher at some vertices of the reference data versus others. This is due to the link between the value of the fidelity parameter at a vertex and the statistical precision (i.e. the inverse of the variance of the noise) of the reference at that vertex (see* [4, § 3.3] *for details).*

Proposition 3.2.4. *If $\mu \neq \mathbf{0}$, then 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 non-negative. 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 $||A|| = ||\Delta + M|| \leq ||\Delta|| + ||M|| = ||\Delta|| + ||\mu||_\infty$. □

We define a useful map.

Definition 3.2.5. *For $t, x \in \mathbb{R}$, let $F_t(x) := (1 - e^{-tx})/x$. Then F_t has Taylor series*

$$F_t(x) = \sum_{n=0}^{\infty} (-1)^n \frac{t^{n+1}}{(n+1)!} x^n.$$

We extend F_t to (real) matrix input via

$$F_t(X) := \sum_{n=0}^{\infty} (-1)^n \frac{t^{n+1}}{(n+1)!} X^n.$$

Note that

$$XF_t(X) = F_t(X)X = -\sum_{n=1}^{\infty} (-1)^n \frac{t^n}{n!} X^n = I - e^{-tX}$$

so if X is invertible then $F_t(X) = X^{-1}(I - e^{-tX}) = (I - e^{-tX})X^{-1}$.

Theorem 3.2.6. *For given $u_0 \in \mathcal{V}$, (3.6) has a unique solution in $H^1_{loc}([0,\infty); \mathcal{V})$. The solution u to (3.6) is $C^1((0,\infty); \mathcal{V})$ and is given by the map:*

$$u(t) = \mathcal{S}_t u_0 := e^{-tA} u_0 + F_t(A) f. \tag{3.7}$$

This solution map has the following properties:

i. If $u_0 \leq v_0$ vertexwise, then for all $t \geq 0$, $\mathcal{S}_t u_0 \leq \mathcal{S}_t v_0$ vertexwise.

ii. $\mathcal{S}_t : \mathcal{V}_{[0,1]} \to \mathcal{V}_{[0,1]}$ for all $t \geq 0$, i.e. if $u_0 \in \mathcal{V}_{[0,1]}$ then $u(t) \in \mathcal{V}_{[0,1]}$.

Proof. It is straightforward to check directly that (3.7) satisfies (3.6) and is C^1 on $(0,\infty)$. Uniqueness is given by a standard Picard–Lindelöf argument (see e.g. [32, Corollary 2.6]).

i. By definition, $\mathcal{S}_t v_0 - \mathcal{S}_t u_0 = e^{-tA}(v_0 - u_0)$. Thus it suffices to show that e^{-tA} is a non-negative matrix (i.e., a matrix with all entries non-negative) for $t \geq 0$. Note that the off-diagonal elements of $-tA$ are non-negative: 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 non-negative matrix and thus e^Q is a non-negative matrix. It follows that $e^{-tA} = e^{-a}e^Q$ is a non-negative matrix.

ii. Let $u_0 \in \mathcal{V}_{[0,1]}$. Then $\mathbf{0} \leq u_0 \leq \mathbf{1}$ vertexwise and thus by (i) $\mathcal{S}_t \mathbf{0} \leq \mathcal{S}_t u_0 \leq \mathcal{S}_t \mathbf{1}$, so it suffices to show that $\mathcal{S}_t \mathbf{0} \geq \mathbf{0}$ and $\mathcal{S}_t \mathbf{1} \leq \mathbf{1}$. If $\mu = \mathbf{0}$, then $\mathcal{S}_t \mathbf{0} = \mathbf{0}$ and $\mathcal{S}_t \mathbf{1} = \mathbf{1}$. If $\mu \neq \mathbf{0}$, let $v(t) := \mathcal{S}_t \mathbf{0} = A^{-1} f - A^{-1} e^{-tA} f$. Then $v(0) = \mathbf{0}$ and

$$\frac{dv}{dt}(t) = e^{-tA} f \geq \mathbf{0}$$

since $f \geq \mathbf{0}$. Hence $v(t) \geq \mathbf{0}$ as desired. Finally, let

$$w(t) := \mathbf{1} - \mathcal{S}_t \mathbf{1} = \mathbf{1} - A^{-1} f - A^{-1} e^{-tA}(A\mathbf{1} - f) = \mathbf{1} - A^{-1} f - A^{-1} e^{-tA} M(\mathbf{1} - \tilde{f}).$$

Then $w(0) = \mathbf{0}$ and

$$\frac{dw}{dt}(t) = e^{-tA} M(\mathbf{1} - \tilde{f}) \geq \mathbf{0}$$

since $\tilde{f} \in \mathcal{V}_{[0,1]}$ and M is a non-negative matrix, so $w(t) \geq \mathbf{0}$ as desired.

□

We can then define a fidelity forced MBO scheme and AC flow.

Definition 3.2.7 (Graph MBO with fidelity forcing)**.** *For $u_0 \in \mathcal{V}_{[0,1]}$ we follow* [20, 27]*, and define the sequence of MBO iterates by diffusing with fidelity for a time $\tau \geq 0$ and then thresholding, i.e. $u_{n+1} = \Theta(\mathcal{S}_\tau(u_n))$ where $\mathcal{S}_\tau$ is the solution map from* (3.7)*. By the same argument as in the ordinary case, this has variational form*:

$$u_{n+1} \in \underset{u \in \mathcal{V}_{[0,1]}}{\operatorname{argmin}} \langle \mathbf{1} - 2\mathcal{S}_\tau u_n, u \rangle_{\mathcal{V}} \simeq \frac{1}{2\tau} \langle \mathbf{1} - u, u \rangle_{\mathcal{V}} + \frac{||u - \mathcal{S}_\tau u_n||_{\mathcal{V}}^2}{2\tau}. \tag{3.8}$$

Definition 3.2.8 (Graph AC flow with fidelity forcing)**.** *First, define the* graph Ginzburg–Landau functional with fidelity *by*:

$$\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u) := \frac{1}{2} ||\nabla u||_{\mathcal{E}}^2 + \frac{1}{\varepsilon} \langle W \circ u, \mathbf{1} \rangle_{\mathcal{V}} + \frac{1}{2} \langle u - \tilde{f}, M(u - \tilde{f}) \rangle_{\mathcal{V}}. \tag{3.9}$$

The gradient flow of this with respect to $\langle \cdot, \cdot \rangle_{\mathcal{V}}$ is the fidelity forced AC flow:

$$\frac{du}{dt} = -\Delta u - \frac{1}{\varepsilon} W' \circ u - M(u - \tilde{f}) = -Au - \frac{1}{\varepsilon} W' \circ u + f. \tag{3.10}$$

Note 4. *If $\mu = \mathbf{0}$, then we observe that $A = \Delta$, $f = \mathbf{0}$, and thus $\mathcal{S}_t u = e^{-t\Delta} u$. Therefore, the fidelity forced MBO scheme and AC flow reduce to the ordinary MBO scheme and AC flow when $\mu = \mathbf{0}$, so the ordinary case is a special case of the fidelity forced case.*

3.2.3. The mass-conserving case

We first define what we mean by "mass" in this setting.

Definition 3.2.9. *Define the* mass *of $u \in \mathcal{V}$ to be*

$$\mathcal{M}(u) := \langle u, \mathbf{1} \rangle_{\mathcal{V}}. \tag{3.11}$$

Furthermore, define the average value *of $u \in \mathcal{V}$ to be*

$$\bar{u} := \frac{\mathcal{M}(u)}{\mathcal{M}(\mathbf{1})}. \tag{3.12}$$

Given this definition, we can use the variational form to define a mass-conserving MBO scheme.

Definition 3.2.10 (Mass-conserving graph MBO scheme)**.** *We define the* mass-conserving graph MBO scheme *by the sequence of variational problems*:

$$u_{n+1} \in \underset{\substack{u \in \mathcal{V}_{[0,1]} \\ \mathcal{M}(u) = \mathcal{M}(u_n)}}{\operatorname{argmin}} \left\langle \mathbf{1} - 2e^{-\tau\Delta} u_n, u \right\rangle_{\mathcal{V}} \simeq -\left\langle e^{-\tau\Delta} u_n, u \right\rangle_{\mathcal{V}}.$$

To define a mass-conserving AC flow, we follow the definition of Rubinstein and Sternberg, who in [30] define a mass-conserving Allen–Cahn flow (on a domain Ω) as the non-local reaction-diffusion PDE, where $u : \Omega \to \mathbb{R}$,

$$\frac{\partial u}{\partial t} = \Delta u - W'(u) + \frac{1}{|\Omega|}\int_\Omega W'(u)\, dx \tag{3.13}$$

with Neumann boundary conditions. We can readily formulate this on a graph, noting the differing sign convention on Δ and introducing our scaling, as follows.

Definition 3.2.11 (Mass-conserving graph AC flow)**.** Mass-conserving graph AC flow *is given by the ODE:*

$$\frac{du}{dt} = -\Delta u - \frac{1}{\varepsilon}W' \circ u + \frac{1}{\varepsilon}\frac{\langle W' \circ u, \mathbf{1}\rangle_{\mathcal{V}}}{\langle \mathbf{1}, \mathbf{1}\rangle_{\mathcal{V}}}\mathbf{1}. \tag{3.14}$$

We verify the mass conservation property for u continuous and H^1. We first recall a standard fact about continuous representatives of H^1 functions.

Lemma 3.2.12 (See [12, Lemma 3.1])**.** *For any interval T, if $u \in H^1_{loc}(T;\mathcal{V}) \cap C^0(T;\mathcal{V})$ or $u \in H^1_{loc}(T;\mathbb{R}) \cap C^0(T;\mathbb{R})$, then u is locally absolutely continous on T. It follows that u is differentiable a.e. in T, and the weak derivative equals the classical derivative a.e. in T.*

Proof. By Proposition 2.1.1, $u \in H^1_{loc}(T;\mathcal{V}) \cap C^0(T;\mathcal{V})$ if and only if for all $i \in V$, $u_i \in H^1_{loc}(T;\mathbb{R}) \cap C^0(T;\mathbb{R})$. The result then follows from standard results, see [26, Theorem 7.13]. □

Proposition 3.2.13. *For any interval T and $u \in H^1_{loc}(T;\mathcal{V}) \cap C^0(T;\mathcal{V})$, if u obeys (3.14) at a.e. $t \in T$, then for a.e. $t \in T$*

$$\frac{d}{dt}\mathcal{M}(u(t)) = 0$$

and so $\mathcal{M}(u(t))$ is constant.

Proof. First, note that $\mathcal{M}(u(t)) \in H^1_{loc}(T;\mathbb{R}) \cap C^0(T;\mathbb{R})$ with

$$\frac{d}{dt}\mathcal{M}(u(t)) = \left\langle \frac{du}{dt}, \mathbf{1}\right\rangle_{\mathcal{V}}$$

since for any $\varphi \in C^\infty_c(T;\mathbb{R})$

$$\begin{aligned}\int_T \langle u(t), \mathbf{1}\rangle_{\mathcal{V}} \frac{d\varphi}{dt}\, dt &= \int_T \left\langle u(t), \frac{d\varphi}{dt}\mathbf{1}\right\rangle_{\mathcal{V}} dt \\ &= -\int_T \left\langle \frac{du}{dt}, \varphi(t)\mathbf{1}\right\rangle_{\mathcal{V}} dt = -\int_T \left\langle \frac{du}{dt}, \mathbf{1}\right\rangle_{\mathcal{V}} \varphi(t)\, dt.\end{aligned}$$

Then for almost every $t \in T$, taking the mass of both sides of (3.14):

$$\left\langle \frac{du}{dt}, \mathbf{1} \right\rangle_{\mathcal{V}} = -\langle \Delta u(t), \mathbf{1} \rangle_{\mathcal{V}} - \frac{1}{\varepsilon} \langle W' \circ u, \mathbf{1} \rangle_{\mathcal{V}} + \frac{1}{\varepsilon} \frac{\langle W' \circ u, \mathbf{1} \rangle_{\mathcal{V}}}{\langle \mathbf{1}, \mathbf{1} \rangle_{\mathcal{V}}} \langle \mathbf{1}, \mathbf{1} \rangle_{\mathcal{V}}$$

So most of the terms cancel and we are left with

$$\left\langle \frac{du}{dt}, \mathbf{1} \right\rangle_{\mathcal{V}} = -\langle \Delta u(t), \mathbf{1} \rangle_{\mathcal{V}} = 0$$

with the final equality because Δ is self-adjoint and $\Delta \mathbf{1} = \mathbf{0}$. Then by absolute continuity we infer that $\mathcal{M}(u(t))$ is constant. □

3.3. Set-up for the next chapter

The key result of the next chapter will be a rigorous link between the AC flow and MBO schemes. This link will be demonstrated by showing that the MBO scheme is a special case of what we call a semi-discrete implicit Euler (SDIE) scheme for AC flow, for a particular choice of W. We will here define that SDIE scheme in the ordinary, fidelity forced, and mass-conserving cases, as well as describe (and motivate) our choice of potential.

3.3.1. Definitions of the SDIE scheme

Definition 3.3.1 (SDIE scheme in the ordinary, fidelity forced, and mass-conserving cases)**.** *Let $\tau \geq 0$ be the time step for the scheme. Then for the AC flow (3.5), the SDIE scheme is defined by*

$$u_{n+1} = e^{-\tau\Delta} u_n - \frac{\tau}{\varepsilon} W' \circ u_{n+1}. \tag{3.15}$$

For the fidelity forced AC flow (3.10), the fidelity forced SDIE scheme is defined by

$$u_{n+1} = \mathcal{S}_\tau u_n - \frac{\tau}{\varepsilon} W' \circ u_{n+1}. \tag{3.16}$$

Finally, for the mass-conserving AC flow (3.14), the mass-conserving SDIE scheme is defined by

$$u_{n+1} = e^{-\tau\Delta} u_n - \frac{\tau}{\varepsilon} W' \circ u_{n+1} + \frac{\tau}{\varepsilon} \overline{W' \circ u_{n+1}} \mathbf{1}. \tag{3.17}$$

These schemes can be rewritten in variational form. For now, we shall show this just for (3.15), as that will suffice to motivate what follows.

Theorem 3.3.2. *Let $\lambda := \tau/\varepsilon$, and assume that $W \in C^2(\mathbb{R})$ and that for all $x \in \mathbb{R}$, $W''(x) \geq -1/\lambda$. Then u_{n+1} solves (3.15) if and only if*

$$u_{n+1} \in \operatorname*{argmin}_{u \in \mathcal{V}} \lambda \langle W \circ u, \mathbf{1} \rangle_{\mathcal{V}} + \frac{1}{2} \left\| u - e^{-\tau\Delta} u_n \right\|_{\mathcal{V}}^2. \tag{3.18}$$

Proof. The objective function in (3.18) is a sum of independent terms which only depend on the value of u at a single vertex, hence u_{n+1} solves (3.18) if and only if for all $i \in V$

$$(u_{n+1})_i \in \operatorname*{argmin}_{x\in\mathbb{R}} \lambda W(x) + \frac{1}{2}\left(x - (e^{-\tau\Delta}u_n)_i\right)^2 =: g_i(x).$$

Suppose that u_{n+1} solves (3.18). Then for all $i \in V$ we have $g_i'((u_{n+1})_i) = 0$, i.e.

$$(u_{n+1})_i - (e^{-\tau\Delta}u_n)_i + \lambda W'((u_{n+1})_i) = 0$$

and so u_{n+1} obeys (3.15).

Next, suppose that u_{n+1} solves (3.15). Let $i \in V$, $y := (u_{n+1})_i$, and $z := (e^{-\tau\Delta}u_n)_i$. Then $y = z - \lambda W'(y)$ and we seek to prove that y is a global minimiser of

$$h(x) := \lambda W(x) + \frac{1}{2}(x-z)^2$$

for $x \in \mathbb{R}$. Note that $h'(y) = \lambda W'(y) + y - z = 0$, and that for all $x \in \mathbb{R}$, $h''(x) = \lambda W''(x) + 1 \geq 0$ by the assumption on W. Finally, by Taylor's theorem, for all $x \in \mathbb{R}$ there exists $\xi \in \mathbb{R}$ such that

$$h(x) = h(y) + (x-y)h'(y) + \frac{1}{2}(x-y)^2 h''(\xi) \geq h(y),$$

where the inequality follows from the above notes on h, and thus y is a global minimiser of h. It follows that u_{n+1} solves (3.18). □

Note 5. *This result does not rely on the double-well or non-negativity properties of W.*

3.3.2. Connection to time-splitting for AC flow

The name "semi-discrete" refers to the fact that the scheme uses the exact solution operator for the diffusion part of the AC ODE, and uses an implicit Euler time-discretisation for the potential term. We can make this motivation more precise by interpreting equation (3.15) as an Euler scheme for a time-splitting scheme for AC flow, as follows. As with the previous theorem, for simplicity we shall show this just for the ordinary case (3.15).

We fix $\tau > 0$ and take $\tilde{u}_0 \in \mathcal{V}$, then iteratively apply the steps:

1. **(Diffusion step)** Define $v := e^{-t\Delta}\tilde{u}_n$ the heat equation solution with $v(0) = \tilde{u}_n$ and define $v_n := v(\tau)$.
2. **(Reaction step)** Define $U_n \in H^1((0,\tau);\mathcal{V}) \cap C^0([0,\tau];\mathcal{V})$ obeying

$$\frac{dU_n}{dt} = -\varepsilon^{-1}W' \circ U_n, \quad U_n(0) = v_n = e^{-\tau\Delta}\tilde{u}_n. \tag{3.19}$$

3. Finally, define $\tilde{u}_{n+1} := U_n(\tau)$.

The relation of this time-splitting to the semi-discrete scheme is that if $u_n = \tilde{u}_n$ we can recognise from (3.15) the semi-discrete update $u_{n+1} \approx \tilde{u}_{n+1}$ as the implicit Euler approximation of (3.19):

$$\frac{u_{n+1} - v_n}{\tau} = -\frac{1}{\varepsilon} W' \circ u_{n+1}.$$

That is, we get the semi-discrete update by dissecting the flow in (3.5) into a diffusion for time τ followed by a gradient flow of W, again for time τ. Then, we use the exact solution for the former and approximate the latter by an implicit Euler scheme. For more detail on the connection between these schemes, see [12, §4.3], we omit these details here as they are not particularly relevant to the rest of this thesis.

3.3.3. The double-obstacle potential

If we compare the variational form of the MBO scheme (3.3) and the SDIE scheme (3.18), we observe a striking similarity between the objective functions. In order to make this similarity exact, we need three things to be true:

i. W needs to equal $\frac{1}{2}x(1-x)$ on $[0,1]$.

ii. W needs to force the minimisers to lie in $\mathcal{V}_{[0,1]}$.

iii. τ needs to equal ε, i.e. $\lambda = 1$.

Conditions (i) and (ii) force us to choose as our potential W the *double-obstacle potential*:

$$W(x) := \begin{cases} \frac{1}{2}x(1-x), & \text{for } 0 \leq x \leq 1, \\ \infty, & \text{otherwise.} \end{cases} \tag{3.20}$$

See Oono and Puri [29] and Blowey and Elliott [6, 7, 8] for study of this potential in the continuum context and Bosch, Klamt, and Stoll [9] for recent work in the graph context. Henceforth, W will exclusively refer to this potential.

This choice of potential is essential to the whole of this work, so to motivate this choice we review some of its virtues. One of the key advantages of this potential over smooth alternatives (as previously noted by Chen and Elliot in [17, p. 430], where a bunch of other more continuum-centric virtues are also discussed) is that it forces solutions to lie in $\mathcal{V}_{[0,1]}$ come-what-may, whatever extra constraints or dynamics are imposed. This property is especially important when trying to link up with the MBO hard thresholding. The quadratic form of the potential between the wells is also very convenient for a number of reasons. Firstly, it meets condition (i). Secondly, it means that W' is an affine function on $(0,1)$, leading to the resulting AC flow being analysable using the tools for linear ODEs/differential inclusions. Finally, in the variational forms the negative quadratic part of $W \circ u$ is cancelled out by the quadratic term (e.g., $||u - e^{-\tau\Delta}u_n||_V^2$ in (3.18)) to give a convex objective function, which will allow us to employ the tools of convex optimisation to study these variational problems.

However, this choice does have one major drawback, which is that it is not differentiable at 0 or 1. Therefore, we shall have to replace the references to W' in the above definitions of AC flow and the SDIE scheme with more careful considerations of the subdifferential, and we shall rigorously prove that the various definitions and different forms continue to make sense. Towards this, we must define some further notation. Write

$$W(x) = \frac{1}{2}x(1-x) + I_{[0,1]}(x)$$

where $I_{[0,1]}$ is the indicator function taking value 0 on $[0,1]$ and ∞ elsewhere. Now, we will rewrite our gradient flows against W using the subdifferential. That is, we will rewrite a differential equation of the form (for $\mathcal{D}$ some differential operator)

$$\mathcal{D}u + \frac{1}{\varepsilon}W'(u) = 0$$

as the differential inclusion

$$\mathcal{D}u \in -\frac{1}{\varepsilon}\partial W(u)$$

where $\partial W(u)$ denotes the subdifferential of W at u (see Ekeland and Temam [19, Definition 5.1]). Using the above decomposition of W, this becomes: for a.e. t there exists $\beta(t)$ such that for all $i \in V$, $\beta_i(t) \in -\partial I_{[0,1]}(u_i(t))$ and

$$\varepsilon\mathcal{D}u(t) + \frac{1}{2}\mathbf{1} - u(t) = \beta(t).$$

The condition on $\beta(t)$ can be written more transparently as[2]

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

Notice that this expression only makes sense for trajectories such that $u(t) \in \mathcal{V}_{[0,1]}$ at a.e. t. For tidiness of notation, we define

$$\mathcal{B}(u) := \left\{\alpha \in \mathcal{V} \,\middle|\, \forall i \in V, \alpha_i \in -\partial I_{[0,1]}(u_i)\right\} \tag{3.21}$$

which is non-empty if and only if $u \in \mathcal{V}_{[0,1]}$. Then $\beta(t)$ satisfies the above condition if and only if $\beta(t) \in \mathcal{B}(u(t))$.

[2]This differs slightly from the corresponding expression given in [12, p. 4108]. In that paper $\partial I_{[0,1]}(x)$ was defined such that $\partial I_{[0,1]}(x) = \{-\infty\}$ for $x < 0$ and $\partial I_{[0,1]}(x) = \{\infty\}$ for $x > 1$, following [6, (1.14b)], whilst here we have defined $\partial I_{[0,1]}(x)$ to be empty in both of those cases. This difference of convention has no effect on the results, as in the former case $\beta_i(t)$ was restricted to $\mathbb{R}$.

3.4. Double-obstacle AC flow

Due to the non-differentiability of the double-obstacle potential, we first need to redefine our AC flows along the lines of section 3.3.3. Since the ordinary case is just a special case of the fidelity forced case, we will henceforth only consider two cases: fidelity forced, and mass-conserving. We will then investigate the properties of these flows.

3.4.1. Redefining the AC flow

Let us begin with the fidelity forced case. Rewriting the definition via the subdifferential, we get the following definition.

Definition 3.4.1 (Double-obstacle AC flow 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 AC flow with fidelity forcing on T when $u \in H^1_{loc}(T;\mathcal{V}) \cap C^0(T;\mathcal{V})$ and for almost every $t \in T$,*

$$\varepsilon \frac{du}{dt}(t) + \varepsilon A u(t) - \varepsilon f + \frac{1}{2}\mathbf{1} - u(t) = \beta(t), \qquad \beta(t) \in \mathcal{B}(u(t)). \tag{3.22}$$

Likewise, we redefine in the mass-conserving case as follows.

Definition 3.4.2 (Mass-conserving double-obstacle AC flow)**.** *Let T be any interval. A pair $(u,\beta) \in \mathcal{V}_{[0,1],t\in T} \times \mathcal{V}_{t\in T}$ is a solution to mass-conserving double-obstacle AC flow on T when $u \in H^1_{loc}(T;\mathcal{V}) \cap C^0(T;\mathcal{V})$ and for almost every $t \in T$*

$$\varepsilon \frac{du}{dt} + \varepsilon \Delta u(t) - u(t) + \overline{u(t)}\mathbf{1} = \beta(t) - \overline{\beta(t)}\mathbf{1}, \qquad \beta(t) \in \mathcal{B}(u(t)). \tag{3.23}$$

Proposition 3.4.3. *Let (u,β) be as in Definition 3.4.1 on an interval T. Then $\overline{u(t)}$ is constant on T.*

Proof. Follows as in Proposition 3.2.13 *mutatis mutandis*. □

Notice that what we have done here is redefine our ODEs to be differential inclusions. However, by the use of Lemma 3.2.12, we can in fact characterise the $\beta(t)$ in terms of $u(t)$, as described in the following theorem.

Theorem 3.4.4. *If (u,β) obeys Definition 3.4.1, then for all $i \in V$ and a.e. $t \in T$,*

$$\beta_i(t) = \begin{cases} \frac{1}{2} + \varepsilon(\Delta u(t))_i - \varepsilon f_i, & \text{if } u_i(t) = 0, \\ 0, & \text{if } u_i(t) \in (0,1), \\ -\frac{1}{2} + \varepsilon(\Delta u(t))_i + \varepsilon(\mu_i - f_i), & \text{if } u_i(t) = 1, \end{cases} \tag{3.24}$$

and hence at a.e. $t \in T$,

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

If (u,β) obeys Definition 3.4.2, then for all $i \in V$ and a.e. $t \in T$,

$$\beta_i(t) - \overline{\beta(t)} = \begin{cases} \bar{u} + \varepsilon(\Delta u(t))_i, & \text{if } u_i(t) = 0, \\ -\overline{\beta(t)}, & \text{if } u_i(t) \in (0,1), \\ \bar{u} - 1 + \varepsilon(\Delta u(t))_i, & \text{if } u_i(t) = 1. \end{cases} \tag{3.25}$$

Proof. Since in either case $\beta(t) \in \mathcal{B}(u(t))$ at a.e. $t \in T$, for all $i \in V$ we have $\beta_i(t) = 0$ at a.e. $t \in T$ for which $u_i(t) \in (0,1)$, from which the $u_i(t) \in (0,1)$ case of (3.24) and (3.25) follow. For the remaining cases, first note that we can write both (3.22) and (3.23) in the form

$$\varepsilon \frac{du}{dt} + \varepsilon \Delta u(t) + v(t) = \gamma(t)$$

where in the former case $v(t) = \varepsilon M u(t) - \varepsilon f + \frac{1}{2}\mathbf{1} - u(t)$ and $\gamma(t) = \beta(t)$, and in the latter case $v(t) = -u(t) + \overline{u(t)}\mathbf{1}$ and $\gamma(t) = \beta(t) - \overline{\beta(t)}\mathbf{1}$.

Fix $i \in V$. Let $\tilde{T} \subseteq T$ denote the set of times when u_i is differentiable and has classical derivative equal to its weak derivative. Since $u_i(t) \in [0,1]$ at all times, when $t \in \tilde{T}$ and $u_i(t) \in \{0,1\}$ we have $du_i/dt = 0$. Then for a.e. such $t \in \tilde{T}$

$$0 = \varepsilon \frac{du_i}{dt}(t) = -\varepsilon(\Delta u(t))_i - v_i(t) + \gamma_i(t)$$

so, rearranging, at a.e. $t \in \tilde{T}$ with $u_i(t) \in \{0,1\}$

$$\gamma_i(t) = v_i(t) + \varepsilon(\Delta u(t))_i.$$

By a simple case check of the value of $v_i(t)$ for $u_i(t) = 0$ and for $u_i(t) = 1$ in the two cases, we see that (3.24)/(3.25) holds at a.e. $t \in \tilde{T}$. By Lemma 3.2.12, $T \setminus \tilde{T}$ is null, so (3.24)/(3.25) holds at a.e. $t \in T$.

Finally, since $u(t) \in \mathcal{V}_{[0,1]}$, if $u_i(t) = 0$ then i is a minimiser of $u(t)$, and hence $(\Delta u(t))_i \leq 0$. Likewise, if $u_i(t) = 1$ then $(\Delta u(t))_i \geq 0$. Hence if $\beta(t) \in \mathcal{B}(u(t))$ obeys (3.24), then if $u_i(t) = 0$ we have $\beta_i(t) \leq 1/2$ by (3.24), and if $u_i(t) > 0$ we have $\beta_i(t) \leq 0$ by (3.21), and hence $\beta_i(t) \leq 1/2$. Likewise, such a $\beta_i(t) \geq -1/2$. Hence we have the desired bounds at a.e. $t \in T$. □

Note 6. *A similar bound on β will be proved in the mass-conserving case (see Lemma 4.5.14), but requires further machinery.*

In light of this theorem, for brevity we will often refer to just u as a solution to (3.22) or (3.23) where we also understand that equation to inherit the conditions on u (including the existence of a corresponding β).

3.4.2. Comparison principle for the fidelity forced flow

Theorem 3.4.5. *Let $T = [0, T_0]$ or $[0, \infty)$, and let $(u, \beta), (v, \gamma) \in \mathcal{V}_{[0,1],t\in T} \times \mathcal{V}_{t\in T}$, with $u, v \in H^1_{loc}(T; \mathcal{V}) \cap C^0(T; \mathcal{V})$ be super- and subsolutions to (3.22), i.e. obeying*

$$\varepsilon \frac{du}{dt}(t) + \varepsilon A u(t) - \varepsilon f + \frac{1}{2}\mathbf{1} - u(t) \geq \beta(t), \qquad \beta(t) \in \mathcal{B}(u(t)), \tag{3.26a}$$

and

$$\varepsilon \frac{dv}{dt}(t) + \varepsilon A v(t) - \varepsilon f + \frac{1}{2}\mathbf{1} - v(t) \leq \gamma(t), \qquad \gamma(t) \in \mathcal{B}(v(t)), \tag{3.26b}$$

vertexwise at a.e. $t \in T$. Then if $v(0) \leq u(0)$ vertexwise, then $v(t) \leq u(t)$ vertexwise for all $t \in T$.

Proof. Let $w := v - u$, and let $w_+ := \max\{w, 0\}$, the pointwise positive part of w. Then we have that $w_+(0) = \mathbf{0}$ and seek to show that $w_+(t) = \mathbf{0}$ for all $t \in T$. By subtracting (3.26a) from (3.26b) and then taking the inner product with $w_+(t)$, we have that

$$\varepsilon \left\langle \frac{dw}{dt}(t), w_+(t) \right\rangle_{\mathcal{V}} + \varepsilon \langle Aw(t), w_+(t) \rangle_{\mathcal{V}} - \langle w(t), w_+(t) \rangle_{\mathcal{V}} \leq \langle \gamma(t) - \beta(t), w_+(t) \rangle_{\mathcal{V}} \tag{3.27}$$

that $\beta(t) \in \mathcal{B}(u(t))$, and that $\gamma(t) \in \mathcal{B}(v(t))$ at a.e. $t \in T$. At each such t, we make the following claims:

i. $\langle w(t), w_+(t) \rangle_{\mathcal{V}} \leq \langle w_+(t), w_+(t) \rangle_{\mathcal{V}}$.

ii. $\langle \gamma(t) - \beta(t), w_+(t) \rangle_{\mathcal{V}} \leq 0$.

iii. $\langle \Delta w(t), w_+(t) \rangle_{\mathcal{V}} \geq \langle \Delta w_+(t), w_+(t) \rangle_{\mathcal{V}} \geq 0$ and so $\langle Aw(t), w_+(t) \rangle_{\mathcal{V}} \geq 0$.

iv. $\int_0^{t^*} \left\langle \frac{dw}{dt}(t), w_+(t) \right\rangle_{\mathcal{V}} dt = \frac{1}{2}||w_+(t^*)||_{\mathcal{V}}^2 - \frac{1}{2}||w_+(0)||_{\mathcal{V}}^2 = \frac{1}{2}||w_+(t^*)||_{\mathcal{V}}^2$ for all $t^* \in T$.

Given (i-iv) it follows, by rearranging and integrating (3.27), that for all $t^* \in T$

$$\frac{1}{2}\varepsilon||w_+(t^*)||_{\mathcal{V}}^2 \leq \int_0^{t^*} ||w_+(t)||_{\mathcal{V}}^2 \, dt.$$

Thus since $||w_+(0)||_{\mathcal{V}}^2 = 0$, it follows that $||w_+(t)||_{\mathcal{V}}^2 \leq 0$ at all $t \in T$ by Grönwall's integral inequality [3], and therefore that $w_+(t) = \mathbf{0}$ for all $t \in T$.

It suffices then to prove (i-iv). Claim (i) is trivial, since $w_i(w_+)_i \leq (w_+)_i^2$ for all $i \in V$. To show (ii), consider $(v_i(t) - u_i(t))_+(\gamma_i(t) - \beta_i(t))$. If $v_i(t) \leq u_i(t)$ then the term is zero, and if $v_i(t) > u_i(t)$ then by (3.21) $\gamma_i(t) \leq \beta_i(t)$. Hence the term is non-positive for all $i \in V$, and thus (ii) follows.

Towards (iii), note that $\langle Aw(t), w_+(t) \rangle_{\mathcal{V}} = \langle \Delta w(t), w_+(t) \rangle_{\mathcal{V}} + \langle Mw(t), w_+(t) \rangle_{\mathcal{V}}$ and that since $\mu_i w_i(w_+)_i \geq 0$ for all $i \in V$ the latter term is non-negative. It suffices therefore to show that

$$0 \leq \langle \Delta(w(t) - w_+(t)), w_+(t) \rangle_{\mathcal{V}} = \langle \nabla w(t) - \nabla w_+(t), \nabla w_+(t) \rangle_{\mathcal{E}} = \frac{1}{2} \sum_{i,j \in V} \omega_{ij} X_{ij}$$

where $X_{ij} := ((w_+)_i(t) - (w_+)_j(t))(w_i(t) - (w_+)_i(t) - w_j(t) + (w_+)_j(t))$. It suffices to show that $X_{ij} \geq 0$ for all $i, j \in V$. WLOG suppose that $(w_+)_i(t) \geq (w_+)_j(t)$. If $(w_+)_i(t) = (w_+)_j(t)$ then $X_{ij} = 0$, and if $w_i(t), w_j(t) > 0$ then $X_{ij} = 0$. Finally if $w_i(t) > 0, w_j(t) \leq 0$ then $X_{ij} = -w_i(t)w_j(t) \geq 0$.

Finally, to show (iv) fix $i \in V$, $t^* \in T$, and let $x(t) := w_i(t)$. Then we desire that

$$\int_0^{t^*} \frac{dx}{dt}(t) x_+(t) \, dt = \frac{1}{2} x_+(t^*)^2 - \frac{1}{2} x_+(0)^2.$$

Since by Proposition 2.1.1 $x \in H^1((0, t^*); \mathbb{R}) \cap C^0([0, t^*]; \mathbb{R})$, this follows from [33, Lemma 3.3]. □

3.4.3. Weak forms and explicit integral forms

In this section, we prove first weak forms (Theorem 3.4.7) of the fidelity forced and the mass-conserving AC flows, and then explicit integral forms (Theorem 3.4.8). The weak forms are not used in the remainder of this thesis, however they are of general interest as they provide graph analogues of the *variational inequality form* of the continuum double-obstacle AC flow (see e.g. [6, (1.16)]), which played an important role in the analysis performed by Blowey and Elliott in [6, 7, 8]. The explicit integral forms we will use in section 4.5.3 to show the convergence of the SDIE scheme.

We first prove a useful lemma, which will turn up again when we analyse the SDIE scheme.

Lemma 3.4.6. *Let $u \in \mathcal{V}_{[0,1]}$ and $\beta \in \mathcal{B}(u)$. Then for all $\eta \in \mathcal{V}_{[0,1]}$, $\langle \beta, \eta - u \rangle_{\mathcal{V}} \geq 0$.*

Proof. Consider $\beta_i(\eta_i - u_i)$. If $u_i \in (0,1)$ then $\beta_i = 0$, so this term equals 0. If $u_i = 0$ then $\beta_i \geq 0$ and $\eta_i - u_i = \eta_i \geq 0$, so the term is non-negative. If $u_i = 1$ then $\beta_i \leq 0$ and $\eta_i - u_i = \eta_i - 1 \leq 0$, so the term is non-negative. Hence for all $i \in V$, $\beta_i(\eta_i - u_i) \geq 0$. □

Theorem 3.4.7 (Weak forms)**.** *Let $u \in \mathcal{V}_{[0,1],t\in T} \cap H^1_{loc}(T;\mathcal{V}) \cap C(T;\mathcal{V})$.*

There exists β such that (u,β) is a solution to (3.22) if and only if for a.e. $t \in T$ and all $\eta \in \mathcal{V}_{[0,1]}$

$$\left\langle \varepsilon \frac{du}{dt}(t) + \varepsilon A u(t) - \varepsilon f + \frac{1}{2}\mathbf{1} - u(t), \eta - u(t) \right\rangle_{\mathcal{V}} \geq 0. \tag{3.28}$$

Likewise, there exists β such that (u,β) is a solution to (3.23) if and only if for a.e. $t \in T$ and all $\eta \in \mathcal{V}_{[0,1]}$ such that $\mathcal{M}(\eta) = \mathcal{M}(u(t))$ (i.e. $\eta - u(t) \perp \mathbf{1}$), the following hold

$$\left\langle \varepsilon \frac{du}{dt}(t) + \varepsilon \Delta u(t) - u(t), \eta - u(t) \right\rangle_{\mathcal{V}} \geq 0, \tag{3.29a}$$

$$\left\langle \frac{du}{dt}(t), \mathbf{1} \right\rangle_{\mathcal{V}} = 0. \tag{3.29b}$$

Proof. Let (u,β) solve (3.22). Then at a.e. $t \in T$, $\beta(t) \in \mathcal{B}(u(t))$ and by (3.22) the LHS of (3.28) can be written as $\langle \beta(t), \eta - u(t) \rangle_{\mathcal{V}}$, and hence is non-negative for all $\eta \in \mathcal{V}_{[0,1]}$ by Lemma 3.4.6. Next, suppose that for a.e. $t \in T$, $u(t)$ obeys (3.28) for all $\eta \in \mathcal{V}_{[0,1]}$. Fix such a $t \in T$. Then defining $\beta(t) := \varepsilon \frac{du}{dt}(t) + \varepsilon A u(t) - \varepsilon f + \frac{1}{2}\mathbf{1} - u(t)$, we have supposed that $\langle \beta(t), \eta - u(t) \rangle_{\mathcal{V}} \geq 0$ for all $\eta \in \mathcal{V}_{[0,1]}$, and since (u,β) satisfies the ODE by definition it suffices to show that $\beta(t) \in \mathcal{B}(u(t))$. Let $i \in V$, and define $\eta, \eta' \in \mathcal{V}_{[0,1]}$ by: $\eta_j := u_j(t)$ for all $j \neq i$, $\eta_i := 0$, $\eta'_j := u_j(t)$ for all $j \neq i$, and $\eta'_i := 1$. Substituting η and η' into the above we have $\beta_i(t) u_i(t) \leq 0$ and

$\beta_i(t)(1-u_i(t)) \geq 0$. Therefore

$$\beta_i(t)\begin{cases} = 0, & u_i(t) \in (0,1) \\ \leq 0, & u_i(t) = 1 \\ \geq 0, & u_i(t) = 0 \end{cases}$$

so $\beta(t) \in \mathcal{B}(u(t))$.

The proof in the mass-conserving case follows the same pattern, but requires a bit more care. Let (u,β) solve (3.23). Then for a.e. $t \in T$ we have (3.29b) and $\beta(t) \in \mathcal{B}(u(t))$, and for all $\eta \in \mathcal{V}_{[0,1]}$ with $\eta - u(t) \perp \mathbf{1}$

$$\begin{aligned} LHS\,(3.29a) &= \left\langle \varepsilon \frac{du}{dt}(t) + \varepsilon \Delta u(t) - u(t) + \bar{u}\mathbf{1} + \overline{\beta(t)}\mathbf{1}, \eta - u(t) \right\rangle_{\mathcal{V}} \\ &= \langle \beta(t), \eta - u(t) \rangle_{\mathcal{V}} \geq 0 \end{aligned}$$

by Lemma 3.4.6.

Now let u satisfy (3.29) at a.e. $t \in T$. Therefore by (3.29a), for a.e. $t \in T$ and all $\eta \in \mathcal{V}_{[0,1]}$ with $\eta - u(t) \perp \mathbf{1}$

$$\left\langle \varepsilon \frac{du}{dt} - u(t) + \varepsilon \Delta u(t), \eta - u(t) \right\rangle_{\mathcal{V}} \geq 0$$

and so for any $\theta : T \to \mathbb{R}$, a.e. $t \in T$, and any η as before,

$$\left\langle \varepsilon \frac{du}{dt}(t) - u(t) + \varepsilon \Delta u(t) + \overline{u(t)}\mathbf{1} + \theta(t)\mathbf{1}, \eta - u(t) \right\rangle_{\mathcal{V}} \geq 0. \tag{3.30}$$

Fix $t \in T$ to be any such t. For a specific θ to be determined later, define

$$\beta(t) := \varepsilon \frac{du}{dt}(t) + \varepsilon \Delta u(t) - u(t) + \overline{u(t)}\mathbf{1} + \theta(t)\mathbf{1}. \tag{3.31}$$

By considering certain valid test functions η for (3.30), we will show that $\theta(t)$ can be chosen so that $\beta(t) \in \mathcal{B}(u(t))$. Towards this, for any $i,j \in V$ and $v \in \mathcal{V}$ we define the set

$$\Xi_{i,j,v} := \{\xi \in \mathcal{V} \mid \forall k \notin \{i,j\}, \xi_k = 0, \forall k \in \{i,j\}, \xi_k \in [-v_k, 1-v_k], \text{and } \mathcal{M}(\xi) = 0\}$$

which is constructed so that if $\xi \in \Xi_{i,j,u(t)}$ then $\eta := u(t) + \xi$ is a valid test function. Hence for any $\xi \in \Xi_{i,j,u(t)}$, by (3.30) and (3.31) we have that

$$d_i^r \xi_i \beta_i(t) + d_j^r \xi_j \beta_j(t) \geq 0$$

and so, since $\mathcal{M}(\xi) = 0$ (i.e. $d_i^r \xi_i + d_j^r \xi_j = 0$), for any $\xi \in \Xi_{i,j,u(t)}$ we have that

$$d_i^r \xi_i (\beta_i(t) - \beta_j(t)) \geq 0. \tag{3.32}$$

We now embark on a brief interlude on the existence of certain elements of $\Xi_{i,j,u(t)}$ with specific properties, given certain conditions on $u_i(t)$ and $u_j(t)$. This information will soon be of help to us in this proof.

Note 7. *If $u_i(t) = 0$ and $u_j(t) > 0$, then for $0 < \alpha \leq 1$ sufficiently small*

$$\xi_j := -\alpha u_j(t) \in [-u_j(t), 0) \qquad \xi_i := \alpha d_i^{-r} d_j^r u_j \in (0, 1 - u_i(t)]$$

defines a $\xi \in \Xi_{i,j,u(t)}$ with $\xi_i > 0$. Likewise, if $u_i(t) = 1$ and $u_j(t) < 1$, there is a $\xi \in \Xi_{i,j,u(t)}$ with $\xi_i < 0$, and if $u_i(t), u_j(t) \in (0,1)$, there exist $\xi, \xi' \in \Xi_{i,j,u(t)}$ with $\xi_i > 0$ and $\xi_i' < 0$.

Next, first suppose $u_j(t) \in (0,1)$ for some $j \in V$. Then we fix such a j and choose $\theta(t)$ so that $\beta_j(t) = 0$, and thus by (3.32) for any $i \in V$ and $\xi \in \Xi_{i,j,u(t)}$:

$$\xi_i \beta_i(t) \geq 0.$$

Then by the above note, if we choose a $\xi \in \Xi_{i,j,u(t)}$ with ξ_i of the appropriate sign,

$$\beta_i(t) \begin{cases} = 0, & \text{if } u_i(t) \in (0,1), \\ \leq 0, & \text{if } u_i(t) = 1, \\ \geq 0, & \text{if } u_i(t) = 0, \end{cases}$$

and so $\beta(t) \in \mathcal{B}(u(t))$.

Next, suppose no such j exists. By above if $u_i(t) = 0$ and $u_j(t) = 1$ then we can choose $\xi \in \Xi_{i,j,u(t)}$ with $\xi_i > 0$ and so by (3.32) we have that $\beta_j(t) \leq \beta_i(t)$. Thus we can choose $\theta(t)$ to add an appropriate constant to the values of $\beta(t)$ so that

$$0 \in \left[\max_{u_j(t)=1} \beta_j(t), \min_{u_i(t)=0} \beta_i(t) \right].$$

Hence we have

$$\beta_i(t) \begin{cases} \leq 0, & \text{if } u_i(t) = 1, \\ \geq 0, & \text{if } u_i(t) = 0, \end{cases}$$

so $\beta(t) \in \mathcal{B}(u(t))$. Therefore we can choose θ so that $\beta(t) \in \mathcal{B}(u(t))$ at a.e. $t \in T$.

Note finally that whatever the choice of θ, by (3.29b) and (3.31) we have at a.e. $t \in T$

$$\overline{\beta(t)} = \theta(t)$$

Hence by (3.31), at all such t

$$\varepsilon \frac{du}{dt} + \varepsilon \Delta u(t) - u(t) + \overline{u(t)} \mathbf{1} = \beta(t) - \overline{\beta(t)} \mathbf{1}$$

and, by choice of $\theta(t)$, $\beta(t) \in \mathcal{B}(u(t))$. Hence (u, β) solves (3.23). □

Theorem 3.4.8 (Explicit integral forms)**.** *Let $(u, \beta) \in \mathcal{V}_{[0,1],t\in T} \times \mathcal{V}_{t\in T}$, and recall F_t from Definition 3.2.5.*

Then (u, β) satisfies Definition 3.4.1 if and only if the following hold:

- *β is locally integrable,*

- *for a.e.* $t \in T$, $\beta(t) \in \mathcal{B}(u(t))$ *and* $\beta(t) \in \mathcal{V}_{[-1/2,1/2]}$, *and*
- *for all* $t \in T$ (*for* $B := A - \varepsilon^{-1} I$):

$$u(t) = e^{-tB}u(0) + F_t(B)\left(f - \frac{1}{2\varepsilon}\mathbf{1}\right) + \frac{1}{\varepsilon}\int_0^t e^{-(t-s)B}\beta(s)\, ds. \tag{3.33}$$

Furthermore (u, β) *satisfies Definition 3.4.2 if and only if the following hold:*

- $\beta - \bar{\beta}\mathbf{1}$ *is locally integrable,*
- *for a.e.* $t \in T$, $\beta(t) \in \mathcal{B}(u(t))$ *and* $\beta(t) - \bar{\beta}(t)\mathbf{1} \in \mathcal{V}_{[\bar{u}-1,\bar{u}]}$, *and*
- *for all* $t \in T$ (*for* $B := \Delta - \varepsilon^{-1} I$):

$$u(t) = \bar{u}\mathbf{1} + e^{-tB}\left(u(0) - \bar{u}\mathbf{1}\right) + \frac{1}{\varepsilon}\int_0^t e^{-(t-s)B}\left(\beta(s) - \bar{\beta}(s)\mathbf{1}\right) ds. \tag{3.34}$$

Note 8. *Previously published (namely, in [12, 13, 14]) versions of the results concerning the explicit integral form (as well as results concerning existence of solutions, well-posedness, monotonic decrease of Ginzburg–Landau, and Lipschitz continuity, all to be stated in sections 3.4.4 and 3.4.6 of this chapter) required a technical condition that* $\varepsilon^{-1} \notin \sigma(\Delta)$ *(or* $\sigma(A)$ *in the fidelity forced case). In this chapter, we slightly modify our approach to no longer require this condition. This will also remove the condition from the SDIE convergence result (Theorem 4.5.9) proved in the next chapter.*

Proof. Let (u, β) obey Definition 3.4.1 or 3.4.2. Note that we can rewrite both (3.22) and (3.23) in the form:

$$\varepsilon\frac{du}{dt}(t) + \varepsilon B u(t) - v = \gamma(t) \tag{3.35}$$

where $v = \varepsilon f - \frac{1}{2}\mathbf{1}$ and $\gamma = \beta$ in the former case, and $v = -\bar{u}\mathbf{1}$ and $\gamma = \beta - \bar{\beta}\mathbf{1}$ in the latter case. Then γ is a sum of a continuous function and the derivative of a H^1_{loc} function and hence is locally integrable. The a.e. pointwise bounds on γ follow from Theorem 3.4.4 in the former case and Lemma 4.5.14 in the latter case.

Finally (3.35) can be further rewritten:

$$\varepsilon\frac{d}{dt}\left(e^{tB}u(t)\right) = e^{tB}(\gamma(t) + v).$$

Hence, by the the 'fundamental theorem of calculus' on H^1 [10, Theorem 8.2], if $u \in H^1_{loc}(T; \mathcal{V}) \cap C^0(T; \mathcal{V})$ solves (3.22) or (3.23), then for all $t \in T$

$$e^{tB}u(t) - u(0) = \frac{1}{\varepsilon}\left(\int_0^t e^{sB}v\, ds + \int_0^t e^{sB}\gamma(s)\, ds\right) = \frac{1}{\varepsilon}e^{tB}F_t(B)v + \frac{1}{\varepsilon}\int_0^t e^{sB}\gamma(s)\, ds$$

(where we have used that $\int_0^t e^{sB}\,ds = e^{tB}F_t(B)$, which is simple to verify) and so

$$u(t) = e^{-tB}u(0) + \frac{1}{\varepsilon}F_t(B)v + \frac{1}{\varepsilon}e^{-tB}\int_0^t e^{sB}\gamma(s)\,ds. \tag{3.36}$$

Thus if u solves (3.22) then u solves (3.33). If u solves (3.23), then note that v is an eigenvector of Q with eigenvalue $-1/\varepsilon$ and so $F_t(B)v = F_t(-1/\varepsilon)v = \varepsilon(1-e^{t/\varepsilon})\bar{u}\mathbf{1}$. Since $e^{-tB}\mathbf{1} = e^{t/\varepsilon}\mathbf{1}$ we thus have that

$$u(t) = \bar{u}\mathbf{1} + e^{-tB}(u(0) - \bar{u}\mathbf{1}) + \frac{1}{\varepsilon}e^{-tB}\int_0^t e^{sB}\gamma(s)\,ds$$

and so u solves (3.34).

Now, let $(u,\beta) \in \mathcal{V}_{[0,1],t\in T} \times \mathcal{V}_{t\in T}$ satisfy, at a.e. $t \in T$, $\beta(t) \in \mathcal{B}(u(t))$, and $\gamma(t) \in \mathcal{V}_{[a,b]}$ (where the ordered pair (a,b) equals $(-1/2,1/2)$ in the former case and $(\bar{u}-1,\bar{u})$ in the latter case) is locally integrable, and let at all $t \in T$ $u(t)$ be given by (3.36). Then, differentiating (3.36), at all $t \in T$ u has formal weak derivative

$$\frac{du}{dt}(t) = -Be^{-tB}u(0) + \frac{1}{\varepsilon}e^{-tB}v + \frac{1}{\varepsilon}\gamma(t) - \frac{1}{\varepsilon}B\int_0^t e^{-(t-s)B}\gamma(s)\,ds$$

which can be checked to satisfy (3.35). All that remains is to check the regularity of u. The continuity of u is immediate, as it is a sum of two smooth terms and the integral of an essentially 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, an essentially bounded function, and the integral of an essentially bounded function, so is locally essentially bounded and hence locally L^2. □

Note 9. *The forward reference to Lemma 4.5.14 does not introduce circularity here, because we do not use this aspect of the forward direction of this theorem until after proving that lemma. We will however use the converse direction in proving the convergence of the semi-discrete scheme (Theorem 4.5.9).*

3.4.4. Existence and uniqueness theory

We here prove uniqueness results and state existence results (which we shall prove in section 4.5). These results in fact follow from standard gradient flow theory (see [1, Chapter 4 especially Theorem 4.0.4] for details; to apply this theory in the present case it is important to note that GL_ε is proper, coercive, lower semicontinuous and is (-1)-convex, and hence [1, (4.0.1) and Assumption 4.0.1] are satisfied). However, these techniques are more theoretically involved than is needed in the present case, so we will present these results with more elementary proofs.

We begin with uniqueness.

Theorem 3.4.9. *Let $T = [0,T_0]$ or $[0,\infty)$, and let (u,β) and (v,γ) satisfy $u,v \in \mathcal{V}_{[0,1],t\in T} \cap H^1_{loc}(T;\mathcal{V}) \cap C^0(T;\mathcal{V})$ and $u(0) = v(0)$.*

If (u,β) and (v,γ) solve (3.22), then $u(t) = v(t)$ for all $t \in T$ and $\beta(t) = \gamma(t)$ at a.e. $t \in T$.

If (u,β) and (v,γ) solve (3.23), then for all $t \in T$, $u(t) = v(t)$, and there exists $\tilde{T}$ such that $T \setminus \tilde{T}$ has zero measure and for all $t \in \tilde{T}$, $\beta(t) - \gamma(t) = (\bar{\beta}(t) - \bar{\gamma}(t))\mathbf{1}$. Furthermore, if $u_i(t) \in (0,1)$ for some $i \in V$ and $t \in \tilde{T}$, then $\beta(t) = \gamma(t)$.

Proof. In the case for (3.22), $u(t) = v(t)$ for all $t \in T$ follows immediately from Theorem 3.4.5 and therefore $\beta(t) = \gamma(t)$ at a.e. $t \in T$ follows from Theorem 3.4.4.

Next, let u and v solve (3.23). Then by subtracting and since $\bar{u} = \bar{v}$ we get for a.e. $t \in T$

$$\varepsilon \frac{d}{dt}(v(t) - u(t)) + \varepsilon\Delta(v(t) - u(t)) - (v(t) - u(t)) = (\gamma(t) - \beta(t)) + (\bar{\beta}(t) - \bar{\gamma}(t))\mathbf{1}.$$

Let $w := v - u$ and take the inner product with w, noting that $\langle w, \mathbf{1}\rangle_V = 0$,

$$\varepsilon\left\langle \frac{dw}{dt}, w(t)\right\rangle_V + \varepsilon\langle \Delta w(t), w(t)\rangle_V - \langle w(t), w(t)\rangle_V = \langle \gamma(t) - \beta(t), w(t)\rangle_V.$$

Consider $(v_i(t) - u_i(t))(\gamma_i(t) - \beta_i(t))$. If $v_i(t) = u_i(t)$ this equals 0. If $v_i(t) > u_i(t)$ then a simple case check of the possible values of $u_i(t)$ and $v_i(t)$ gives that $\gamma_i(t) \leq \beta_i(t)$ and likewise if $v_i(t) < u_i(t)$ then $\gamma_i(t) \geq \beta_i(t)$. Hence $\langle \gamma(t) - \beta(t), w(t)\rangle_V \leq 0$. Furthermore since Δ is positive semi-definite we have $\langle \Delta w(t), w(t)\rangle_V \geq 0$. Therefore by the above we have for a.e. $t \in T$,

$$\frac{1}{2}\varepsilon \frac{d}{dt}||w(t)||_V^2 \leq ||w(t)||_V^2$$

and note that $w(0) = \mathbf{0}$. Hence by Grönwall's differential inequality [24] we have that for all $t \in T$, $||w(t)||_V^2 \leq 0$. Therefore, for all $t \in T$, $v(t) = u(t)$.

Finally by Theorem 3.4.4, since $u = v$ on T, at a.e. $t \in T$ (in particular, at $t \in \tilde{T}$ for some $\tilde{T} \subseteq T$ with $T \setminus \tilde{T}$ of zero measure):

$$\beta_i(t) - \gamma_i(t) = \begin{cases} \bar{\beta}(t) - \bar{\gamma}(t), & \text{if } u_i(t) = 0, \\ 0, & \text{if } u_i(t) \in (0,1), \\ \bar{\beta}(t) - \bar{\gamma}(t), & \text{if } u_i(t) = 1. \end{cases}$$

Therefore at $t \in \tilde{T}$, either $\beta(t) - \gamma(t) = (\bar{\beta}(t) - \bar{\gamma}(t))\mathbf{1}$ or, if $u_i(t) \in (0,1)$ for some $i \in V$, then taking the average value of both sides we get

$$\bar{\beta}(t) - \bar{\gamma}(t) = (\bar{\beta}(t) - \bar{\gamma}(t))\overline{\chi_{\{i | u_i(t) \in \{0,1\}\}}}$$

so $\bar{\beta}(t) - \bar{\gamma}(t) = 0$ and hence $\beta(t) = \gamma(t)$ (and thus also $\beta(t) - \gamma(t) = (\bar{\beta}(t) - \bar{\gamma}(t))\mathbf{1}$). □

Note 10. *There are only $2^{|V|}$ distinct u such that $u_i \in \{0,1\}$ for all $i \in V$. Hence if $\overline{u(0)} \in [0,1] \setminus \{\bar{u} \mid u \in \mathcal{V} \text{ and } \forall i \in V, u_i \in \{0,1\}\}$, then we must have $\beta(t) = \gamma(t)$ for a.e. $t \in T$ (since $\overline{u(t)} = \overline{u(0)}$). Note that this set contains all but finitely many of the values of $[0,1]$.*

We now state existence results, which we shall prove in section 4.5.

Theorem 3.4.10. *Let $T = [0,\infty)$. Then for all $u_0 \in \mathcal{V}_{[0,1]}$, there exists (u,β) satisfying Definition 3.4.1 with $u(0) = u_0$, and (u,β) satisfying Definition 3.4.2 with $u(0) = u_0$.*

Proof. We prove this as Theorem 4.5.9, by taking the limit as $\tau \downarrow 0$ of the semi-discrete approximations. (We avoid circularity as we do not use this theorem until after we have proved Theorem 4.5.9.) □

3

3.4.5. Conditions for freezing

It was observed in [22, Theorem 5.3] (though for a different choice of W) that if ε is too small the AC flow "freezes", i.e. any $i \in V$ with $u_i(0) \approx 0$ has $u_i(t) \approx 0$ for all $t \geq 0$ (and likewise for $u_i(0) \approx 1$). We now show that a similar result holds true in the fidelity forced and mass-conserving cases with the double-obstacle potential.

Theorem 3.4.11. *Let $S \subseteq V$ and let $u(t) := \chi_S$ for all $t \in T$. Then u solves (3.22) if and only if*

$$\varepsilon \max_{i\in S^c} |(\Delta\chi_S)_i| \leq \frac{1}{2} - \varepsilon f_i, \qquad \text{and } \varepsilon \max_{i\in S} |(\Delta\chi_S)_i| \leq \frac{1}{2} + \varepsilon f_i - \varepsilon\mu_i. \tag{3.37}$$

Furthermore, u solves (3.23) if and only if $\varepsilon \max_{i\in S^c} |(\Delta\chi_S)_i| \leq 1 - \varepsilon \max_{i\in S} |(\Delta\chi_S)_i|$, which always holds if $\varepsilon \leq \frac{1}{2}||\Delta\chi_S||_\infty^{-1}$.

Proof. Note that $u(t) := \chi_S$ for all $t \in T$ satisfies $u \in H^1_{loc}(T;\mathcal{V}) \cap C^0(T;\mathcal{V}) \cap \mathcal{V}_{[0,1],t\in T}$, and has weak derivative $du/dt(t) = \mathbf{0}$ for all $t \in T$. Hence, such a u solves (3.22) or (3.23) if and only if there exists a $\beta \in \mathcal{V}_{t\in T}$ such that for a.e. $t \in T$, $\beta(t) \in \mathcal{B}(\chi_S)$ and

$$\varepsilon A\chi_S - \varepsilon f + \frac{1}{2}\mathbf{1} - \chi_S = \beta(t), \qquad \text{or} \qquad \varepsilon\Delta\chi_S - \chi_S + \overline{\chi_S}\mathbf{1} = \beta(t) - \overline{\beta(t)}\mathbf{1},$$

respectively.

In turn, this holds if and only if there exists a $\beta' \in \mathcal{V}$ such that for all $i \in S$, $\beta'_i \leq 0$, for all $i \in S^c$, $\beta'_i \geq 0$, and $\beta(t) := \beta'$ solves the above equations. In the fidelity forced case, this immediately gives the desired conditions.

In the mass-conserving case, we want to have that

$$\varepsilon\Delta\chi_S - \chi_S + \overline{\chi_S}\mathbf{1} = \beta' - \overline{\beta'}\mathbf{1}. \tag{3.38}$$

Observe that for all $\theta \in \mathbb{R}$, $\beta'' := \varepsilon\Delta\chi_S - \chi_S + \theta\mathbf{1}$ satisfies (3.38), and furthermore if some β''' satisfies (3.38) then $\beta'' - \overline{\beta''}\mathbf{1} = \beta''' - \overline{\beta'''}\mathbf{1}$ and so $\beta''' = \beta'' + (\overline{\beta'''} - \overline{\beta''})\mathbf{1}$. Thus all β' satisfying (3.38) are of the form $\beta' = \varepsilon\Delta\chi_S - \chi_S + \theta\mathbf{1}$.

Hence, u solves (3.23) if and only if there exists a $\theta \in \mathbb{R}$ such that for all $i \in S$, $\varepsilon(\Delta\chi_S)_i - 1 + \theta \leq 0$, and for all $i \in S^c$, $\varepsilon(\Delta\chi_S)_i + \theta \geq 0$. Note that by the definition of Δ, $(\Delta\chi_S)_i \geq 0$ for $i \in S$ and $(\Delta\chi_S)_i \leq 0$ for $i \in S^c$. Therefore, such a θ exists if and only if $[\varepsilon \max_{i\in S^c} |(\Delta\chi_S)_i|, 1 - \varepsilon \max_{i\in S} |(\Delta\chi_S)_i|]$ is non-empty. Finally, if $\varepsilon||\Delta\chi_S||_\infty \leq \frac{1}{2}$, then it suffices to take $\theta = \frac{1}{2}$. □

Note 11. *It is common in practice for $\tilde{f} = \chi_{Z'}$ for $Z' \subseteq Z$ (recall that $Z := \operatorname{supp} \mu$). If $\mu_i \varepsilon > \frac{1}{2}$ for all $i \in Z$ and S satisfies (3.37), then since the RHSs must be non-negative it follows that if $i \in S \cap Z$ then $i \in Z'$, and if $i \in S^c$ then $i \in Z'^c$. That is, we must have $S \cap Z = Z'$. In words, if the reference is binary and the fidelity forcing is sufficiently strong, then any frozen solutions must agree with the reference on the reference data.*

3.4.6. Miscellaneous properties to be proved in chapter 4

For completeness, we state three more important properties of this AC flow, which we will prove in the next chapter using the machinery we will develop there using the SDIE scheme.

Theorem 3.4.12 (Well-posedness of fidelity forced AC flow)**.** *Let $u_0, v_0 \in \mathcal{V}_{[0,1]}$, $T_0 \geq 0$, $T = [0, T_0]$ or $[0, \infty)$, and let $(u, \beta), (v, \gamma)$ be fidelity forced AC trajectories on T as in Definition 3.4.1 with $u(0) = u_0$ and $v(0) = v_0$. Then, for $\xi_1 := \min \sigma(A)$,*

$$||u(t) - v(t)||_{\mathcal{V}} \leq e^{-\xi_1 t} e^{t/\varepsilon} ||u_0 - v_0||_{\mathcal{V}}. \tag{3.39}$$

Proof. We prove this as Theorem 4.5.13 for the solution given by Theorem 4.5.9, which by uniqueness is the generic solution. □

Theorem 3.4.13 (Gradient flow property)**.** *For u as in Definition 3.4.1, $\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u(t))$ monotonically decreases, and for u as in Definition 3.4.2, $\mathrm{GL}_{\varepsilon}(u(t))$ monotonically decreases.*

Proof. We prove this as Theorem 4.5.12 for the solution given by Theorem 4.5.9, which by uniqueness is the generic solution.

In the case of u as in Definition 3.4.1 we here give a more direct proof. It suffices to show that at a.e. t,

$$\frac{d\, \mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u(t))}{dt} \leq 0.$$

Define

$$G_{\varepsilon,\mu,\tilde{f}}(u) := \frac{1}{2} \left|\left|\nabla u(t)\right|\right|_{\varepsilon}^2 + \frac{1}{2\varepsilon} \langle u(t), \mathbf{1} - u(t) \rangle_{\mathcal{V}} + \frac{1}{2} \langle u - \tilde{f}, M(u - \tilde{f}) \rangle_{\mathcal{V}}$$

then by (3.9) we have, for all t,

$$\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u(t)) = G_{\varepsilon,\mu,\tilde{f}}(u(t)) + \frac{1}{\varepsilon} \left\langle I_{[0,1]} \circ u(t), \mathbf{1} \right\rangle_{\mathcal{V}} = G_{\varepsilon,\mu,\tilde{f}}(u(t))$$

since $u(t) \in \mathcal{V}_{[0,1]}$ for all t. Hence, since $\nabla_{\mathcal{V}} G_{\varepsilon,\mu,\tilde{f}}(u) = Au - f + \frac{1}{\varepsilon}\left(\frac{1}{2}\mathbf{1} - u\right)$, we

note that

$$\begin{aligned}
&\varepsilon^2 \frac{d\,\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u(t))}{dt} \\
&= \varepsilon^2 \frac{dG_{\varepsilon,\mu,\tilde{f}}(u(t))}{dt} = \left\langle \varepsilon \frac{du}{dt}, \varepsilon Au(t) - \varepsilon f + \frac{1}{2}\mathbf{1} - u(t) \right\rangle_{\mathcal{V}} \\
&= \left\langle \beta(t) - \varepsilon Au(t) + \varepsilon f - \frac{1}{2}\mathbf{1} + u(t), \varepsilon Au(t) - \varepsilon f + \frac{1}{2}\mathbf{1} - u(t) \right\rangle_{\mathcal{V}}. \qquad (*)
\end{aligned}$$

By Theorem 3.4.4, at a.e. t and all $i \in V$, if $u_i(t) \in \{0,1\}$, then $\beta_i(t) - \varepsilon(Au(t))_i + \varepsilon f_i - \frac{1}{2} + u_i(t) = 0$, and if $u_i(t) \in (0,1)$, then $\beta_i(t) = 0$. Thus for a.e. t, let $V_t' := \{i \in V \mid u_i(t) \in (0,1)\}$, and so

$$(*) = -\sum_{i \in V_t'} d_i^r \left(\varepsilon(Au(t))_i - \varepsilon f_i + \frac{1}{2} - u_i(t) \right)^2 \leq 0$$

as desired. □

Theorem 3.4.14 (Lipschitz continuity of trajectories)**.** *If u satisfies Definition 3.4.1 or 3.4.2, then $u \in C^{0,1}(T;\mathcal{V})$.*

Proof. We prove this as Theorem 4.5.15. □

Bibliography

[1] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savare. *Gradient Flows*. 2nd ed. Lectures in Mathematics ETH Zürich. Basel: Birkhäuser, 1993. ISBN: 978-3-7643-8721-1. DOI: 10.1007/978-3-7643-8722-8.

[2] Egil Bae and E. Merkurjev. "Convex Variational Methods on Graphs for Multiclass Segmentation of High-Dimensional Data and Point Clouds". In: *Journal of Mathematical Imaging and Vision* 58 (2017), pp. 468–493.

[3] Richard Bellman. "The stability of solutions of linear differential equations". In: *Duke Mathematical Journal* 10.4 (1943), pp. 643–647. DOI: 10.1215/S0012-7094-43-01059-2.

[4] A. Bertozzi et al. "Uncertainty Quantification in Graph-Based Classification of High Dimensional Data". In: *SIAM/ASA J. Uncertain. Quantification* 6 (2018), pp. 568–595.

[5] Andrea Bertozzi and Arjuna Flenner. "Diffuse Interface Models on Graphs for Classification of High Dimensional Data". In: *Multiscale Modeling Simulation* 10 (July 2012), pp. 1090–1118. DOI: 10.1137/11083109X.

[6] J. F. Blowey and C. M. Elliott. "Curvature Dependent Phase Boundary Motion and Parabolic Double Obstacle Problems". In: *Degenerate Diffusions*. Ed. by Wei-Ming Ni, L. A. Peletier, and J. L. Vazquez. New York, NY: Springer New York, 1993, pp. 19–60. ISBN: 978-1-4612-0885-3.

[7] J. F. Blowey and C. M. Elliott. "The Cahn–Hilliard gradient theory for phase separation with non-smooth free energy Part I: Mathematical analysis". In: *European Journal of Applied Mathematics* 2.3 (1991), pp. 233–280. DOI: 10.1017/S095679250000053X.

[8] J. F. Blowey and C. M. Elliott. "The Cahn–Hilliard gradient theory for phase separation with non-smooth free energy Part II: Numerical analysis". In: *European Journal of Applied Mathematics* 3.2 (1992), pp. 147–179. DOI: 10.1017/S0956792500000759.

[9] Jessica Bosch, Steffen Klamt, and Martin Stoll. "Generalizing Diffuse Interface Methods on Graphs: Nonsmooth Potentials and Hypergraphs". In: *SIAM Journal on Applied Mathematics* 78.3 (Nov. 2018), pp. 1350–1377. DOI: 10.1137/17M1117835.

[10] Haim Brezis. *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. 1st ed. New York: Springer-Verlag, 2011. ISBN: 978-0-387-70913-0. DOI: 10.1007/978-0-387-70914-7.

[11] Lia Bronsard and Robert V. Kohn. "Motion by mean curvature as the singular limit of Ginzburg-Landau dynamics". English (US). In: *Journal of Differential Equations* 90.2 (Apr. 1991), pp. 211–237. DOI: 10.1016/0022-0396(91)90147-2.

[12] Jeremy Budd and Yves van Gennip. "Graph Merriman–Bence–Osher as a SemiDiscrete Implicit Euler Scheme for Graph Allen–Cahn Flow". In: *SIAM Journal on Mathematical Analysis* 52 (Jan. 2020), pp. 4101–4139. DOI: 10.1137/19M1277394.

[13] Jeremy Budd and Yves van Gennip. "Mass-conserving diffusion-based dynamics on graphs". In: *European Journal of Applied Mathematics* (Apr. 2021), pp. 1–49. DOI: 10.1017/S0956792521000061.

[14] Jeremy Budd, Yves van Gennip, and Jonas Latz. "Classification and image processing with a semi-discrete scheme for fidelity forced Allen–Cahn on graphs". English. In: *GAMM Mitteilungen* 44.1 (2021), pp. 1–43. ISSN: 0936-7195. DOI: 10.1002/gamm.202100004.

[15] John W Cahn. "On spinodal decomposition". In: *Acta Metallurgica* 9.9 (1961), pp. 795–801. ISSN: 0001-6160. DOI: 10.1016/0001-6160(61)90182-1. URL: https://www.sciencedirect.com/science/article/pii/0001616061901821.

[16] John W. Cahn and John E. Hilliard. "Free Energy of a Nonuniform System. I. Interfacial Free Energy". In: *The Journal of Chemical Physics* 28.2 (1958), pp. 258–267. DOI: 10.1063/1.1744102.

[17] Xinfu Chen and Charles Elliott. "Asymptotics for a Parabolic Double Obstacle Problem". In: *Proceedings of The Royal Society A: Mathematical, Physical and Engineering Sciences* 444 (Mar. 1994), pp. 429–445. DOI: 10.1098/rspa.1994.0030.

[18] Xinfu Chen, Danielle Hilhorst, and Elisabeth Logak. "Mass conserving Allen–Cahn equation and volume preserving mean curvature flow". In: *Interfaces and Free Boundaries* (2010), pp. 527–549. ISSN: 1463-9963. DOI: 10.4171/ifb/244.

[19] Ivar Ekeland and Roger Téman. *Convex Analysis and Variational Problems*. Vol. 28. Classics in Applied Mathematics. Philadelphia, USA: Society for Industrial and Applied Mathematics, 1999. ISBN: 978-0-89871-450-0. DOI: 10.1137/1.9781611971088.

[20] Selim Esedoglu and Yen-Hsi Richard Tsai. "Threshold dynamics for the piecewise constant Mumford–Shah functional". In: *Journal of Computational Physics* 211.1 (2006), pp. 367–384. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2005.05.027. URL: https://www.sciencedirect.com/science/article/pii/S0021999105002792.

[21] Lawrence C. Evans. "Convergence of an Algorithm for Mean Curvature Motion". In: *Indiana University Mathematics Journal* 42.2 (1993), pp. 533–557. ISSN: 00222518, 19435258. URL: http://www.jstor.org/stable/24897106.

[22] Y. van Gennip et al. "Mean Curvature, Threshold Dynamics, and Phase Field Theory on Finite Graphs". In: *Milan Journal of Mathematics* 82 (2014), pp. 3–65.

[23] Yves van Gennip. "An MBO Scheme for Minimizing the Graph Ohta–Kawasaki Functional". In: *Journal of Nonlinear Science* 30.2 (Oct. 2020), pp. 2325–2373. DOI: 10.1007/s00332-018-9468-8.

[24] T. H. Gronwall. "Note on the Derivatives with Respect to a Parameter of the Solutions of a System of Differential Equations". In: *Annals of Mathematics* 20.4 (1919), pp. 292–296. ISSN: 0003486X. URL: http://www.jstor.org/stable/1967124.

[25] Tim Laux and Drew Swartz. "Convergence of thresholding schemes incorporating bulk effects". In: *Interfaces and Free Boundaries* 19 (2017), pp. 273–304. DOI: 10.4171/IFB/383.

[26] Giovanni Leoni. *A First Course in Sobolev Spaces*. Vol. 105. Graduate Studies in Mathematics. Providence, RI: American Mathematical Society, 2009. ISBN: 978-1-4704-1169-5. DOI: 10.1090/gsm/105.

[27] Ekaterina Merkurjev, Tijana Kostić, and Andrea Bertozzi. "An MBO Scheme on Graphs for Classification and Image Processing". In: *SIAM Journal on Imaging Sciences* 6 (Oct. 2013), pp. 1903–1910. DOI: 10.1137/120886935.

[28] Luca Mugnai, Christian Seis, and Emanuele Spadaro. "Global solutions to the volume-preserving mean-curvature flow". In: *Calculus of Variations and Partial Differential Equations* 55 (Feb. 2016). DOI: 10.1007/s00526-015-0943-x.

[29] Y. Oono and S. Puri. "Study of phase-separation dynamics by use of cell dynamical systems. I. Modeling". In: *Phys. Rev. A* 38 (1 June 1988), pp. 434–453. DOI: 10.1103/PhysRevA.38.434. URL: https://link.aps.org/doi/10.1103/PhysRevA.38.434.

[30] Jacob Rubinstein and Peter Sternberg. "Nonlocal reaction-diffusion equations and nucleation". In: *IMA Journal of Applied Mathematics* 48.3 (Sept. 1992), pp. 249–264. ISSN: 0272-4960. DOI: 10.1093/imamat/48.3.249. eprint: https://academic.oup.com/imamat/article-pdf/48/3/249/6765794/48-3-249.pdf.

[31] Steven J. Ruuth and B. Wetton. "A Simple Scheme for Volume-Preserving Motion by Mean Curvature". In: *Journal of Scientific Computing* 19 (2003), pp. 373–384.

[32] Gerald Teschl. *Ordinary differential equations and dynamical systems*. Vol. 140. Graduate Studies in Mathematics. Providence, RI: American Mathematical Society, 2012. ISBN: 978-0821883280. DOI: 10.1090/gsm/140.

[33] Daniel Wachsmuth. *The regularity of the positive part of functions in* $L^2(I;H^1(\Omega))\cap H^1(I;H^1(\Omega)^*)$ *with applications to parabolic equations*. 2016. arXiv: 1604.04392 [math.AP].

4

The SDIE link between Allen–Cahn flow and the MBO scheme

The problem with modern mathematics is that it's all in black boxes and you can't see the whirry bits going on inside.

T. W. Körner, during a 1982 Tripos lecture (quoted by Chris Budd)

In chapter 3, we defined and examined the properties of Allen–Cahn (AC) flow and the Merriman–Bence–Osher (MBO) scheme on a graph, including under mass-conserving and fidelity forcing constraints, and also defined the semi-discrete implicit Euler (SDIE) scheme for AC flow. In particular, we defined a graph AC flow against the non-differentiable double-obstacle potential. Furthermore, we noted that in the continuum setting these two processes are known to be deeply linked. In this chapter, we rigorously demonstrate that these processes are also linked in the graph setting. In particular, we shall prove that the graph MBO scheme is a special case of an SDIE scheme for the double-obstacle AC flow, and that this fact remains true in the mass-conserving and fidelity forced cases. Furthermore, we shall give an explicit form for the solution to the SDIE scheme, and show that as its time step tends to zero the SDIE scheme converges to a solution to the AC flow.

Parts of this chapter have been published in *SIAM J. Math. Anal.* 52 (2020) [3], *Eur. J. Appl. Math.* (2021) [4], and *GAMM Mitteilungen* 44 (2021) [5].

In this chapter, we will rigorously link together the graph MBO scheme and the double-obstacle AC flow via our SDIE scheme, using the theory developed for the double-obstacle AC flow in the last chapter. To keep the wood visible for the many trees, we will begin with a summary of the key results of this chapter.

First, we will define SDIE schemes for each of the AC flows defined in the previous chapter. We will prove that these numerical schemes are equivalent to variational equations, and then investigate the solution to these variational schemes. We will discover that the SDIE updates are given by a diffusion followed by a piecewise linear thresholding, with the MBO thresholding as a special case. We will furthermore show that as the parameters of the SDIE scheme converge to those corresponding to the MBO special case, the SDIE solutions given by those parameters converge to an MBO solution. We will also find conditions under which the SDIE schemes freeze, connecting to the results of Van Gennip *et al.* [9] regarding the freezing of the MBO scheme.

Next, by defining Lyapunov functionals for the SDIE schemes, we will investigate their long-time behaviour. We will show that the MBO special cases are eventually constant (under certain conditions), and that in the non-MBO cases the distances between consecutive terms in an SDIE sequence are square-summable (and hence converge to zero) and that the SDIE sequences converge along a sub-sequence, however we will be unable to prove convergence of the whole sequence from these facts. We will also examine the properties of the gradient of the Lyapunov functional.

Finally, we will show the convergence of the SDIE schemes as their time step tends to zero. We will prove that the SDIE schemes converge pointwise to solutions of the corresponding AC flows, proving that solutions to those flows exist. We will furthermore use this characterisation of the AC flow solutions to prove the results about the AC flows which were stated in section 3.4.6 of the last chapter.

4.1. The SDIE schemes, and the link to the MBO scheme

We now define the fidelity forced and mass-conserving SDIE schemes for W the double-obstacle potential.

Definition 4.1.1 (SDIE scheme with fidelity forcing)**.** *For $u_0 \in \mathcal{V}_{[0,1]}$, $n \in \mathbb{N}$, and $\lambda := \tau/\varepsilon \in [0,1]$ we define the fidelity forced SDIE scheme iteratively:*

$$(1-\lambda)u_{n+1} - \delta_\tau u_n + \frac{\lambda}{2}\mathbf{1} = \lambda\beta_{n+1} \tag{4.1}$$

for $\beta_{n+1} \in \mathcal{B}(u_{n+1})$.

Definition 4.1.2 (Mass-conserving SDIE scheme)**.** *For $u_0 \in \mathcal{V}_{[0,1]}$, $n \in \mathbb{N}$, and $\lambda \in [0,1]$ we define the mass-conserving SDIE scheme iteratively:*

$$u_{n+1} - e^{-\tau\Delta}u_n - \lambda u_{n+1} + \lambda\overline{u_{n+1}}\mathbf{1} = \lambda\beta_{n+1} - \lambda\overline{\beta_{n+1}}\mathbf{1} \tag{4.2}$$

for $\beta_{n+1} \in \mathcal{B}(u_{n+1})$.

Note 12. *Recall that since in both of these schemes $\mathcal{B}(u_n)$ is non-empty for all n, we must have $u_n \in \mathcal{V}_{[0,1]}$ for all n. Note also that* a priori, *these definitions permit non-unique trajectories. However, for $\lambda < 1$ we will have as a consequence of Theorem 4.1.4 that the updates are in fact unique. The $\lambda = 1$ case we will prove to be the MBO special case, which has non-unique solutions, which we shall characterise in sections 4.2.1 and 4.2.4 in the fidelity forced and mass-conserving cases respectively.*

We check that the latter scheme conserves mass.

Proposition 4.1.3. *For u_{n+1} given by (4.2),*

$$\mathcal{M}(u_{n+1}) = \mathcal{M}(u_n).$$

Proof. Taking the mass of both sides of (4.2) and cancelling gives

$$\langle u_{n+1}, \mathbf{1}\rangle_{\mathcal{V}} = \langle e^{-\tau\Delta}u_n, \mathbf{1}\rangle_{\mathcal{V}} = \langle u_n, \mathbf{1}\rangle_{\mathcal{V}}$$

with the final equality because $e^{-\tau\Delta}$ is self-adjoint and $e^{-\tau\Delta}\mathbf{1} = \mathbf{1}$. □

We now express these schemes variationally, and link them to the MBO scheme.

Theorem 4.1.4. *Suppose $\lambda := \tau/\varepsilon \in [0,1]$.*
If (u_{n+1}, β_{n+1}) solves (4.1) with $\beta_{n+1} \in \mathcal{B}(u_{n+1})$, then u_{n+1} solves:

$$u_{n+1} \in \operatorname*{argmin}_{u\in\mathcal{V}_{[0,1]}} \lambda\langle u, \mathbf{1}-u\rangle_{\mathcal{V}} + ||u - \mathcal{S}_\tau u_n||_{\mathcal{V}}^2. \tag{4.3}$$

Note that for $\lambda = 1$ (4.3) is equivalent to the variational problem (3.8) that defines the fidelity forced MBO scheme.
If (u_{n+1}, β_{n+1}) solves (4.2) with $\beta_{n+1} \in \mathcal{B}(u_{n+1})$, then u_{n+1} solves:

$$\begin{aligned} u_{n+1} \in \operatorname*{argmin}_{\substack{u\in\mathcal{V}_{[0,1]}\\ \mathcal{M}(u)=\mathcal{M}(u_n)}} & \lambda\langle u, \mathbf{1}-u\rangle_{\mathcal{V}} + \left|\left|u - e^{-\tau\Delta}u_n\right|\right|_{\mathcal{V}}^2 \\ & \simeq (1-\lambda)\,||u||_{\mathcal{V}}^2 - 2\langle u, e^{-\tau\Delta}u_n\rangle_{\mathcal{V}}. \end{aligned} \tag{4.4}$$

In particular, when $\lambda = 1$ we have

$$u_{n+1} \in \operatorname*{argmax}_{\substack{u\in\mathcal{V}_{[0,1]}\\ \mathcal{M}(u)=\mathcal{M}(u_n)}} \left\langle u, e^{-\tau\Delta}u_n\right\rangle_{\mathcal{V}} \tag{4.5}$$

which is equivalent to the mass-conserving MBO scheme as in Definition 3.2.10.

Proof. Let (u_{n+1}, β_{n+1}) solve (4.1) or (4.2) with $\beta_{n+1} \in \mathcal{B}(u_{n+1})$. Let $z := \mathcal{S}_\tau u_n$ in the former case or $z := e^{-\tau\Delta}u_n$ in the latter case. We seek to show that for $0 \le \lambda \le 1$

$$\lambda\langle u_{n+1}, \mathbf{1}-u_{n+1}\rangle_{\mathcal{V}} + \langle u_{n+1}-z, u_{n+1}-z\rangle_{\mathcal{V}} \le \lambda\langle \eta, \mathbf{1}-\eta\rangle_{\mathcal{V}} + \langle \eta - z, \eta - z\rangle_{\mathcal{V}}$$

either for all $\eta \in \mathcal{V}_{[0,1]}$ (in the former case) or for all $\eta \in \mathcal{V}_{[0,1]}$ with $\eta - u_{n+1} \perp \mathbf{1}$ (in the latter case). By rearranging and cancelling this is equivalent to

$$\begin{aligned} 0 &\leq \langle \eta - u_{n+1}, \lambda\mathbf{1} - 2z\rangle_{\mathcal{V}} + (1-\lambda)\left(\langle \eta, \eta\rangle_{\mathcal{V}} - \langle u_{n+1}, u_{n+1}\rangle_{\mathcal{V}}\right) \\ &= \langle \eta - u_{n+1}, \lambda\mathbf{1} - 2z + (1-\lambda)(\eta + u_{n+1})\rangle_{\mathcal{V}} \\ &= \langle \eta - u_{n+1}, 2\lambda\beta_{n+1} + (1-\lambda)(\eta - u_{n+1})\rangle_{\mathcal{V}} \\ &= 2\lambda\,\langle \eta - u_{n+1}, \beta_{n+1}\rangle_{\mathcal{V}} + (1-\lambda)||\eta - u_{n+1}||_{\mathcal{V}}^2 \end{aligned}$$

where the second equality follows directly from (4.1) in the former case, and in the latter case from (4.2) via adding $\lambda(-1 + 2\bar{u} + 2\overline{\beta_{n+1}})\mathbf{1}$ to the latter term in the inner product (which doesn't affect the product since $\eta - u_{n+1} \perp \mathbf{1}$). Finally, we have by Lemma 3.4.6 that this inequality holds for all $\eta \in \mathcal{V}_{[0,1]}$. □

Note 13. *This theorem shows that solutions to either of the SDIE schemes solve a corresponding variational equation. A natural question, which will be the topic of the next section, is whether the converse holds. That is, are these variational equations equivalent to the SDIE schemes?*

4.2. Solving the variational form, and the converse of Theorem 4.1.4

In this section, we characterise the solutions to the above variational equations for the SDIE schemes, and show that these solutions satisfy the definitions of those schemes. We will begin with the fidelity forced case, which will take us only a single subsection, and the remainder of this section will be devoted to the mass-conserving case.

4.2.1. The fidelity forced case

Theorem 4.2.1. *The variational equation (4.3) has unique solution for $\lambda \in (0,1)$*[1]

$$(u_{n+1})_i = \begin{cases} 0, & \text{if } (\mathcal{S}_\tau u_n)_i < \frac{1}{2}\lambda, \\ \frac{1}{2} + \frac{(\mathcal{S}_\tau u_n)_i - 1/2}{1-\lambda}, & \text{if } \frac{1}{2}\lambda \leq (\mathcal{S}_\tau u_n)_i < 1 - \frac{1}{2}\lambda, \\ 1, & \text{if } (\mathcal{S}_\tau u_n)_i \geq 1 - \frac{1}{2}\lambda, \end{cases} \tag{4.6}$$

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

$$(u_{n+1})_i \in \begin{cases} \{1\}, & (\mathcal{S}_\tau u_n)_i > 1/2, \\ [0,1], & (\mathcal{S}_\tau u_n)_i = 1/2, \\ \{0\}, & (\mathcal{S}_\tau u_n)_i < 1/2, \end{cases} \tag{4.7}$$

(i.e. the MBO thresholding) with corresponding $\beta_{n+1} = \frac{1}{2}\mathbf{1} - \mathcal{S}_\tau u_n$.

[1] If $\lambda = 0$ then $\mathcal{S}_\tau u_n = u_n$ and so we have trivial solution $u_{n+1} = u_n$. It follows that we can take $\beta_{n+1} = \frac{1}{2}\mathbf{1} - u_n \in \mathcal{B}(u_{n+1})$.

Hence if u_{n+1} solves (4.3) then there exists $\beta_{n+1} \in \mathcal{B}(u_{n+1})$ such that (u_{n+1}, β_{n+1}) solves (4.1).

Proof. Let u solve (4.3). The functional in (4.3) can be rewritten as

$$\lambda \langle u, \mathbf{1} - u\rangle_{\mathcal{V}} + ||u - \mathcal{S}_\tau u_n||_{\mathcal{V}}^2 = \sum_{i \in V} d_i^r g_{i,n}(u_i)$$

where

$$g_{i,n}(x) := \lambda x(1-x) + \left(x - (\mathcal{S}_\tau u_n)_i\right)^2$$

so we can reduce (4.3) to the system of 1-dimensional problems

$$(u_{n+1})_i \in \operatorname*{argmin}_{x \in [0,1]} g_{i,n}(x).$$

Differentiating, we get that for $0 < \lambda < 1$, $g_{i,n}$ is minimised at

$$x = \frac{(\mathcal{S}_\tau u_n)_i - \lambda/2}{1-\lambda} = \frac{1}{2} + \frac{(\mathcal{S}_\tau u_n)_i - 1/2}{1-\lambda}.$$

Therefore for $0 \le \lambda < 1$ the solution u is given by

$$u_i = \begin{cases} 0, & \text{if } (\mathcal{S}_\tau u_n)_i < \frac{1}{2}\lambda \\ \frac{1}{2} + \frac{(\mathcal{S}_\tau u_n)_i - 1/2}{1-\lambda}, & \text{if } \frac{1}{2}\lambda \le \left(e^{-\tau\Delta} u_n\right)_i < 1 - \frac{1}{2}\lambda \\ 1, & \text{if } (\mathcal{S}_\tau u_n)_i \ge 1 - \frac{1}{2}\lambda \end{cases}$$

and hence

$$\begin{aligned} &\lambda^{-1}\left((1-\lambda)u_i - (\mathcal{S}_\tau u_n)_i + \frac{\lambda}{2}\right) \\ &= \begin{cases} \frac{1}{2} - \lambda^{-1}(\mathcal{S}_\tau u_n)_i, & \text{if } (\mathcal{S}_\tau u_n)_i < \frac{1}{2}\lambda, \\ 0, & \text{if } \frac{1}{2}\lambda \le (\mathcal{S}_\tau u_n)_i < 1 - \frac{1}{2}\lambda, \\ -\frac{1}{2} + \lambda^{-1}(1 - (\mathcal{S}_\tau u_n)_i), & \text{if } (\mathcal{S}_\tau u_n)_i \ge 1 - \frac{1}{2}\lambda. \end{cases} \\ &= \begin{cases} \frac{1}{2} - \lambda^{-1}(\mathcal{S}_\tau u_n)_i, & \text{if } u_i = 0, \\ 0, & \text{if } u_i \in (0,1), \\ -\frac{1}{2} + \lambda^{-1}(1 - (\mathcal{S}_\tau u_n)_i), & \text{if } u_i = 1. \end{cases} \end{aligned}$$

Thus, noting that the top case has a non-negative value and the bottom case always has a non-positive value, we observe that $\beta := \lambda^{-1}\left((1-\lambda)u - \mathcal{S}_\tau u_n + \frac{\lambda}{2}\mathbf{1}\right) \in \mathcal{B}(u)$, so (u, β) solves (4.1).

If $\lambda = 1$ then examine the functional in (4.3) for $\lambda = 1$:

$$\begin{aligned} &\langle u, \mathbf{1} - u\rangle_{\mathcal{V}} + ||u - \mathcal{S}_\tau u_n||_{\mathcal{V}}^2 \\ &= \langle u, \mathbf{1} - u\rangle_{\mathcal{V}} + \langle u - \mathcal{S}_\tau u_n, u - \mathcal{S}_\tau u_n\rangle_{\mathcal{V}} \\ &= \langle u, \mathbf{1}\rangle_{\mathcal{V}} - \langle u, u\rangle_{\mathcal{V}} + \langle u, u\rangle_{\mathcal{V}} - 2\langle u, \mathcal{S}_\tau u_n\rangle_{\mathcal{V}} + \langle \mathcal{S}_\tau u_n, \mathcal{S}_\tau u_n\rangle_{\mathcal{V}} \\ &\simeq \langle u, \mathbf{1} - 2\mathcal{S}_\tau u_n\rangle_{\mathcal{V}}, \end{aligned}$$

and therefore u as a minimiser must obey

$$u_i \in \begin{cases} \{1\}, & (\mathcal{S}_\tau u_n)_i > 1/2, \\ [0,1], & (\mathcal{S}_\tau u_n)_i = 1/2, \\ \{0\}, & (\mathcal{S}_\tau u_n)_i < 1/2. \end{cases}$$

Hence $\beta \in \mathcal{B}(u)$ if and only if for each $i \in V$

$$\beta_i \in \begin{cases} [0,\infty), & (\mathcal{S}_\tau u_n)_i \le 1/2 \\ \{0\}, & (\mathcal{S}_\tau u_n)_i = 1/2, u_i \in (0,1) \\ (-\infty,0], & (\mathcal{S}_\tau u_n)_i \ge 1/2 \end{cases}$$

and thus $\frac{1}{2}\mathbf{1} - \mathcal{S}_\tau u_n \in \mathcal{B}(u)$, so (u,β) solves (4.1). □

We note a useful consequence of this result.

Theorem 4.2.2. *For $\lambda \in [0,1)$[2] and all $n \in \mathbb{N}$, if u_n and v_n are SDIE sequences defined according to Definition 4.1.1 with initial states $u_0, v_0 \in \mathcal{V}_{[0,1]}$ and $\xi_1 := \min \sigma(A)$ then*

$$||u_n - v_n||_\mathcal{V} \le e^{-n\xi_1\tau}(1-\lambda)^{-n}||u_0 - v_0||_\mathcal{V}. \tag{4.8}$$

Proof. If $\lambda = 0$ (and thus $\tau = 0$) then $u_n \equiv u_0$ and $v_n \equiv v_0$ and so the result trivially follows.

For $\lambda > 0$, let $\rho_\lambda : \mathcal{V}_{[0,1]} \to \mathcal{V}_{[0,1]}$ be the thresholding operator in (4.6), i.e.

$$(\rho_\lambda(u))_i := \begin{cases} 0, & \text{if } u_i < \frac{1}{2}\lambda, \\ \frac{1}{2} + \frac{u_i - 1/2}{1-\lambda}, & \text{if } \frac{1}{2}\lambda \le u_i < 1 - \frac{1}{2}\lambda, \\ 1, & \text{if } u_i \ge 1 - \frac{1}{2}\lambda. \end{cases}$$

Then by Theorem 4.2.1 it follows that

$$u_n = (\rho_\lambda \circ \mathcal{S}_\tau)^n(u_0)$$

and likewise for v_n. Finally, note that $\mathcal{S}_\tau u$ is affine in u, and hence is Lipschitz with constant $e^{-\xi_1\tau}$, and $\rho_\lambda(u)$ is piecewise affine in u with greatest slope $(1-\lambda)^{-1}$ and hence is Lipschitz with constant $(1-\lambda)^{-1}$. Thus $(\rho_\lambda \circ \mathcal{S}_\tau)^n$ is Lipschitz with constant $e^{-n\tau\xi_1}(1-\lambda)^{-n}$. □

4.2.2. Set-up for the mass-conserving case

Compared to the fidelity forced case, the addition of the mass conservation constraint substantially increases the difficulty in solving the equations from Theorem 4.1.4. We here employ the techniques of convex optimisation, particularly the Krein–Milman theorem, complementary slackness, and strong duality, to help resolve this difficulty.

We consider the set of feasible solutions to (4.4) and (4.5).

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

Definition 4.2.3. *For a given $M \in [0, \mathcal{M}(\mathbf{1})]$, we define the hyperplane $S_M := \{u \in \mathcal{V} \mid \langle u, \mathbf{1}\rangle_{\mathcal{V}} = M\}$. We can visualise this as the plane through some $u_0 \in S_M$ with $\mathcal{V}$-normal vector $\mathbf{1}$. Then we write the set of feasible solutions to (4.4) and (4.5)*

$$X_M := \mathcal{V}_{[0,1]} \cap S_M. \tag{4.9}$$

Note 14. *We have that X_M is compact (since the topology on $\mathcal{V}$ is equivalent to the standard topology on $\mathbb{R}^{|V|}$, and X_M is closed and bounded), and is the intersection of two convex sets, so is convex. Furthermore, note that X_M can be described as the set of solutions to linear inequalities, in particular*

$$\forall i \in V\ \langle u, \chi_{\{i\}}\rangle_{\mathcal{V}} \geq 0 \text{ and } \langle u, \chi_{\{i\}}\rangle_{\mathcal{V}} \leq d_i^r \quad \text{and} \quad \langle u, \mathbf{1}\rangle_{\mathcal{V}} \geq M \text{ and } \langle u, \mathbf{1}\rangle_{\mathcal{V}} \leq M,$$

and thus X_M is said to be a polyhedral *set.*

Definition 4.2.4. *For a convex set C, define $x \in C$ to be an* extreme point *of C when*

$$\forall y, z \in C, \forall t \in (0,1)\ \ \big(x = ty + (1-t)z \Rightarrow y = z = x\big)$$

and write $\operatorname{Ext} C$ for the subset of C consisting of all such points.

We can then characterise the extreme points of the feasible set.

Proposition 4.2.5. *The set $\operatorname{Ext} X_M$ of extreme points of X_M is finite and is given by*

$$\operatorname{Ext} X_M = \left\{u \in X_M \,\middle|\, \exists i^* \in V\ \forall j \in V \setminus \{i^*\}\ u_j \in \{0,1\}\right\}.$$

Proof. Since X_M is polyhedral, $\operatorname{Ext} X_M$ is finite by a standard result [12, Corollary 1.3.1]. Suppose $u \in X_M$ and $\exists i, j \in V$ such that $i \neq j$ and $u_i, u_j \in (0,1)$. Now for $\delta > 0$ let

$$v_1 := u - \delta d_i^{-r}\chi_{\{i\}} + \delta d_j^{-r}\chi_{\{j\}},$$
$$v_2 := u + \delta d_i^{-r}\chi_{\{i\}} - \delta d_j^{-r}\chi_{\{j\}}.$$

Then $\mathcal{M}(v_1) = \mathcal{M}(v_2) = \mathcal{M}(u) - \delta + \delta = \mathcal{M}(u) = M$ so $v_1, v_2 \in S_M$. And for $\delta < \min\left\{d_i^r u_i, d_i^r(1-u_i), d_j^r u_j, d_j^r(1-u_j)\right\}$ we have $v_1, v_2 \in \mathcal{V}_{[0,1]}$. Therefore we have $u = \frac{1}{2}v_1 + \frac{1}{2}v_2$ for $v_1, v_2 \in X_M \setminus \{u\}$. Therefore $u \notin \operatorname{Ext} X_M$.

Now let $u \in \left\{u \in X_M \,\middle|\, \exists i^* \in V\ \forall j \in V \setminus \{i^*\}\ u_j \in \{0,1\}\right\}$, and suppose $u = tv_1 + (1-t)v_2$ for some $v_1, v_2 \in X_M$ and $0 < t < 1$. As $\operatorname{Ext}([0,1]) = \{0,1\}$ we have that $u_i = 0$ if and only if $(v_1)_i = (v_2)_i = 0$ and likewise for $u_i = 1$. So $v_1 - v_2 = \theta\chi_{\{i^*\}}$ for some θ, and

$$0 = \langle v_1 - v_2, \mathbf{1}\rangle_{\mathcal{V}} = \theta\langle\chi_{\{i^*\}}, \mathbf{1}\rangle_{\mathcal{V}} = \theta d_{i^*}^r$$

and so $\theta = 0$, i.e. $v_1 = v_2$. Thus $u = tv_1 + (1-t)v_2 \Rightarrow v_1 = v_2 = u$, so $u \in \operatorname{Ext} X_M$. □

For tidiness, we define some useful notation.

Definition 4.2.6. *For $u \in \mathcal{V}_{[0,1]}$ and $\tau > 0$ define the set*

$$A_{u,\tau} := \{\alpha \in [0,1] \,|\, \exists i \in V \ (e^{-\tau\Delta}u)_i = \alpha\} \tag{4.10}$$

with ordering $\alpha_1 < \alpha_2 < \ldots < \alpha_K$ for the elements of $A_{u,\tau}$, where $K = |A_{u,\tau}|$. Define the quantities

$$a_{u,\tau,\alpha} := \sum_{i:(e^{-\tau\Delta}u)_i = \alpha} d_i^r. \tag{4.11}$$

Proposition 4.2.7. *If $\tau > 0$, then $0 \in A_{u,\tau} \Rightarrow u = \mathbf{0}$, and $1 \in A_{u,\tau} \Rightarrow u = \mathbf{1}$.*

Proof. Follows immediately from the connected graph case of [9, Lemma 2.6(d)]. □

4

4.2.3. The MBO case

We first solve the case when $\lambda = 1$.

Definition 4.2.8. *Define the set of solutions to (4.5), where $M = \mathcal{M}(u_n)$*

$$S_{\tau,u_n} := \operatorname*{argmax}_{u \in X_M} \left\langle u, e^{-\tau\Delta}u_n \right\rangle_{\mathcal{V}}. \tag{4.12}$$

Note 15. *This is convex as the objective function is linear and X_M is convex, compact as it is a closed subset of X_M, and non-empty as X_M is compact so the continuous objective function attains its maximum value.*

Proposition 4.2.9. *Let $M = \mathcal{M}(u_n)$. Then S_{τ,u_n} is a face of X_M, i.e. if $u, v \in X_M$ and $t \in (0,1)$, then*

$$tu + (1-t)v \in S_{\tau,u_n} \Rightarrow u, v \in S_{\tau,u_n}.$$

Proof. Let $u, v \in X_M$, $t \in (0,1)$, and $tu + (1-t)v \in S_{\tau,u_n}$. Then

$$t\left\langle u, e^{-\tau\Delta}u_n\right\rangle_{\mathcal{V}} + (1-t)\left\langle v, e^{-\tau\Delta}u_n\right\rangle_{\mathcal{V}} = \max_{w \in X_M} \left\langle w, e^{-\tau\Delta}u_n\right\rangle_{\mathcal{V}}$$

and so

$$t\left\langle u, e^{-\tau\Delta}u_n\right\rangle_{\mathcal{V}} \geq \max_{w \in X_M} \left\langle w, e^{-\tau\Delta}u_n\right\rangle_{\mathcal{V}} - (1-t)\max_{w \in X_M} \left\langle w, e^{-\tau\Delta}u_n\right\rangle_{\mathcal{V}} = t\max_{w \in X_M} \left\langle w, e^{-\tau\Delta}u_n\right\rangle_{\mathcal{V}}$$

and likewise for $\left\langle v, e^{-\tau\Delta}u_n\right\rangle_{\mathcal{V}}$. Hence

$$\left\langle u, e^{-\tau\Delta}u_n\right\rangle_{\mathcal{V}} = \left\langle v, e^{-\tau\Delta}u_n\right\rangle_{\mathcal{V}} = \max_{w \in X_M} \left\langle w, e^{-\tau\Delta}u_n\right\rangle_{\mathcal{V}},$$

which is to say that $u, v \in S_{\tau,u_n}$. □

Definition 4.2.10. *Let $\mathcal{A} \subseteq \mathcal{V}$. Then define the* convex hull *of $\mathcal{A}$, written $\operatorname{conv}(\mathcal{A})$, to be the intersection of all convex sets C satisfying $\mathcal{A} \subseteq C \subseteq \mathcal{V}$. Equivalently, the convex hull of $\mathcal{A}$ is the set of all* convex combinations *of points from $\mathcal{A}$, i.e.*[3]

$$\operatorname{conv}(\mathcal{A}) = \left\{ \sum_{k=1}^{n} a_k x_k \,\middle|\, \forall k, x_k \in \mathcal{A} \text{ and } a_k \geq 0, \text{ and } \sum_{k=1}^{n} a_k = 1 \right\}.$$

[3] This equivalence is a standard result. We leave it as an exercise for the reader.

Proposition 4.2.11. *Let $M = \mathcal{M}(u_n)$. Then the extreme points of S_{τ,u_n} are given by*

$$\operatorname{Ext} S_{\tau,u_n} = S_{\tau,u_n} \cap \operatorname{Ext} X_M$$

and the solutions to (4.5) are given by the convex hull of the solutions which lie in $\operatorname{Ext} X_M$, i.e.

$$S_{\tau,u_n} = \operatorname{conv}(S_{\tau,u_n} \cap \operatorname{Ext} X_M).$$

Proof. Let $u \in S_{\tau,u_n} \cap \operatorname{Ext} X_M$, $v_1, v_2 \in S_{\tau,u_n} \subseteq X_M$, $t \in (0,1)$, and $u = tv_1 + (1-t)v_2$. Then $v_1 = v_2$ since $u \in \operatorname{Ext} X_M$. So $u \in \operatorname{Ext} S_{\tau,u_n}$.

Next, let $u \in \operatorname{Ext} S_{\tau,u_n} \subseteq S_{\tau,u_n}$, $v_1, v_2 \in X_M$, and $u = tv_1 + (1-t)v_2$. Then $v_1, v_2 \in S_{\tau,u_n}$ as S_{τ,u_n} is a face, and so $v_1 = v_2$ since $u \in \operatorname{Ext} S_{\tau,u_n}$. Hence $u \in S_{\tau,u_n} \cap \operatorname{Ext} X_M$.

So $\operatorname{Ext} S_{\tau,u_n} = S_{\tau,u_n} \cap \operatorname{Ext} X_M$, and finally we apply the Krein–Milman Theorem (see e.g. [14, p. 75]), which entails in particular that a finite-dimensional compact convex set is the convex hull of its extreme points. □

Corollary 4.2.12. *For $\mathcal{M}(u_0) = M$, there exists a trajectory u_n obeying (4.5) such that*

$$\forall n \in \mathbb{N},\ u_n \in \operatorname{Ext} X_M = \left\{ u \in X_M \,\middle|\, \exists i^* \in V\ \forall j \in V \setminus \{i^*\}\ u_j \in \{0,1\} \right\}.$$

Proof. Follows immediately from the fact that S_{τ,u_n} is non-empty, and so $S_{\tau,u_n} \cap \operatorname{Ext} X_M$ is non-empty as otherwise $S_{\tau,u_n} = \operatorname{conv}(\emptyset) = \emptyset$. □

In [10, §5.3], Van Gennip considered a mass-conserving MBO scheme for minimising the Ohta–Kawasaki functional with a modified graph diffusion, which in the $\gamma = 0$ special case reduces to ordinary graph diffusion and hence is the same problem as (4.5). We here repeat a property he proved for the solutions to (4.5) extreme points.

Theorem 4.2.13. *Let $M = \mathcal{M}(u_n)$, $u_{n+1} \in S_{\tau,u_n} \cap \operatorname{Ext} X_M$, and write*

$$E := \{i \in V \mid (u_{n+1})_i = 1\},\ F := \{i \in V \mid (u_{n+1})_i = 0\}$$

Then for each $i \in V \setminus F$, $j \in V \setminus E$ we have $(e^{-\tau\Delta}u_n)_i \geq (e^{-\tau\Delta}u_n)_j$.

Proof. By Proposition 4.2.5 we have that $u_{n+1} = \chi_E + \theta\chi_{V\setminus(E\cup F)}$ where $\theta \in (0,1)$ and $V \setminus (E \cup F)$ has at most one element which we will denote i^* (when it exists). Now choose some $0 < \delta < \min_{i\in V}\{d_i^r, d_{i^*}^r\theta, d_{i^*}^r(1-\theta)\}$, and any $i \in V\setminus F$, $j \in V\setminus E$. Define

$$u := u_{n+1} - \delta d_i^{-r}\chi_{\{i\}} + \delta d_j^{-r}\chi_{\{j\}}$$

where by choice of δ we have ensured that $u \in X_M$. Therefore

$$0 \leq \langle u_{n+1} - u, e^{-\tau\Delta}u_n\rangle_V = \delta((e^{-\tau\Delta}u_n)_i - (e^{-\tau\Delta}u_n)_j)$$

and so $(e^{-\tau\Delta}u_n)_i \geq (e^{-\tau\Delta}u_n)_j$ as desired. □

4.2.4. Uniqueness conditions for the mass-conserving MBO scheme

We consider when (4.5) has a unique solution, and characterise all solutions to (4.5).

Corollary 4.2.14. *Let $M = \mathcal{M}(u_n)$. Then S_{τ,u_n} has one element if and only if $S_{\tau,u_n} \cap \operatorname{Ext} X_M$ has one element.*

Proof. As S_{τ,u_n} is non-empty, $S_{\tau,u_n} \cap \operatorname{Ext} X_M$ is non-empty as else $S_{\tau,u_n} = \operatorname{conv}(\emptyset) = \emptyset$. Thus, if $S_{\tau,u_n} = \{u\}$ then $S_{\tau,u_n} \cap \operatorname{Ext} X_M = \{u\}$ as this is the only non-empty subset of S_{τ,u_n}. Conversely, if $S_{\tau,u_n} \cap \operatorname{Ext} X_M = \{u\}$ then by Proposition 4.2.11 $S_{\tau,u_n} = \operatorname{conv}(\{u\}) = \{u\}$. □

Usefully, Theorem 4.2.13 gives a necessary condition for $u \in S_{\tau,u_n} \cap \operatorname{Ext} X_M$. We demonstrate the following sufficient condition for uniqueness of solutions.

Theorem 4.2.15. *Define the condition*

$$\forall i, j \in V, \quad i \neq j \Rightarrow (e^{-\tau\Delta}u_n)_i \neq (e^{-\tau\Delta}u_n)_j. \tag{4.13}$$

Then if (4.13) *holds, S_{τ,u_n} has a unique element* (*i.e.* (4.5) *has a unique solution*).

Proof. WLOG, up to a relabelling of V, we may write (4.13) as

$$i < j \Leftrightarrow (e^{-\tau\Delta}u_n)_i < (e^{-\tau\Delta}u_n)_j.$$

Let $M = \mathcal{M}(u_n)$.

Let $u \in S_{\tau,u_n} \cap \operatorname{Ext} X_M$. By Theorem 4.2.13 we thus have

$$i < j \Rightarrow u_i = 0 \text{ or } u_j = 1$$

and hence by Proposition 4.2.5 u must have the form

$$u = \Big(\underbrace{0, 0, \ldots, 0}_{a-1}, \theta, \underbrace{1, 1, \ldots, 1}_{|V|-a} \Big)$$

where $\theta \in (0, 1]$ so (a, θ) uniquely determines any element of $S_{\tau,u_n} \cap \operatorname{Ext} X_M$. Let

$$\mathcal{M}(a, \theta) := \mathcal{M}(u) \text{ for } u \text{ defined by } (a, \theta) \text{ as above.}$$

Then for $a < b$,

$$\mathcal{M}(a, \theta) - \mathcal{M}(b, \phi) = \theta d_a^r + \sum_{a<i<b} d_i^r + (1 - \phi) d_b^r > 0$$

and clearly $\mathcal{M}(a, \theta) = \mathcal{M}(a, \phi)$ if and only if $\theta = \phi$. If $u \in S_{\tau,u_n} \cap \operatorname{Ext} X_M$, then $\mathcal{M}(u) = M$, and by the above we have that $\mathcal{M}(a, \theta) = M$ for a unique (a, θ). Thus $S_{\tau,u_n} \cap \operatorname{Ext} X_M$ has a unique element (as by the proof of Corollary 4.2.12 $S_{\tau,u_n} \cap \operatorname{Ext} X_M$ is non-empty), so by Corollary 4.2.14 S_{τ,u_n} has a unique element. □

Following this idea, we get a characterisation of S_{τ,u_n} and a necessary and sufficient condition for uniqueness.

Theorem 4.2.16. *Suppose $u_n \in \mathcal{V}_{[0,1]}$ and $M = \mathcal{M}(u_n) > 0$, then there is a unique k such that $1 \le k \le K$ and*

$$\sum_{\ell=k+1}^{K} a_{u_n,\tau,\alpha_\ell} < M \le \sum_{\ell=k}^{K} a_{u_n,\tau,\alpha_\ell}$$

recalling K and $a_{u,\tau,\alpha}$ from Definition 4.2.6. Then $u \in S_{\tau,u_n}$ if and only if $u \in X_M$ and

$$u_i = 0, \text{ if } (e^{-\tau\Delta}u_n)_i < \alpha_k, \tag{4.14a}$$

$$u_i = 1, \text{ if } (e^{-\tau\Delta}u_n)_i > \alpha_k, \tag{4.14b}$$

$$M - \sum_{\ell=k+1}^{K} a_{u_n,\tau,\alpha_\ell} = \sum_{(e^{-\tau\Delta}u_n)_i=\alpha_k} d_i^r u_i. \tag{4.14c}$$

Therefore S_{τ,u_n} has a unique element if and only if

$$M = \sum_{\ell=k}^{K} a_{u_n,\tau,\alpha_\ell} \quad \text{or} \quad \exists! i \in V,\ (e^{-\tau\Delta}u_n)_i = \alpha_k. \tag{4.15}$$

Proof. First, we show that k exists and is unique. Let $B_r := \sum_{\ell=r}^{K} a_{u_n,\tau,\alpha_\ell}$. Then as $a_{u_n,\tau,\alpha_\ell} > 0$ the B_r are strictly decreasing in r and we observe that $B_1 = \mathcal{M}(\mathbf{1}) \ge M$ and $B_{K+1} = 0 < M$. Hence there exists a unique $k \in \{1,\dots,K\}$ such that $B_{k+1} < M \le B_k$.

Next, for $v \in \mathcal{V}$, define $\tilde{v} : \{1,\dots,K\} \to \mathbb{R}$ by

$$\tilde{v}_\ell := a_{u_n,\tau,\alpha_\ell}^{-1} \sum_{i:(e^{-\tau\Delta}u_n)_i=\alpha_\ell} d_i^r v_i$$

and define the inner product

$$\langle \tilde{v}, \tilde{w} \rangle_\alpha := \sum_{\ell=1}^{K} a_{u_n,\tau,\alpha_\ell} \tilde{v}_\ell \tilde{w}_\ell.$$

Then note by a simple calculation we have that

$$\langle \tilde{v}, \mathbf{1} \rangle_\alpha = \mathcal{M}(v)$$

and

$$\langle \tilde{v}, \widetilde{e^{-\tau\Delta}u_n} \rangle_\alpha = \langle v, e^{-\tau\Delta}u_n \rangle_\mathcal{V}.$$

Hence, defining $\tilde{X}_M = \{\tilde{v} | v \in X_M\}$, we have that $u \in S_{\tau,u_n}$ if and only if

$$\tilde{u} \in \operatorname*{argmax}_{\tilde{v}\in\tilde{X}_M} \left\langle \tilde{v}, \widetilde{e^{-\tau\Delta}u_n} \right\rangle_\alpha$$

and note that (4.13) is satisfied by $\widetilde{e^{-\tau\Delta}u_n}$ (i.e. $(\widetilde{e^{-\tau\Delta}u_n})_\ell \neq (\widetilde{e^{-\tau\Delta}u_n})_r$ for all $\ell \neq r \in \{1,2,...,K\}$). Therefore by the same argument as in the proof of the previous theorem *mutatis mutandis* (i.e. replacing instances of $\langle\cdot,\cdot\rangle_v$ with $\langle\cdot,\cdot\rangle_\alpha$, of d_i^r with a_{u_n,τ,α_ℓ} etc.) there is a unique such $\tilde{u}$ of the form

$$\tilde{u} = \Big(\underbrace{0,0,...,0}_{b-1}, \theta, \underbrace{1,1,...,1}_{K-b} \Big)$$

4

where $\theta \in (0,1]$. Then we have

$$M = \langle \tilde{u}, \mathbf{1} \rangle_\alpha = \theta a_{u_n,\tau,\alpha_b} + \sum_{\ell=b+1}^{K} a_{u_n,\tau,\alpha_\ell}$$

so we must have $b = k$ and

$$\theta = a^{-1}_{u_n,\tau,\alpha_k} \left(M - \sum_{\ell=k+1}^{K} a_{u_n,\tau,\alpha_\ell} \right).$$

Taking $\ell < k$,

$$0 = \tilde{u}_\ell = a^{-1}_{u_n,\tau,\alpha_\ell} \sum_{i:(e^{-\tau\Delta}u_n)_i=\alpha_\ell} d_i^r u_i$$

and so $u_i = 0$ if $(e^{-\tau\Delta}u_n)_i < \alpha_k$, and taking $\ell > k$

$$1 = \tilde{u}_\ell = a^{-1}_{u_n,\tau,\alpha_\ell} \sum_{i:(e^{-\tau\Delta}u_n)_i=\alpha_\ell} d_i^r u_i$$

and so $u_i = 1$ if $(e^{-\tau\Delta}u_n)_i > \alpha_k$. Finally taking $\ell = k$ we get the equivalences

$$u \in S_{\tau,u_n} \text{ if and only if } \tilde{u} \in \operatorname*{argmax}_{\tilde{v}\in\tilde{X}_M} \left\langle \tilde{v}, \widetilde{e^{-\tau\Delta}u_n} \right\rangle_\alpha$$

$$\text{if and only if } \begin{cases} u_i = 0, \text{ if } (e^{-\tau\Delta}u_n)_i < \alpha_k, \\ u_i = 1, \text{ if } (e^{-\tau\Delta}u_n)_i > \alpha_k, \\ \theta = a^{-1}_{u_n,\tau,\alpha_k} \sum_{(e^{-\tau\Delta}u_n)_i=\alpha_k} d_i^r u_i. \end{cases}$$

Hence we have a unique solution if and only if $(e^{-\tau\Delta}u_n)_i = \alpha_k$ at a unique $i \in V$ or $\theta = 1$ (and therefore $u_i = 1$ for $(e^{-\tau\Delta}u_n)_i = \alpha_k$), i.e. when (4.15) holds. □

Note 16. *If $M = 0$ then $X_M = \{\mathbf{0}\}$, so uniqueness is trivial, hence supposing that $M > 0$ incurs no loss of generality.*

Note 17. *The solution in (4.14), with an adjustible threshold level (i.e. α_k) to ensure that mass is conserved, accords with the definition of the mass-conserving graph MBO scheme in Van Gennip [10] and with the definition of the mass-conserving continuum MBO scheme in Ruuth and Wetton [16]. We here note that there is a typo in the definition in [10] (i.e., [10, Algorithm (mcOKMBO)]): all instances of $d_i^r u_i$ in that definition should just read d_i^r.*

4.2.5. The non-MBO case

We now solve the case for $0 \leq \lambda < 1$. To solve (4.4) in this case, we use duality. For a detailed description of this framework, we refer the reader to e.g. [2, §5]. Let $M := \mathcal{M}(u_n)$ and define the functions

$$f_i(u) := -d_i^r u_i, \qquad g_i(u) := (u_i - 1)d_i^r, \qquad h(u) := 2(\mathcal{M}(u) - M). \tag{4.16}$$

Then (4.4) can be written as the primal problem:

$$\min_{u \in \mathcal{V}} (1-\lambda)\,||u||_{\mathcal{V}}^2 - 2\langle u, e^{-\tau\Delta}u_n\rangle_{\mathcal{V}} \text{ s.t. } \forall i \in V,\ f_i(u) \leq 0,\ g_i(u) \leq 0, \text{ and } h(u) = 0.$$

Hence for $\xi, \mu \in \mathcal{V}$ and $\nu \in \mathbb{R}$ dual variables, (4.4) has Lagrangian:

$$\begin{aligned} L(u,\xi,\mu,\nu) &:= (1-\lambda)\,||u||_{\mathcal{V}}^2 - 2\langle u, e^{-\tau\Delta}u_n\rangle_{\mathcal{V}} + \sum_{i\in V}(\xi_i f_i(u) + \mu_i g_i(u)) + \nu h(u) \\ &= (1-\lambda)\,||u||_{\mathcal{V}}^2 - 2\langle u, e^{-\tau\Delta}u_n\rangle_{\mathcal{V}} + \langle u, \mu - \xi\rangle_{\mathcal{V}} + \langle 2\nu u - \mu, \mathbf{1}\rangle_{\mathcal{V}} - 2\nu M. \end{aligned} \tag{4.17}$$

We can rewrite this by making the following definition:

$$u^*(\xi,\mu,\nu) := \frac{1}{2(1-\lambda)}\left(2e^{-\tau\Delta}u_n + \xi - \mu - 2\nu\mathbf{1}\right) \tag{4.18}$$

so that

$$\begin{aligned} L(u,\xi,\mu,\nu) &= (1-\lambda)\,||u||_{\mathcal{V}}^2 - 2(1-\lambda)\,\langle u, u^*(\xi,\mu,\nu)\rangle_{\mathcal{V}} - \langle \mu, \mathbf{1}\rangle_{\mathcal{V}} - 2\nu M \\ &= (1-\lambda)\left|\left|u - u^*(\xi,\mu,\nu)\right|\right|_{\mathcal{V}}^2 - (1-\lambda)||u^*(\xi,\mu,\nu)||_{\mathcal{V}}^2 - \langle \mu, \mathbf{1}\rangle_{\mathcal{V}} - 2\nu M \end{aligned}$$

which we note is strictly convex, proper, and bounded below in u (for fixed ξ, μ, and ν). Next, we define the dual objective function:

$$G(\xi,\mu,\nu) := \inf_{u\in\mathcal{V}} L(u,\xi,\mu,\nu) = L(u^*(\xi,\mu,\nu),\xi,\mu,\nu). \tag{4.19}$$

and therefore

$$G(\xi,\mu,\nu) = -\left((1-\lambda)\left|\left|u^*(\xi,\mu,\nu)\right|\right|_{\mathcal{V}}^2 + \langle \mu, \mathbf{1}\rangle_{\mathcal{V}} + 2\nu M\right). \tag{4.20}$$

The dual problem to (4.4) is given by

$$\sup_{\xi\geq 0, \mu \geq 0, \nu} G(\xi,\mu,\nu). \tag{4.21}$$

Lemma 4.2.17. *For $u_n \in \mathcal{V}_{[0,1]}$, $M = \mathcal{M}(u_n)$, (4.4) and (4.21) have strong duality, i.e.*

$$\sup_{\xi \geq 0, \mu \geq 0, \nu} G(\xi, \mu, \nu) = \min_{u \in X_M} (1-\lambda)\,||u||_{\mathcal{V}}^2 - 2\langle u, e^{-\tau\Delta} u_n \rangle_{\mathcal{V}}$$

and if ξ^*, μ^*, and ν^* optimise (4.21), then $u^* := u^*(\xi^*, \mu^*, \nu^*) \in X_M$ as in (4.18) solves (4.4).

Proof. We apply Slater's condition for strong duality from [2, §5.2.3] that states in particular that if the primal problem is convex, the domain of the problem is open and affine, the problem has affine constraints, and the set of feasible solutions to the problem is non-empty, then we have strong duality, i.e. the minimum value of the primal problem equals the maximum value of the dual problem. As the f_i and g_i are affine on $\mathcal{V}$ and $\mathcal{V}$ is open and affine, Slater's condition is satisfied if $\exists u \in \mathcal{V}$ with $f_i(u) \leq 0, g_i(u) \leq 0$, and $h(u) = 0$, i.e. if $\exists u \in X_M$. As $u_n \in X_M$ we thus have strong duality.

Now let $\xi^* \geq 0, \mu^* \geq 0$, and ν^* be optimal for (4.21), and let $\hat{u} \in X_M$ be optimal for (4.4), which we know exists since X_M is compact and the objective function is continuous. Writing $q(u) := (1-\lambda)\,||u||_{\mathcal{V}}^2 - 2\langle u, e^{-\tau\Delta} u_n \rangle_{\mathcal{V}}$ we have by strong duality:

$$q(\hat{u}) = G(\xi^*, \mu^*, \nu^*) = L(u^*, \xi^*, \mu^*, \nu^*) = \inf_{u \in \mathcal{V}} L(u, \xi^*, \mu^*, \nu^*) \leq L(\hat{u}, \xi^*, \mu^*, \nu^*) \leq q(\hat{u})$$

where the final inequality holds by (4.17), as $\hat{u} \in X_M$ and so $f_i(\hat{u}), g_i(\hat{u}) \leq 0$ and $h(\hat{u}) = 0$. So the inequalities are equalities and $L(u, \xi^*, \mu^*, \nu^*)$ is minimised at $\hat{u}$. As L is strictly convex in u it has a unique minimiser, so $u^*(\xi^*, \mu^*, \nu^*) = \hat{u}$ is optimal for (4.4). □

By Lemma 4.2.17 we have that $u^* := u^*(\xi^*, \mu^*, \nu^*) \in X_M$ for (ξ^*, μ^*, ν^*) dual optimal, and by applying complementary slackness (see [2, §5.5.2]) we have that

$$\begin{aligned} u_i^* > 0 &\Rightarrow \xi_i^* = 0, \quad \text{and} \quad & u_i^* < 1 &\Rightarrow \mu_i^* = 0, \\ \xi_i^* > 0 &\Rightarrow u_i^* = 0, \quad \text{and} \quad & \mu_i^* > 0 &\Rightarrow u_i^* = 1. \end{aligned}$$

Thus at each $i \in V$, $\xi_i^* = 0$ or $\mu_i^* = 0$. So we have the necessary conditions

$$u_i^* = \begin{cases} 0 & \Rightarrow \ \mu_i^* = 0, \\ \in (0,1) & \Rightarrow \ \xi_i^* = \mu_i^* = 0, \\ 1 & \Rightarrow \ \xi_i^* = 0. \end{cases}$$

Then by substituting into (4.18)

$$u_i^* = \begin{cases} 0, & \text{if and only if } \ \mu_i^* = 0, \ \xi_i^* = 2\nu^* - 2(e^{-\tau\Delta} u_n)_i \geq 0, \\ \dfrac{(e^{-\tau\Delta} u_n)_i - \nu^*}{1-\lambda} \in (0,1), & \text{if and only if } \ \xi_i^* = \mu_i^* = 0, \ 0 < (e^{-\tau\Delta} u_n)_i - \nu^* < 1 - \lambda, \\ 1, & \text{if and only if } \ \xi_i^* = 0, \ \mu_i^* = 2(e^{-\tau\Delta} u_n)_i - 2(1-\lambda) - 2\nu^* \geq 0. \end{cases}$$

We simplify by noting that the ν^* inequality conditions are disjoint and exhaustive, so we need only consider those conditions (to see this, note that if for example $\nu^* \geq (e^{-\tau\Delta}u_n)_i$ then each of the $u_i^* > 0$ cases are ruled out, so u_i^* must equal zero):

$$u_i^* = \begin{cases} 0, & \text{if and only if } \nu^* - (e^{-\tau\Delta}u_n)_i \geq 0, \\ \frac{(e^{-\tau\Delta}u_n)_i - \nu^*}{1-\lambda} \in (0,1), & \text{if and only if } 0 < (e^{-\tau\Delta}u_n)_i - \nu^* < 1-\lambda, \\ 1, & \text{if and only if } \nu^* \leq (e^{-\tau\Delta}u_n)_i - (1-\lambda). \end{cases} \tag{4.22}$$

But by Lemma 4.2.17, $u^* \in X_M$, so we have $\mathcal{M}(u^*) = M$. Thus $\nu = \nu^*$ is a solution to:

$$0 = M + \sum_{i \in V} d_i^r \begin{cases} -1, & \nu \leq (e^{-\tau\Delta}u_n)_i - (1-\lambda), \\ \frac{\nu - (e^{-\tau\Delta}u_n)_i}{1-\lambda}, & (e^{-\tau\Delta}u_n)_i - (1-\lambda) < \nu < (e^{-\tau\Delta}u_n)_i, \\ 0, & \nu \geq (e^{-\tau\Delta}u_n)_i, \end{cases} \tag{4.23}$$

which exists by the Intermediate Value Theorem (since the RHS of (4.23) is a sum of continuous functions in ν). By Definition 4.2.6, we rewrite (4.23)

$$M = \sum_{\alpha \in A_{u_n,\tau}} a_{u_n,\tau,\alpha} \begin{cases} 1, & \nu \leq \alpha - (1-\lambda), \\ \frac{\alpha - \nu}{1-\lambda}, & \alpha - (1-\lambda) < \nu < \alpha, \\ 0, & \nu \geq \alpha. \end{cases} \tag{4.24}$$

Note 18. *Although u^* is unique, ν^* is not in general unique, but in such cases each solution ν^* gives rise to the same u^*. For example if $A_{u_n,\tau} = \{0\}$ (i.e. $u_n = \mathbf{0}$) then any $\nu \geq 0$ solves (4.24), but by the same token in that case any $\nu \geq 0$ gives $u^* = \mathbf{0}$. In general, the right hand side of (4.24) is constant in ν if and only if $\nu \in [\alpha_k, \alpha_{k+1} - (1-\lambda)]$, where α_k, α_{k+1} are consecutive elements in $A_{u_n,\tau}$. But if ν in that interval solves (4.24), then by (4.22)*

$$u_i^* = \begin{cases} 0, & \text{if and only if } (e^{-\tau\Delta}u_n)_i \leq \alpha_k, \\ 1, & \text{if and only if } (e^{-\tau\Delta}u_n)_i > \alpha_k. \end{cases}$$

Finally, note that therefore this situation of non-unique ν^* can only arise if $M \in \{\mathcal{M}(u) \mid u \in \mathcal{V}_{\{0,1\}}\}$, which is a finite set of values.

Proposition 4.2.18. *Let $u_n \in \mathcal{V}_{[0,1]}$, $M = \mathcal{M}(u_n)$, and suppose $0 < M < \langle \mathbf{1}, \mathbf{1} \rangle_\mathcal{V}$ and $\tau > 0$. If ν solves (4.24), then $\nu \in [\lambda \min A_{u_n,\tau}, \lambda \max A_{u_n,\tau}] \subseteq (0, \lambda)$.*

Proof. By Proposition 4.2.7 and the condition on M, note that $A_{u_n,\tau} \subseteq (0,1)$. Since diffusion preserves mass, $M = \mathcal{M}(e^{-\tau\Delta}u_n)$ and therefore

$$M = \sum_{\alpha \in A_{u_n,\tau}} a_{u_n,\tau,\alpha}\alpha$$

and so we have by (4.24):

$$0 = \sum_{\alpha\in A_{u_n,\tau}} a_{u_n,\tau,\alpha} \begin{cases} 1-\alpha, & \nu \le \alpha - (1-\lambda), \\ \frac{\alpha-\nu}{1-\lambda} - \alpha, & \alpha-(1-\lambda) < \nu < \alpha, \\ -\alpha, & \nu \ge \alpha, \end{cases} \tag{4.25}$$

i.e., ν is a solution to

$$0 = \sum_{\alpha\in[1-\lambda+\nu,1)\cap A_{u_n,\tau}} a_{u_n,\tau,\alpha}(1-\alpha) + \sum_{\alpha\in(\nu,1-\lambda+\nu)\cap A_{u_n,\tau}} a_{u_n,\tau,\alpha}\frac{\alpha\lambda-\nu}{1-\lambda} + \sum_{\alpha\in(0,\nu]\cap A_{u_n,\tau}} a_{u_n,\tau,\alpha}(-\alpha).$$

First, suppose that $\nu < \lambda \min A_{u_n,\tau} < \min A_{u_n,\tau}$ (recall that we are still in the $\lambda \in [0,1)$ case). Then

$$\sum_{\alpha\in(0,\nu]\cap A_{u_n,\tau}} a_{u_n,\tau,\alpha}(-\alpha) = 0$$

and $\alpha\lambda - \nu > \lambda(\alpha - \min A_{u_n,\tau}) \ge 0$ for $\alpha \in A_{u_n,\tau}$ so

$$\sum_{\alpha\in[1-\lambda+\nu,1)\cap A_{u_n,\tau}} a_{u_n,\tau,\alpha}(1-\alpha) + \sum_{\alpha\in(\nu,1-\lambda+\nu)\cap A_{u_n,\tau}} a_{u_n,\tau,\alpha}\frac{\alpha\lambda-\nu}{1-\lambda} > 0$$

hence ν does not solve (4.25). (Note that at least one of these sums is non-empty due to our supposition on ν.)

Next, suppose that $\nu > \lambda \max A_{u_n,\tau}$. Then we have $\max A_{u_n,\tau} = (1-\lambda)\max A_{u_n,\tau} + \lambda \max A_{u_n,\tau} < 1-\lambda+\nu$ so

$$\sum_{\alpha\in[1-\lambda+\nu,1)\cap A_{u_n,\tau}} a_{u_n,\tau,\alpha}(1-\alpha) = 0$$

and $\alpha\lambda - \nu < \lambda(\alpha - \max A_{u_n,\tau}) \le 0$ for $\alpha \in A_{u_n,\tau}$ so

$$\sum_{\alpha\in(\nu,1-\lambda+\nu)\cap A_{u_n,\tau}} a_{u_n,\tau,\alpha}\frac{\alpha\lambda-\nu}{1-\lambda} + \sum_{\alpha\in(0,\nu]\cap A_{u_n,\tau}} a_{u_n,\tau,\alpha}(-\alpha) < 0.$$

Thus if ν solves (4.25) we must have $\nu \in [\lambda \min A_{u_n,\tau}, \lambda \max A_{u_n,\tau}]$. □

Note 19. *If $M = 0$ then $u^* = \mathbf{0} = u_n$, which is satisfied if and only if for all $i \in V$, $\nu \ge (e^{-\tau\Delta}u_n)_i = 0$. If $M = \langle \mathbf{1}, \mathbf{1}\rangle_V$ then $u^* = \mathbf{1} = u_n$, which is satisfied if and only if for all $i \in V$, $\nu \le (e^{-\tau\Delta}u_n)_i - 1 + \lambda = \lambda$. Hence we can always assume ν to lie in $[0,\lambda]$.*

In summary, we have the following theorem.

Theorem 4.2.19. *For $0 \leq \lambda < 1$, (4.4) has a unique solution*

$$(u_{n+1}^{\lambda})_i = \begin{cases} 0, & \text{if and only if } \nu \geq (e^{-\tau\Delta}u_n)_i, \\ \frac{(e^{-\tau\Delta}u_n)_i - \nu}{1-\lambda}, & \text{if and only if } (e^{-\tau\Delta}u_n)_i - (1-\lambda) < \nu < (e^{-\tau\Delta}u_n)_i, \\ 1, & \text{if and only if } \nu \leq (e^{-\tau\Delta}u_n)_i - (1-\lambda), \end{cases} \tag{4.26}$$

where ν is a solution to (4.24) and hence $\nu \in [0, \lambda]$.

4.2.6. The converse of Theorem 4.1.4 in the mass-conserving case

In this section we prove the following theorem.

Theorem 4.2.20. *If $u = u_{n+1}$ solves (4.4), then $\exists \beta \in \mathcal{B}(u)$ (given by (4.28) when $\lambda = 1$ and (4.30) when $0 \leq \lambda < 1$), such that (u, β) is a solution to (4.2) (for β as β_{n+1}).*

Note 20. *If (u, β) and (u, β') solve (4.2) then rearranging we get*

$$\beta - \beta' = \bar{\beta}\mathbf{1} - \bar{\beta}'\mathbf{1}$$

i.e. β and β' differ only by a multiple of $\mathbf{1}$. So, for a given u and $\beta \in \mathcal{B}(u)$, (u, β) is a solution if and only if (u, β') is a solution for all and only the $\beta' \in \{\beta + \theta\mathbf{1} \mid \theta \in \mathbb{R}\} \cap \mathcal{B}(u)$. If $u_i \in (0,1)$ for an $i \in V$ and (u, β) and (u, β') solve (4.2), then $\beta = \beta'$ as $\beta_i = \beta_i' = 0$ by the definition of $\mathcal{B}(u)$ (3.21).

Our proof of this theorem will split into two cases, $\lambda = 1$ and $\lambda \in [0, 1)$. In each case, we shall first engage in some preliminary work to discover a candidate β, and then we will prove that this choice of β suffices to prove the theorem.

Case: $\lambda = 1$

If $M = 0$ then $u = u_n = \mathbf{0}$ is trivially a solution to (4.2), for example taking $\beta = \mathbf{0}$, and hence WLOG we can suppose $M = \mathcal{M}(u_n) > 0$. Let k be as in Theorem 4.2.16, such that

$$\sum_{\ell=k+1}^{K} a_{u_n,\tau,\alpha_\ell} < M \leq \sum_{\ell=k}^{K} a_{u_n,\tau,\alpha_\ell}.$$

Then, recalling Theorem 4.2.16, any solution u to (4.4) for $\lambda = 1$ must satisfy

$$u_i = 0, \text{ if } (e^{-\tau\Delta}u_n)_i < \alpha_k,$$
$$u_i = 1, \text{ if } (e^{-\tau\Delta}u_n)_i > \alpha_k,$$
$$M - \sum_{\ell=k+1}^{K} a_{u_n,\tau,\alpha_\ell} = \sum_{(e^{-\tau\Delta}u_n)_i = \alpha_k} d_i^r u_i.$$

For $\lambda = 1$, (4.2) becomes

$$-e^{-\tau\Delta}u_n + \frac{M}{\langle \mathbf{1}, \mathbf{1} \rangle_{\mathcal{V}}}\mathbf{1} = \beta - \bar{\beta}\mathbf{1}. \tag{4.27}$$

We seek to find a β such that $\beta_i = 0$ if $u_i \in (0,1)$. Note that if $u_i \in (0,1)$, then by Theorem 4.2.16 we have $(e^{-\tau\Delta}u_n)_i = \alpha_k$, so we desire to have

$$-\alpha_k + \frac{M}{\langle \mathbf{1}, \mathbf{1}\rangle_\nu} = -\bar{\beta}.$$

Therefore substituting into (4.27) we have candidate solution:

$$\beta = \alpha_k \mathbf{1} - e^{-\tau\Delta}u_n. \tag{4.28}$$

We now verify that this candidate solution works even for binary u.

Proof of Theorem 4.2.20 for $\lambda = 1$. We check that the β from (4.28) solves (4.27):

$$-e^{-\tau\Delta}u_n + \frac{M}{\langle \mathbf{1}, \mathbf{1}\rangle_\nu}\mathbf{1} = \alpha_k \mathbf{1} - e^{-\tau\Delta}u_n - \alpha_k \mathbf{1} + \overline{e^{-\tau\Delta}u_n}\mathbf{1}.$$

Moreover, by the form for u from Theorem 4.2.16 it follows that $\beta \in \mathcal{B}(u)$. □

Case: $0 \le \lambda < 1$

For $0 \le \lambda < 1$, (4.4) is strictly convex, so recalling (4.26) it has unique solution

$$u_i = \begin{cases} 0, & \text{if and only if } \nu \ge (e^{-\tau\Delta}u_n)_i, \\ \frac{(e^{-\tau\Delta}u_n)_i - \nu}{1-\lambda}, & \text{if and only if } (e^{-\tau\Delta}u_n)_i - (1-\lambda) < \nu < (e^{-\tau\Delta}u_n)_i, \\ 1, & \text{if and only if } \nu \le (e^{-\tau\Delta}u_n)_i - (1-\lambda), \end{cases}$$

where $\nu \in [0, \lambda]$ solving (4.24) is such that $\bar{u} = \overline{u_n}$.

Hence (4.2) is satisfied if and only if for all $i \in V$

$$\lambda\beta_i - \lambda\bar{\beta} = \lambda\bar{u} + \begin{cases} -(e^{-\tau\Delta}u_n)_i, & \text{if } \nu \ge (e^{-\tau\Delta}u_n)_i, \\ -\nu, & \text{if } (e^{-\tau\Delta}u_n)_i - (1-\lambda) < \nu < (e^{-\tau\Delta}u_n)_i, \\ 1-\lambda-(e^{-\tau\Delta}u_n)_i, & \text{if } \nu \le (e^{-\tau\Delta}u_n)_i - (1-\lambda). \end{cases} \tag{4.29}$$

We seek a β solving this with $\beta_i = 0$ if $u_i \in (0,1)$. Suppose $\exists i \in V$ for which $u_i \in (0,1)$. This occurs when $(e^{-\tau\Delta}u_n)_i - (1-\lambda) < \nu < (e^{-\tau\Delta}u_n)_i$, and so at this i:

$$-\lambda\bar{\beta} = \lambda\bar{u} - \nu.$$

Plugging into (4.29) we get the candidate solution:

$$\beta_i = \lambda^{-1}\begin{cases} \nu - (e^{-\tau\Delta}u_n)_i, & \text{if } \nu \ge (e^{-\tau\Delta}u_n)_i, \\ 0, & \text{if } (e^{-\tau\Delta}u_n)_i - (1-\lambda) < \nu < (e^{-\tau\Delta}u_n)_i, \\ \nu - (e^{-\tau\Delta}u_n)_i + 1 - \lambda, & \text{if } \nu \le (e^{-\tau\Delta}u_n)_i - (1-\lambda), \end{cases} \tag{4.30}$$

which obeys $\beta \in \mathcal{B}(u)$ since u obeys (4.26).

Proof of Theorem 4.2.20 for $0 \leq \lambda < 1$. If $\lambda = 0$ (and hence $\tau = 0$), then $u = u_n$ and $v = 0$. We will therefore define the solution to (4.30) in this case to be $\beta = \mathbf{0}$. Note that $\mathbf{0} \in \mathcal{B}(u)$, and that $(u, \mathbf{0})$ solves (4.2) for $\lambda = 0$. It therefore remains to prove the $\lambda \in (0,1)$ case.

By the above discussion, taking u as in (4.26) and β as in (4.30) entails that (u, β) is a solution to (4.2) *if* $\exists i \in V$ with $u_i \in (0,1)$. We check the alternative case, i.e. for all $i \in V$, $u_i \in \{0,1\}$. Take β as in (4.30). As u is binary, either $v \geq (e^{-\tau\Delta}u_n)_i$ or $v \leq (e^{-\tau\Delta}u_n)_i - (1-\lambda)$ at each $i \in V$, so

$$\lambda\beta = v\mathbf{1} - e^{-\tau\Delta}u_n + (1-\lambda)\chi_{\{i|(e^{-\tau\Delta}u_n)_i \geq v+1-\lambda\}}.$$

But as u is binary we have $u = \chi_{\{i|(e^{-\tau\Delta}u_n)_i \geq v+1-\lambda\}}$ and so by (4.30)

$$\lambda\bar{\beta} = v - \bar{u} + (1-\lambda)\bar{u} = v - \lambda\bar{u}.$$

Thus β solves (4.29). Therefore (u, β) is always a solution to (4.2). □

4.3. Properties of the SDIE schemes

4.3.1. The $\lambda \uparrow 1$ limit

For $\lambda < 1$ (4.3) and (4.4) are strictly convex, so have unique solution u_{n+1}^{λ}. In this section we show that as $\lambda \uparrow 1$ these solutions converge pointwise to a solution of the $\lambda = 1$ case, yielding a choice function for the MBO solutions. Recalling that $\lambda := \tau/\varepsilon$, we will make precise the way that we are taking $\lambda \uparrow 1$, when relevant.

As a prelude to investigating the convergence properties of u_{n+1}^{λ}, we first show that convergence of solutions as $\lambda \uparrow 1$ is relevant to solving the $\lambda = 1$ case.

Theorem 4.3.1. *Fix* u_n, *denote the objective function in* (4.3) *or* (4.4) *by*

$$q_{\tau,\varepsilon} : u \mapsto \frac{\tau}{\varepsilon}\langle u, \mathbf{1} - u\rangle_V + \left|\left|u - z(\tau)\right|\right|_V^2$$

where $z(\tau) := \mathcal{S}_\tau u_n$ *or* $z(\tau) := e^{-\tau\Delta}u_n$ *respectively, and let* $X := \mathcal{V}_{[0,1]}$ *or* $X := \mathcal{V}_{[0,1]} \cap S_M$ *respectively, where* $M := \mathcal{M}(u_n)$. *Furthermore, let* τ_n *and* ε_n *obey:* $0 < \tau_n < \varepsilon_n$ *for all* n, $\tau_n/\varepsilon_n \uparrow 1$, *and* $\tau_n, \varepsilon_n \to \ell$.

Then $q_{\tau_n,\varepsilon_n} \to q_{\ell,\ell}$ *uniformly on* X. *Furthermore, if* $(u^{\tau_n,\varepsilon_n}) \in X$ *solves* (4.3) (*respectively* (4.4)) *with* $\tau = \tau_n$ *and* $\varepsilon = \varepsilon_n$, *and* $u^{\tau_n,\varepsilon_n} \to u$, *then* $u \in X$ *is a solution to* (4.3) (*respectively* (4.4)) *with* $\tau = \varepsilon = \ell$.

Proof. Let $\lambda_n := \tau_n/\varepsilon_n$. Then for any $u \in X$ and $n \in \mathbb{N}$,

$$\begin{aligned}
|q_{\tau_n,\varepsilon_n}(u) - q_{\ell,\ell}(u)| &\leq (1-\lambda_n)\langle u, \mathbf{1} - u\rangle_V + |\langle z(\tau_n) - z(\ell), z(\tau_n) + z(\ell) - 2u\rangle_V| \\
&\leq (1-\lambda_n)\frac{1}{4}||\mathbf{1}||_V^2 + \left|\left|z(\tau_n) - z(\ell)\right|\right|_V \left|\left|z(\tau_n) + z(\ell) - 2u\right|\right|_V \\
&\leq (1-\lambda_n)\frac{1}{4}||\mathbf{1}||_V^2 + 4\left|\left|z(\tau_n) - z(\ell)\right|\right|_V ||\mathbf{1}||_V
\end{aligned}$$

(with the final inequality since $u, z(\tau_n), z(\ell) \in \mathcal{V}_{[0,1]}$) which tends to zero uniformly as $n \to \infty$.

Next, suppose $u^{\tau_n,\varepsilon_n} \to u$ is as in the second part of the statement of the theorem. Then $u \in X$ since X is closed. By uniform convergence, for all $\delta > 0$ we have some N such that for all $n > N$ and all $v \in X$

$$|q_{\tau_n,\varepsilon_n}(v) - q_{\ell,\ell}(v)| \leq \delta/2.$$

Therefore since the u^{τ_n,ε_n} minimise q_{τ_n,ε_n}, for any $v \in X$ we have for all $n > N$

$$q_{\ell,\ell}(u^{\tau_n,\varepsilon_n}) - \delta/2 \leq q_{\tau_n,\varepsilon_n}(u^{\tau_n,\varepsilon_n}) \leq q_{\tau_n,\varepsilon_n}(v) \leq q_{\ell,\ell}(v) + \delta/2.$$

Since $q_{\ell,\ell}$ is continuous we can take $n \to \infty$ and rearrange to get

$$q_{\ell,\ell}(u) \leq q_{\ell,\ell}(v) + \delta$$

and since δ was arbitrary we must have that u is a minimiser of $q_{\ell,\ell}$. □

Theorem 4.3.2. *Let $\tau \geq 0$ be fixed, and so any limit as $\lambda \uparrow 1$ corresponds to a limit as $\varepsilon \downarrow \tau$.*

If u^{λ}_{n+1} solves (4.3), then for some sufficiently small $\delta > 0$, depending only on $\mathcal{S}_\tau u_n$, and each $\lambda \in (1-\delta, 1)$,

$$(u^{\lambda}_{n+1})_i = \begin{cases} 0, & \text{if and only if } (\mathcal{S}_\tau u_n)_i < \frac{1}{2}, \\ \frac{1}{2}, & \text{if and only if } (\mathcal{S}_\tau u_n)_i = \frac{1}{2}, \\ 1, & \text{if and only if } (\mathcal{S}_\tau u_n)_i > \frac{1}{2}, \end{cases}$$

and thus u^{λ}_{n+1} converges to a solution of (4.7) as $\lambda \uparrow 1$.

Next let u^{λ}_{n+1} solve (4.4). If $u_n \in \{\mathbf{0}, \mathbf{1}\}$ then $u^{\lambda}_{n+1} = u_n$ for all $\lambda \in [0,1)$, and thus converges to u_n. If $u_n \in \mathcal{V}_{[0,1]} \setminus \{\mathbf{0}, \mathbf{1}\}$, then $M = \mathcal{M}(u_n) \in (0, \mathcal{M}(\mathbf{1}))$, and so there exists k as in Theorem 4.2.16 with

$$\sum_{l=k+1}^{K} a_{u_n,\tau,\alpha_l} < M \leq \sum_{l=k}^{K} a_{u_n,\tau,\alpha_l}. \tag{4.31}$$

Then for some sufficiently small $\delta > 0$, depending only on $e^{-\tau\Delta}u_n$, and each $\lambda \in (1-\delta, 1)$

$$(u^{\lambda}_{n+1})_i = \begin{cases} 0, & \text{if and only if } (e^{-\tau\Delta}u_n)_i \leq \alpha_{k-1}, \\ a^{-1}_{u_n,\tau,\alpha_k}\left(M - \sum_{\ell=k+1}^{K} a_{u_n,\tau,\alpha_\ell}\right), & \text{if and only if } (e^{-\tau\Delta}u_n)_i = \alpha_k, \\ 1, & \text{if and only if } (e^{-\tau\Delta}u_n)_i \geq \alpha_{k+1}, \end{cases} \tag{4.32}$$

and thus u^{λ}_{n+1} converges to the RHS of (4.32) as $\lambda \uparrow 1$.

Proof. For the fidelity forced case (4.3), since $\{(\mathcal{S}_\tau u_n)_i \mid i \in V\}$ is a finite set there exists $\delta > 0$ such that $\{(\mathcal{S}_\tau u_n)_i \mid i \in V\} \subseteq [0, \frac{1}{2} - \frac{1}{2}\delta) \cup \{\frac{1}{2}\} \cup (\frac{1}{2} + \frac{1}{2}\delta, 1]$. Considering $\lambda \in (1-\delta, 1)$, the result then follows immediately from Theorem 4.2.1.

For the mass-conserving case (4.4), only the $u_n \in \mathcal{V}_{[0,1]} \setminus \{\mathbf{0}, \mathbf{1}\}$ case is non-trivial. As $A_{u_n,\tau}$ is a finite set, we can take $\delta > 0$ sufficiently small so that the open δ-balls around the $\alpha \in A_{u_n,\tau}$ are disjoint. Let $\lambda \in (1-\delta, 1)$ and choose ν solving (4.24). Then by Proposition 4.2.18, $\nu \in (0, \lambda)$, and recall that (4.24) states that

$$M = \sum_{\alpha \in A_{u_n,\tau}} a_{u_n,\tau,\alpha} \begin{cases} 1, & \nu \le \alpha - (1-\lambda), \\ \frac{\alpha - \nu}{1-\lambda}, & \alpha - (1-\lambda) < \nu < \alpha, \\ 0, & \nu \ge \alpha, \end{cases}$$

and by choice of δ, ν is within $1-\lambda$ of at most one α, since $0 < 1-\lambda < \delta$. Let $\alpha_0 := 0$ and $\alpha_{K+1} := 1$. Then there exists $1 \le m \le K$ such that $\nu \in (\alpha_{m-1}, \alpha_{m+1} - (1-\lambda))$, since these intervals cover $(0, \lambda)$, and we have

$$M = \sum_{\ell=m+1}^{K} a_{u_n,\tau,\alpha_l} + a_{u_n,\tau,\alpha_m} \max\left\{\min\left\{\frac{\alpha_m - \nu}{1-\lambda}, 1\right\}, 0\right\}.$$

Hence by (4.31) we must have either $m = k$ if $\nu < \alpha_m$ or $m = k-1$ if $\nu \ge \alpha_m$. If $\nu < \alpha_m$,

$$\frac{\alpha_k - \nu}{1-\lambda} = a^{-1}_{u_n,\tau,\alpha_k}\left(M - \sum_{\ell=k+1}^{K} a_{u_n,\tau,\alpha_\ell}\right)$$

which by (4.26) gives (4.32). If $\nu \in [\alpha_m, \alpha_{m+1} - (1-\lambda))$ then by (4.26) and since $m = k-1$

$$(u^\lambda_{n+1})_i = \begin{cases} 0, & \text{if and only if } (e^{-\tau\Delta}u_n)_i \le \alpha_m = \alpha_{k-1}, \\ 1, & \text{if and only if } (e^{-\tau\Delta}u_n)_i \ge \alpha_{m+1} = \alpha_k. \end{cases}$$

Therefore

$$M = \sum_{\ell=k}^{K} a_{u_n,\tau,\alpha_\ell},$$

so it follows that

$$a^{-1}_{u_n,\tau,\alpha_k}\left(M - \sum_{\ell=k+1}^{K} a_{u_n,\tau,\alpha_\ell}\right) = 1$$

and so (4.32) follows. □

Note 21. *The RHS of* (4.32) *can immediately be seen to solve* (4.5) *as it satisfies the conditions of* (4.14). *Furthermore, note that as* $\lambda \uparrow 1$, u^λ_{n+1} *converges to a point in* $\operatorname{Ext} X$ (*i.e. the RHS of* (4.32) *is in* $\operatorname{Ext} X$) *if and only if* (4.15) *holds, i.e. if and only if* (4.5) *has a unique solution and* u^λ_{n+1} *converges to the unique solution of* (4.5).

Corollary 4.3.3. *Let* $u_0 \in \mathcal{V}_{[0,1]}$, $\tau \ge 0$ *be fixed, and for all* $\lambda \in [0, 1)$, *let* u^λ_n *be defined iteratively by* (4.3) (*respectively* (4.4)) *with* $u^\lambda_0 = u_0$. *Then there exists* u^1_n, *defined iteratively by the* $\lambda = 1$ *case of* (4.3) (*respectively* (4.4)) *with* $u^1_0 = u_0$, *such that for all* N *there exists* $\delta > 0$ *such that for all* $\lambda \in (1-\delta, 1)$ *and* $n \le N$, $u^\lambda_n = u^1_n$.

Proof. By the above theorem, there exists $\delta_1 > 0$ depending only on u_0 and τ such that for $\lambda \in (1-\delta_1, 1)$, u_1^λ is constant in λ. By repeating the same argument, there exists $\delta_2 \in (0, \delta_1]$ such that for $\lambda \in (1-\delta_2, 1)$, u_1^λ and u_2^λ are constant in λ. Hence, for all N there exists $\delta_N > 0$ such that for $\lambda \in (1-\delta_N, 1)$ and $n \leq N$, u_n^λ is constant in λ. Finally, define u_n^1 to be that eventually constant value. □

4.3.2. Conditions on freezing

Similarly to the AC flow, the semi-discrete scheme experiences freezing if τ is taken too small. Results giving sufficient conditions for "too small" in the non-mass-conserving case were proved in [9, Theorem 4.2] (for the MBO scheme) and [3, Theorem 4.5] (for the semi-discrete scheme in the ordinary case). We here prove similar results in the fidelity forced and mass-conserving cases.

Lemma 4.3.4. *For any $S \subseteq V$ and $\alpha \geq 0$, if*

$$\tau < ||\Delta||^{-1} \log\left(1 + \alpha \sqrt{\frac{\min_{i\in V} d_i^r}{\mathcal{M}(\chi_S)}}\right) \text{ or } \tau < \alpha ||\Delta\chi_S||_\infty^{-1}, \tag{4.33}$$

then $||e^{-\tau\Delta}\chi_S - \chi_S||_\infty < \alpha$, and if

$$\tau \leq ||\Delta||^{-1} \log\left(1 + \alpha \sqrt{\frac{\min_{i\in V} d_i^r}{\mathcal{M}(\chi_S)}}\right) \text{ or } \tau \leq \alpha ||\Delta\chi_S||_\infty^{-1}, \tag{4.34}$$

then $||e^{-\tau\Delta}\chi_S - \chi_S||_\infty \leq \alpha$.

Likewise, if

$$\tau < ||A||^{-1} \log\left(1 + \alpha \frac{\min_{i\in V} d_i^{r/2}}{\sqrt{\mathcal{M}(\chi_S)} + ||A||^{-1}||f||_\mathcal{V}}\right) \text{ or } \tau < \alpha C^{-1} ||A\chi_S - f||_\infty^{-1}, \tag{4.35}$$

then $||\mathcal{S}_\tau\chi_S - \chi_S||_\infty < \alpha$, and if

$$\tau \leq ||A||^{-1} \log\left(1 + \alpha \frac{\min_{i\in V} d_i^{r/2}}{\sqrt{\mathcal{M}(\chi_S)} + ||A||^{-1}||f||_\mathcal{V}}\right) \text{ or } \tau \leq \alpha C^{-1} ||A\chi_S - f||_\infty^{-1}, \tag{4.36}$$

then $||\mathcal{S}_\tau\chi_S - \chi_S||_\infty \leq \alpha$, where $C := \sup_{t\in[0,\infty)} ||e^{-tA}\mathbf{1}||_\infty$ satisfies

$$C \leq \left(\min_{i\in V} d_i^{r/2}\right)^{-1} \sup_{t\in[0,\infty)} ||e^{-tA}\mathbf{1}||_\mathcal{V} \leq \left(\min_{i\in V} d_i^{r/2}\right)^{-1} ||\mathbf{1}||_\mathcal{V}, \text{ and}$$
$$C \geq ||\mathbf{1}||_\infty = 1.$$

Proof. It suffices to prove the (4.36) case, as the (4.35) case is the same just with a strict inequality, and the (4.33) and (4.34) cases can be derived by setting $\mu = \mathbf{0}$ and hence $f = \mathbf{0}$, $A = \Delta$, $\mathcal{S}_\tau = e^{-\tau\Delta}$, and $C = 1$. We now follow the proof of [9, Theorem 4.2].

It is straightforward to check that for all $u \in \mathcal{V}$, $||u||_\infty \leq \left(\min_{i\in V} d_i^{r/2}\right)^{-1} ||u||_\mathcal{V}$. Hence

$$\begin{aligned}||\mathcal{S}_\tau\chi_S - \chi_S||_\infty &\leq \left(\min_{i\in V} d_i^{r/2}\right)^{-1} ||\mathcal{S}_\tau\chi_S - \chi_S||_\mathcal{V} \\ &\leq \left(\min_{i\in V} d_i^{r/2}\right)^{-1} \left(||(e^{-\tau A} - I)\chi_S||_\mathcal{V} + \left|\left| \sum_{n=0}^{\infty} (-1)^n \frac{\tau^{n+1}}{(n+1)!} A^n f \right|\right|_\mathcal{V} \right) \\ &\leq \left(\min_{i\in V} d_i^{r/2}\right)^{-1} \left(e^{\tau||A||} - 1\right)\left(\sqrt{\mathcal{M}(\chi_S)} + ||A||^{-1}||f||_\mathcal{V}\right)\end{aligned}$$

and setting $RHS \leq \alpha$ gives the first condition in (4.36).

Next, note that since $\mathcal{S}_\tau$ is the solution operator for (3.6)

$$\begin{aligned}||\mathcal{S}_\tau\chi_S - \chi_S||_\infty &= \left|\left| \int_0^\tau -A\mathcal{S}_t\chi_S + f \; dt \right|\right|_\infty = \left|\left| \int_0^\tau -e^{-tA}\left(A\chi_S - f\right) \; dt \right|\right|_\infty \\ &\leq \int_0^\tau \left|\left| e^{-tA}\left(A\chi_S - f\right)\right|\right|_\infty \; dt \\ &\leq \int_0^\tau C\,||A\chi_S - f||_\infty \; dt \qquad (*) \\ &= \tau C\,||A\chi_S - f||_\infty \; dt\end{aligned}$$

where the second inequality follows from (3.7) and Definition 3.6. Setting $RHS \leq \alpha$ gives the second condition in (4.36).

It suffices then to check line $(*)$. Recall from the proof of Theorem 3.2.6 that e^{-tA} is a non-negative matrix. Then for any $u \in V$, $-||u||_\infty \mathbf{1} \leq u \leq ||u||_\infty \mathbf{1}$ vertexwise and so $|e^{-tA}u| \leq ||u||_\infty e^{-tA}\mathbf{1}$ vertexwise. It follows that $||e^{-tA}u||_\infty \leq C||u||_\infty$. Finally, since (by Proposition 4.2.7) the eigenvalues of e^{-tA} are all in $[0,1]$ and we have the inequality at the start of this proof relating the $\mathcal{V}$ and infinity norms, we get the desired upper bound for C. The lower bound for C follows immediately from the definition of C by taking $t = 0$. □

Theorem 4.3.5. *If $S \subseteq V$ and τ obey* (4.35) (*respectively* (4.33)) *for $\alpha = \frac{1}{2}$, $\lambda = 1$, and $u_n = \chi_S$, then u solves* (4.3) (*respectively* (4.4)) *if and only if $u = \chi_S$.*

Proof. Let $Q := \mathcal{S}_\tau$ or $Q := e^{-\tau\Delta}$ respectively. By Lemma 4.3.4, we have that $||Q\chi_S - \chi_S||_\infty < \frac{1}{2}$ and it follows that $\max_{i\in S^c}(Q\chi_S)_i < \frac{1}{2} < \min_{i\in S}(Q\chi_S)_i$.

Hence if u solves (4.3), then by (4.7) $u = \chi_S$, and so χ_S solves (4.3).

In the mass-conserving case, recall from Corollary 4.2.12 that for $M = \mathcal{M}(\chi_S)$, $S_{\tau,\chi_S} \cap \mathrm{Ext}\, X_M$ is non-empty, so consider an arbitrary $u \in S_{\tau,\chi_S} \cap \mathrm{Ext}\, X_M$. By Corollary 4.2.14, to prove the theorem it will suffice to prove that u must equal χ_S.

By Theorem 4.2.13, if $u_i > 0$ and $u_j < 1$ then $(e^{-\tau\Delta}\chi_S)_i \geq (e^{-\tau\Delta}\chi_S)_j$. Thus if $i \in S^c$ and $j \in S$ then $(e^{-\tau\Delta}\chi_S)_i < \frac{1}{2} < (e^{-\tau\Delta}\chi_S)_j$ and hence $u_i = 0$ or $u_j = 1$. Suppose that $u_j < 1$ for some $j \in S$, then by the above $u_i = 0$ for all $i \in S^c$. But

then $u \leq \chi_S$ vertexwise and $u_j < (\chi_S)_j$, so $\mathcal{M}(u) < \mathcal{M}(\chi_S)$, a contradiction. Hence $u_j = 1$ for all $j \in S$. Likewise, $u_i = 0$ for all $i \in S^c$. □

Theorem 4.3.6. *Let $S \subseteq V$, $\lambda \in [0,1)$, τ obey (4.36) (respectively (4.34)) for $\alpha = \frac{1}{2}\lambda$, and $u_n = \chi_S$. Then u solves (4.3) (respectively (4.4)) if and only if $u = \chi_S$.*

Proof. Recall that solutions to (4.3) and (4.4) are unique for $\lambda \in [0,1)$, so it suffices to show that $u = \chi_S$ is a valid solution. Let $Q := \mathcal{S}_\tau$ or $Q := e^{-\tau\Delta}$ respectively. Then by Lemma 4.3.4 we have that $||Q\chi_S - \chi_S||_\infty \leq \frac{1}{2}\lambda$, and hence $(Q\chi_S)_i \leq \frac{1}{2}\lambda$ if and only if $i \in S^c$, and $(Q\chi_S)_i \geq 1 - \frac{1}{2}\lambda$ if and only if $i \in S$.

By (4.6), it follows that $u = \chi_S$ solves (4.3) (i.e., the fidelity forced case).

In the mass-conserving case, taking $\nu = \frac{1}{2}\lambda$ we observe that (recalling that $M := \mathcal{M}(u_n) = \mathcal{M}(\chi_S)$)

$$
\begin{aligned}
&M + \sum_i d_i^r \begin{cases} -1, & \nu \leq (e^{-\tau\Delta}u_n)_i - (1-\lambda), \\ \frac{\nu - (e^{-\tau\Delta}u_n)_i}{1-\lambda}, & (e^{-\tau\Delta}u_n)_i - (1-\lambda) < \nu < (e^{-\tau\Delta}u_n)_i, \\ 0, & \nu \geq (e^{-\tau\Delta}u_n)_i, \end{cases} \\
&= M + \sum_i d_i^r \begin{cases} -1, & 1 - \frac{1}{2}\lambda \leq (e^{-\tau\Delta}u_n)_i, \\ \frac{\nu - (e^{-\tau\Delta}u_n)_i}{1-\lambda}, & (e^{-\tau\Delta}u_n)_i \in \left(\frac{1}{2}\lambda, 1 - \frac{1}{2}\lambda\right), \\ 0, & \frac{1}{2}\lambda \geq (e^{-\tau\Delta}u_n)_i, \end{cases} \\
&= M + \sum_i d_i^r \begin{cases} -1, & i \in S, \\ 0, & i \in S^c, \end{cases} \\
&= M - \mathcal{M}(\chi_S) = 0.
\end{aligned}
$$

Thus $\nu = \frac{1}{2}\lambda$ solves (4.23), and so by (4.22) u is given by

$$
\begin{aligned}
u_i &= \begin{cases} 0, & \text{if and only if } \nu \geq (e^{-\tau\Delta}u_n)_i, \\ \frac{(e^{-\tau\Delta}u_n)_i - \nu}{1-\lambda}, & \text{if and only if } (e^{-\tau\Delta}u_n)_i - (1-\lambda) < \nu < (e^{-\tau\Delta}u_n)_i, \\ 1, & \text{if and only if } \nu \leq (e^{-\tau\Delta}u_n)_i - (1-\lambda), \end{cases} \\
&= \begin{cases} 0, & \text{if and only if } \frac{1}{2}\lambda \geq (e^{-\tau\Delta}u_n)_i, \\ \frac{(e^{-\tau\Delta}u_n)_i - \frac{1}{2}\lambda}{1-\lambda}, & \text{if and only if } (e^{-\tau\Delta}u_n)_i \in \left(\frac{1}{2}\lambda, 1 - \frac{1}{2}\lambda\right), \\ 1, & \text{if and only if } 1 - \frac{1}{2}\lambda \leq (e^{-\tau\Delta}u_n)_i, \end{cases} \\
&= \begin{cases} 0, & \text{if and only if } i \in S^c, \\ 1, & \text{if and only if } i \in S, \end{cases}
\end{aligned}
$$

so $u = \chi_S$ is a valid solution. □

4.3.3. Bounds on β_n

Lemma 4.3.7. *Let $0 < \lambda \leq 1$ and $u_n \in \mathcal{V}_{[0,1]}$.*

If (u_{n+1}, β_{n+1}) solves (4.1), then

$$\beta_{n+1} \in \mathcal{V}_{[-1/2,1/2]}. \tag{4.37}$$

If (u_{n+1}, β_{n+1}) solves (4.2) and $\bar{u} := \overline{u_n} = \overline{u_{n+1}} \in (0,1)$, then

$$\beta_{n+1} - \overline{\beta_{n+1}}\mathbf{1} \in \mathcal{V}_{[\bar{u}-1,\bar{u}]} \tag{4.38}$$

and

$$\beta_{n+1} \in \mathcal{V}_{[-1,1]}. \tag{4.39}$$

Proof. If (u_{n+1}, β_{n+1}) solves (4.1), then (4.37) follows immediately from the characterisation of β_{n+1} in Theorem 4.2.1 and from $\mathcal{S}_\tau u_n \in \mathcal{V}_{[0,1]}$ (Theorem 3.2.6).

If (u_{n+1}, β_{n+1}) solves (4.2) and $\bar{u} := \overline{u_n} = \overline{u_{n+1}} \in (0,1)$, then first suppose that $\lambda = 1$. Then by (4.27),

$$(\beta_{n+1})_i - \overline{\beta_{n+1}} = \bar{u} - (e^{-\tau\Delta}u_n)_i \in [\bar{u}-1, \bar{u}].$$

Next, suppose $\lambda \in (0,1)$. Then by (4.29),

$$\begin{aligned}(\beta_{n+1})_i - \overline{\beta_{n+1}} &= \bar{u} + \frac{1}{\lambda}\begin{cases} -(e^{-\tau\Delta}u_n)_i, & \text{if } \nu \geq (e^{-\tau\Delta}u_n)_i, \\ -\nu, & \text{if } (e^{-\tau\Delta}u_n)_i - (1-\lambda) < \nu < (e^{-\tau\Delta}u_n)_i, \\ 1-\lambda-(e^{-\tau\Delta}u_n)_i, & \text{if } \nu \leq (e^{-\tau\Delta}u_n)_i - (1-\lambda), \end{cases} \\ &= \bar{u} - 1 + \frac{1}{\lambda}\begin{cases} \lambda - (e^{-\tau\Delta}u_n)_i, & \text{if } \nu \geq (e^{-\tau\Delta}u_n)_i, \\ \lambda - \nu, & \text{if } (e^{-\tau\Delta}u_n)_i - (1-\lambda) < \nu < (e^{-\tau\Delta}u_n)_i, \\ 1-(e^{-\tau\Delta}u_n)_i, & \text{if } \nu \leq (e^{-\tau\Delta}u_n)_i - (1-\lambda), \end{cases}\end{aligned}$$

where we recall from Proposition 4.2.18 that $\nu \in (0, \lambda)$. It is therefore easy to check that in the first line the conditional term is non-positive, and in the second line the conditional term is non-negative. Therefore we deduce (4.38).

Consider the set $\mathscr{B} := \{(\beta_{n+1})_i \mid i \in V\}$. By (4.38), $\mathscr{B} - \overline{\beta_{n+1}} \subseteq [\bar{u}-1, \bar{u}]$, so we have that $\operatorname{diam}\mathscr{B} \leq 1$. Furthermore, $u_{n+1} \notin \{\mathbf{0}, \mathbf{1}\}$, so since $\beta_{n+1} \in \mathcal{B}(u_{n+1})$ we have $x, y \in \mathscr{B}$ such that $x \geq 0$ and $y \leq 0$. Therefore $\mathscr{B} \subseteq [x-1, x+1] \cap [y-1, y+1] \subseteq [-1, 1]$. □

4.4. Eventual behaviour of the SDIE scheme

In this section we exhibit Lyapunov functionals for each of the cases of the SDIE scheme, and use these to examine the eventual behaviour of the schemes.

Lemma 4.4.1 (Cf. [9, Lemma 4.5])**.** *Define the following functionals on $\mathcal{V}$*

$$\begin{aligned} J_0(u) &:= \langle \mathbf{1} - u, e^{-\tau\Delta}u \rangle_V, \\ J(u) &:= \langle u, \mathbf{1} - 2F_\tau(A)f - e^{-\tau A}u \rangle_V. \end{aligned}$$

These have the following properties:

i. J_0 *and* J *are strictly concave.*

ii. J_0 *has first variation at* u

$$(L_0)_u(v) := \left\langle v, \mathbf{1} - 2e^{-\tau\Delta}u \right\rangle_{\mathcal{V}}.$$

iii. J *has first variation at* u

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

Proof. Note that J_0 is the $\mu = \mathbf{0}$ case of J (since $\langle \mathbf{1}, e^{-\tau\Delta}u\rangle_{\mathcal{V}} = \langle u, \mathbf{1}\rangle_{\mathcal{V}}$), so it suffices to prove the results for J. Let $w := \mathbf{1} - 2F_\tau(A)f$. We expand around u:

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

Then to prove (i), note that $\frac{d^2}{dt^2}J(u+tv) = -2\langle v, e^{-\tau A}v\rangle_{\mathcal{V}} < 0$ for $v \neq \mathbf{0}$. To prove (iii), note that since $e^{-\tau A}$ is self-adjoint, $J(u+tv) = J(u) + tL_u(v) + \mathcal{O}(t^2)$. □

Theorem 4.4.2. *For* $0 \leq \lambda \leq 1$ *we define on* $\mathcal{V}_{[0,1]}$ *the functionals*

$$H_0(u) := J_0(u) + (\lambda - 1)\langle u, \mathbf{1} - u\rangle_{\mathcal{V}}, \tag{4.40a}$$

$$H(u) := J(u) + (\lambda - 1)\langle u, \mathbf{1} - u\rangle_{\mathcal{V}}. \tag{4.40b}$$

These have uniform lower bounds

$$H_0(u) \geq 0, \qquad H(u) \geq -2\tau||f||_{\mathcal{V}}||\mathbf{1}||_{\mathcal{V}}.$$

Furthermore, H *is a Lyapunov functional for the fidelity forced SDIE scheme* (4.1)*, and* H_0 *is a Lyapunov functional for the mass-conserving SDIE scheme* (4.2)*, i.e.* $H(u_{n+1}) \leq H(u_n)$ *with equality if and only if* $u_{n+1} = u_n$ *for* u_{n+1} *defined by* (4.1)*, and* $H_0(u_{n+1}) \leq H_0(u_n)$ *with equality if and only if* $u_{n+1} = u_n$ *for* u_{n+1} *defined by* (4.2)*. In particular, we have that for* u_{n+1} *given by* (4.1)

$$H(u_n) - H(u_{n+1}) \geq (1-\lambda)\,||u_{n+1} - u_n||_{\mathcal{V}}^2 \tag{4.41}$$

and for u_{n+1} *given by* (4.2)

$$H_0(u_n) - H_0(u_{n+1}) \geq (1-\lambda)\,||u_{n+1} - u_n||_{\mathcal{V}}^2. \tag{4.42}$$

Proof. We begin with proving the lower bounds. Note that H_0 is the $\mu = \mathbf{0}$ and hence $f = \mathbf{0}$ case of H, so it suffices to prove the lower bound for H. We can rewrite H as:

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

where the final line follows since $F_\tau(A)$ is self-adjoint (since A is) and has eigenvalues

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

so we have by Proposition 3.2.4 that

$$\begin{aligned} ||F_\tau(A)|| &\leq \sup_{x\in(0,||\Delta||+||\mu||_\infty]} \frac{1-e^{-\tau x}}{x} \\ &= \lim_{x\to 0} \frac{1-e^{-\tau x}}{x} \quad \text{as } x \mapsto x^{-1}(1-e^{-\tau x}) \text{ is monotonically decreasing}^4 \\ &= \tau. \end{aligned}$$

Next we show that H is a Lyapunov functional for (4.1). By the concavity of J:

$$\begin{aligned} &H(u_n) - H(u_{n+1}) \\ &= J(u_n) - J(u_{n+1}) + (1-\lambda)\langle u_{n+1}, \mathbf{1} - u_{n+1}\rangle_\mathcal{V} - (1-\lambda)\langle u_n, \mathbf{1} - u_n\rangle_\mathcal{V} \\ &\geq L_{u_n}(u_n - u_{n+1}) + (1-\lambda)\langle u_{n+1}, \mathbf{1} - u_{n+1}\rangle_\mathcal{V} - (1-\lambda)\langle u_n, \mathbf{1} - u_n\rangle_\mathcal{V} \qquad (*) \\ &= \langle u_n - u_{n+1}, \mathbf{1} - 2\mathcal{S}_\tau u_n\rangle_\mathcal{V} + (1-\lambda)\langle u_{n+1}, \mathbf{1} - u_{n+1}\rangle_\mathcal{V} - (1-\lambda)\langle u_n, \mathbf{1} - u_n\rangle_\mathcal{V} \\ &= \langle u_n - u_{n+1}, \mathbf{1} - 2\mathcal{S}_\tau u_n\rangle_\mathcal{V} + (1-\lambda)(\langle u_{n+1} - u_n, \mathbf{1}\rangle_\mathcal{V} + \langle u_n, u_n\rangle_\mathcal{V} - \langle u_{n+1}, u_{n+1}\rangle_\mathcal{V}) \\ &= \langle u_n - u_{n+1}, \lambda\mathbf{1} - 2\mathcal{S}_\tau u_n + (1-\lambda)u_{n+1} + (1-\lambda)u_n\rangle_\mathcal{V} \\ &= \langle u_n - u_{n+1}, 2\lambda\beta_{n+1} + (1-\lambda)(u_n - u_{n+1})\rangle_\mathcal{V} \text{ by (4.1)} \\ &\geq (1-\lambda)\,||u_{n+1} - u_n||_\mathcal{V}^2 \geq 0 \end{aligned}$$

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 by Lemma 3.4.6.

Finally, we show that H_0 is a Lyapunov functional for (4.2). By the concavity of J_0, the linearity of $(L_0)_{u_n}$, and recalling that $\langle u_n - u_{n+1}, \mathbf{1}\rangle_\mathcal{V} = 0$:

$$\begin{aligned} &H_0(u_n) - H_0(u_{n+1}) \\ &= J_0(u_n) - J_0(u_{n+1}) + (1-\lambda)\langle u_{n+1}, \mathbf{1} - u_{n+1}\rangle_\mathcal{V} - (1-\lambda)\langle u_n, \mathbf{1} - u_n\rangle_\mathcal{V} \\ &\geq (L_0)_{u_n}(u_n - u_{n+1}) - (1-\lambda)\langle u_{n+1}, u_{n+1}\rangle_\mathcal{V} + (1-\lambda)\langle u_n, u_n\rangle_\mathcal{V} \qquad (**) \\ &= \langle u_n - u_{n+1}, \mathbf{1} - 2e^{-\tau\Delta}u_n\rangle_\mathcal{V} - (1-\lambda)\langle u_{n+1}, u_{n+1}\rangle_\mathcal{V} + (1-\lambda)\langle u_n, u_n\rangle_\mathcal{V} \\ &= \langle u_n - u_{n+1}, -2e^{-\tau\Delta}u_n + (1-\lambda)(u_{n+1} + u_n)\rangle_\mathcal{V} \\ &= \langle u_n - u_{n+1}, 2(1-\lambda)u_{n+1} - 2e^{-\tau\Delta}u_n + (1-\lambda)(u_n - u_{n+1})\rangle_\mathcal{V} \\ &= \left\langle u_n - u_{n+1}, 2\lambda\beta_{n+1} - 2\lambda\overline{u_{n+1}}\mathbf{1} - 2\overline{\beta_{n+1}}\mathbf{1} + (1-\lambda)(u_n - u_{n+1}) \right\rangle_\mathcal{V} \text{ by (4.2)} \\ &= \langle u_n - u_{n+1}, 2\lambda\beta_{n+1}\rangle_\mathcal{V} + (1-\lambda)\,||u_{n+1} - u_n||_\mathcal{V}^2 \\ &\geq (1-\lambda)\,||u_{n+1} - u_n||_\mathcal{V}^2 \geq 0 \end{aligned}$$

with equality in $(**)$ if and only if $u_{n+1} = u_n$ as the concavity of J_0 is strict, and where the last line follows by Lemma 3.4.6. □

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

As a corollary, we consider conditions under which the MBO sequence (i.e. the $\lambda = 1$ SDIE sequence) is eventually constant.

Corollary 4.4.3 (Cf. [9, Proposition 4.6] and [10, Lemma 5.18])**.** *Let $\lambda = 1$. If a fidelity forced MBO sequence u_n defined by (4.1) satisfies $u_n \in \mathcal{V}_{\{0,1\}}$ for eventually all n, then there exists $u \in \mathcal{V}_{\{0,1\}}$ such that for eventually all n, $u_n = u$.*

Furthermore, recall from Definitions 4.2.3 and 4.2.8 (respectively) the notations X_M, for the set of $u \in \mathcal{V}_{[0,1]}$ with $\mathcal{M}(u) = M$, and S_{τ,u_n}, for the set of valid mass-conserving MBO updates of u_n, i.e. the set of solutions to (4.5). If $M = \mathcal{M}(u_0)$ and a mass-conserving MBO sequence u_n defined by (4.5) satisfies either:

(i) *for eventually all n, $u_{n+1} \in \operatorname{Ext} S_{\tau,u_n}$, or*

(ii) *for eventually all n, u_{n+1} is as in (4.32) (i.e. the $\lambda \uparrow 1$ limit of the semi-discrete updates u^{λ}_{n+1}),*

then there exists $u \in X_M$ such that for eventually all n, $u_n = u$.

Proof. Note that $\mathcal{V}_{\{0,1\}}$ is a finite set. Hence $\{u_n \mid n \in \mathbb{N}\}$ is a finite set, so if the u_n are not eventually a single u then we must have some $u, v \in \mathcal{V}_{\{0,1\}}$ such that $u \neq v$, $u_n = u$ infinitely often, and $u_n = v$ infinitely often. Therefore we must have $n < m < k$ such that $u_n = u_k = u$ and $u_m = v$, and hence

$$H(u) \geq H(u_{n+1}) \geq \ldots \geq H(u_{m-1}) \geq H(v) \geq H(u_{m+1}) \geq \ldots \geq H(u_{k-1}) \geq H(u).$$

All the inequalities are equalities, and therefore by the equality condition on H from Theorem 4.4.2 we have $u = v$, a contradiction. Thus the u_n are eventually constant.

In the mass-conserving cases, it will likewise suffice to show that the sequence is eventually contained within a finite set. For (i), recall from Propositions 4.2.11 and 4.2.5 (respectively) that $\operatorname{Ext} S_{\tau,u_n} = S_{\tau,u_n} \cap \operatorname{Ext} X_M \subseteq \operatorname{Ext} X_M$ and that $\operatorname{Ext} X_M$ is a finite set. For (ii), we show that there are finitely many possible $u \in X_M$ of the form (4.32). Each such u has the form

$$u = \frac{M - \mathcal{M}(\chi_{V_3})}{\mathcal{M}(\chi_{V_2})}\chi_{V_2} + \chi_{V_3}$$

for a partition $V = V_1 \cup V_2 \cup V_3$ with $0 \leq M - \mathcal{M}(\chi_{V_3}) \leq \mathcal{M}(\chi_{V_2})$. To see this, note that u as in (4.32) has $V_1 = \{i \mid (e^{-\tau\Delta}u_n)_i < \alpha_k\}$, $V_2 = \{i \mid (e^{-\tau\Delta}u_n)_i = \alpha_k\}$, and $V_3 = \{i \mid (e^{-\tau\Delta}u_n)_i > \alpha_k\}$. But since V is finite, there are only finitely many tripartitions of V. □

Corollary 4.4.4. *If $\lambda \in [0, 1)$ and the sequence u_n obeys (4.1) or (4.2), then*

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

and therefore in particular

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

If $\lambda = 1$ then in any of the cases of Corollary 4.4.3, this result also holds.

Proof. For $\lambda < 1$, in the (4.1) case by the lower bound on H from Theorem 4.4.2 and (4.41) we have

$$(1-\lambda)\sum_{n=0}^{N}||u_{n+1}-u_n||_{\mathcal{V}}^2 \leq H(u_0)-H(u_{N+1}) \leq H(u_0)+2\tau||f||_{\mathcal{V}}||\mathbf{1}||_{\mathcal{V}}$$

so the result follows by taking $N \to \infty$. The case for (4.2) is likewise.

For $\lambda = 1$, the result trivially holds when u_n is eventually constant. □

We wish to use the gradient of H and H_0 to investigate critical points of the flow. However as in the mass-conserving case we will restrict the flow to lie in S_M, a non-Hilbert space, we make the following definition.

Definition 4.4.5. *Let Y_0 be a Hilbert space with inner product $\langle\cdot,\cdot\rangle_{Y_0}$, and let $Y_1 \subseteq Y_0$ be a closed subspace. Let $\tilde{Y} := x + Y_1$ for some $x \in Y_0$. Let $f : Y_0 \to \mathbb{R}$ be a Fréchet differentiable map, with Fréchet derivative Df defined at each $u \in Y_0$ to be the unique linear map such that*

$$f(u+h) = f(u) + Df(u)(h) + o(h).$$

Then we define the Fréchet derivative of $f|_{\tilde{Y}}$ at $u \in \tilde{Y}$ by

$$Df|_{\tilde{Y}}(u) := Df(u)|_{Y_1}$$

where the restriction of the argument to Y_1 ensures that for $u \in \tilde{Y}$ the $u+h$ terms satisfy $u+h \in \tilde{Y}$. Then we define the gradient

$$\nabla_{\tilde{Y}} f|_{\tilde{Y}}(u) \in Y_1$$

to be the Riesz representative of $Df|_{\tilde{Y}}(u)$, i.e. the unique element of Y_1 such that

$$\forall v \in Y_1 \ \ \langle \nabla_{\tilde{Y}} f|_{\tilde{Y}}(u), v\rangle_{Y_0} = Df|_{\tilde{Y}}(u)(v) = Df(u)(v).$$

Note therefore that for $u \in \tilde{Y}$, since $\nabla_{Y_0} f(u)$ is the Riesz representative of $Df(u)$,

$$\forall v \in Y_1 \ \ \langle \nabla_{Y_0} f(u), v\rangle_{Y_0} = \langle \nabla_{\tilde{Y}} f|_{\tilde{Y}}(u), v\rangle_{Y_0}$$

and so $\left(\nabla_{\tilde{Y}} f|_{\tilde{Y}}(u) - \nabla_{Y_0} f(u)\right) \perp Y_1$. That is, for $u \in \tilde{Y}$, $\nabla_{\tilde{Y}} f|_{\tilde{Y}}(u)$ is the orthogonal projection of $\nabla_{Y_0} f(u)$ onto Y_1.

Proposition 4.4.6. *Let $u \in \mathcal{V}_{(0,1)}$. Then the Lyapunov functional H for (4.1) has Hilbert space gradient:*

$$\nabla_{\mathcal{V}} H(u) = \lambda \mathbf{1} - 2\mathcal{S}_\tau u + 2(1-\lambda)u. \tag{4.43}$$

Next, let $u \in \mathcal{V}_{(0,1)} \cap X_M$ (and so $M := \mathcal{M}(u)$). Then the Lyapunov functional H_0 for (4.2) has restricted gradient:

$$\nabla_{S_M} H_0|_{S_M}(u) = 2(u - e^{-\tau\Delta}u) - 2\lambda u + 2\lambda\bar{u}\mathbf{1}. \tag{4.44}$$

Therefore:

i. *For a sequence* $u_n \in \mathcal{V}_{(0,1)}$ *given by* (4.1)

$$\nabla_{\mathcal{V}} H(u_n) = 2(1-\lambda)(u_n - u_{n+1}). \tag{4.45}$$

ii. *If* $u \in \mathcal{V}_{(0,1)}$*, it follows that* $\nabla_{\mathcal{V}} H(u) = \mathbf{0}$ *(i.e.* u *is a critical point of* H*) if and only if* $(e^{-\tau A} - (1-\lambda)I)u = \frac{1}{2}\lambda\mathbf{1} - F_\tau(A)f$*.*

iii. *Such a critical point is a global maximum of* H *in* $\mathcal{V}_{[0,1]}$ *if and only if* $e^{-\tau A} - (1-\lambda)I$ *is positive semi-definite, which holds if and only if* $e^{-\lambda||A||} \geq 1-\lambda$*.*

iv. *For a sequence* $u_n \in \mathcal{V}_{(0,1)} \cap X_M$ *obeying* (4.2)

$$\nabla_{S_M} H_0|_{S_M}(u_n) = 2(1-\lambda)(u_n - u_{n+1}). \tag{4.46}$$

v. *Define* $\mathscr{E}$ *to be the eigenspace of* Δ *with eigenvalue* $-\tau^{-1}\log(1-\lambda)$*, or* $\{\mathbf{0}\}$ *if there is no such eigenvalue. If* $u \in \mathcal{V}_{(0,1)} \cap X$ *then* $\nabla_{S_M} H_0|_{S_M}(u) = \mathbf{0}$ *if and only if* $u \in (\bar{u}\mathbf{1} + \mathscr{E}) \cap \mathcal{V}_{(0,1)}$*.*

vi. $\bar{u}\mathbf{1}$ *is a global maximum of* H_0 *in* X_M *if and only if* $e^{-\tau\Delta} - (1-\lambda)I$ *is positive semi-definite, which holds if and only if* $e^{-\lambda||\Delta||} \geq 1-\lambda$*.*

Proof. It is straightforward to check that

$$\langle \nabla_{\mathcal{V}} H(u), v\rangle_{\mathcal{V}} := \lim_{t\to 0} \frac{H(u+tv) - H(u)}{t} = \langle \mathbf{1} - 2\mathcal{S}_\tau u, v\rangle_{\mathcal{V}} + (\lambda-1)\langle \mathbf{1} - 2u, v\rangle_{\mathcal{V}}$$

and therefore

$$\nabla_{\mathcal{V}} H(u) = \mathbf{1} - 2\mathcal{S}_\tau u + (\lambda-1)(\mathbf{1} - 2u) = \lambda\mathbf{1} - 2\mathcal{S}_\tau u + 2(1-\lambda)u.$$

Plugging in $\mu = \mathbf{0}$ into the above gives

$$\nabla_{\mathcal{V}} H_0(u) = 1 - 2e^{-\tau\Delta}u + (\lambda-1)(1-2u) = \lambda\mathbf{1} - 2e^{-\tau\Delta}u + 2(1-\lambda)u.$$

Restricting to $S_M = u + \{\mathbf{1}\}^\perp$, by definition $\nabla_{S_M} H_0|_{S_M}(u) \in \{\mathbf{1}\}^\perp$ and $\nabla_{S_M} H_0|_{S_M}(u) - \nabla_{\mathcal{V}} H_0(u) \in \operatorname{span}\{\mathbf{1}\}$. Thus $\nabla_{S_M} H_0|_{S_M}(u) = \nabla_{\mathcal{V}} H_0(u) - \overline{\nabla_{\mathcal{V}} H_0(u)}\mathbf{1}$, yielding (4.44).

i. Since $u_{n+1} \in \mathcal{V}_{(0,1)}$ we have $\beta_{n+1} = \mathbf{0}$, and by (4.1), $\lambda\mathbf{1} - 2S_\tau u_n = 2\lambda\beta_{n+1} - 2(1-\lambda)u_{n+1}$. Thus substituting into (4.43) gives (4.45).

ii. Follows trivially by (4.43) and the definition of $\mathcal{S}_\tau$.

iii. Let $w := \frac{1}{2}\lambda\mathbf{1} - F_\tau(A)f$ and $P := e^{-\tau A} - (1-\lambda)I$. Then we can rewrite H as

$$H(u) = \langle u, 2w - Pu\rangle_{\mathcal{V}}.$$

If $u^* \in \mathcal{V}_{(0,1)}$ satisfies $Pu^* = w$, then

$$H(u^* + v) = \langle u^* + v, P(u^* - v)\rangle_{\mathcal{V}} = H(u^*) - \langle v, Pv\rangle_{\mathcal{V}}$$

which is less than or equal to $H(u^*)$ for all $v \in \mathcal{V}$ if and only if P is positive semi-definite.

iv. Since $u_{n+1} \in \mathcal{V}_{(0,1)}$ we have $\beta_{n+1} = \mathbf{0}$, so from (4.2) we have

$$u_{n+1} - e^{-\tau\Delta}u_n - \lambda u_{n+1} + \lambda\bar{u}\mathbf{1} = \lambda\beta_{n+1} - \lambda\overline{\beta_{n+1}}\mathbf{1} = \mathbf{0}$$

and so

$$(1-\lambda)u_{n+1} = e^{-\tau\Delta}u_n - \lambda\bar{u}$$

and (4.46) follows by substituting u_n into (4.44).

v. Let $B_1 : v \mapsto \bar{v}\mathbf{1}$ and define $B_2 := 2e^{-\tau\Delta} + 2(\lambda-1)I - 2\lambda B_1$. Then $\nabla_{S_M} H_0|_{S_M}(u) = \mathbf{0}$ if and only if $B_2 u = \mathbf{0}$. Note that $B_2\mathbf{1} = 2\mathbf{1} + 2\lambda\mathbf{1} - 2\mathbf{1} - 2\lambda\mathbf{1} = \mathbf{0}$ so $\frac{M}{\langle \mathbf{1},\mathbf{1}\rangle_{\mathcal{V}}}\mathbf{1} \in X_M$ is a solution. Taking $(\xi_k)_{k>0}$ the eigenvectors for Δ (with eigenvalues γ_k) as a basis for $\{\mathbf{1}\}^{\perp}$ we get that (recalling, from spectral property (c) in chapter 2, that $\xi_k \perp \mathbf{1}$ for $k > 0$ and $\gamma_k > 0$ for $k > 0$)

$$B_2\xi_k = 2(e^{-\tau\gamma_k} + \lambda - 1)\xi_k = \mathbf{0} \text{ if and only if } \xi_k \in \mathscr{E}.$$

Thus $B_2 u = \mathbf{0}$ if and only if

$$B_2\left(u - \frac{M}{\langle \mathbf{1},\mathbf{1}\rangle_{\mathcal{V}}}\mathbf{1}\right) = \mathbf{0}$$

if and only if

$$u - \frac{M}{\langle \mathbf{1},\mathbf{1}\rangle_{\mathcal{V}}}\mathbf{1} \in \mathscr{E}$$

as desired.

vi. Since H_0 is the $\mu = \mathbf{0}$ case of H, by the same argument as in (iii)

$$H_0(u) = \langle u, \lambda\mathbf{1} - Pu\rangle_{\mathcal{V}}.$$

for $P := e^{-\tau\Delta} - (1-\lambda)I$. Let $\eta \perp \mathbf{1}$, then since $P\mathbf{1} = \lambda\mathbf{1}$

$$H_0(\bar{u}\mathbf{1} + \eta) = \langle \bar{u}\mathbf{1} + \eta, \lambda\mathbf{1} - \bar{u}\lambda\mathbf{1} - P\eta\rangle_{\mathcal{V}} = H_0(\bar{u}\mathbf{1}) - \langle \eta, P\eta\rangle_{\mathcal{V}}$$

and the claim follows.

□

Since $H(u_n)$ (or $H_0(u_n)$; the statements in this paragraph apply in the mass-conserving case *mutatis mutandis*) is monotonically decreasing and bounded below, it follows that $H(u_n) \downarrow H_\infty$ for some $H_\infty \geq -2\tau||f||_{\mathcal{V}}||\mathbf{1}||_{\mathcal{V}}$. Furthermore, since the sequence u_n is contained in $\mathcal{V}_{[0,1]}$ which is compact, there exists a subsequence u_{n_k} that converges to some $u^* \in X$ with $H(u^*) = H_\infty$, since H is continuous. Unfortunately, just like Luo and Bertozzi [13] for graph AC flow with the standard quartic potential, we are unable to infer convergence of the whole sequence from these facts. However by the same argument as in [13, Lemma 5] if the set of accumulation points of the u_n is finite then there is in fact only one such point and the whole sequence converges. Notably, if $u^* \in \mathcal{V}_{(0,1)}$ is an accumulation point of the u_n then by Corollary 4.4.4 and (4.45) we have that $\nabla_{\mathcal{V}} H(u^*) = \mathbf{0}$. Thus, by Proposition 4.4.6(iii) if $e^{-\lambda||A||} \geq 1 - \lambda$, u^* is a global maximum of H on $\mathcal{V}_{[0,1]}$. Thus if $H(u_0) \neq H(u^*)$ then no accumulation points of the u_n lie in $\mathcal{V}_{(0,1)}$.

4.5. Convergence of the SDIE scheme to AC flow as $\tau \downarrow 0$

In this section, we shall show that trajectories of the SDIE scheme converge to solutions to AC flow as $\tau \downarrow 0$ with ε fixed. We will then use this characterisation of AC solutions to prove some key properties.

4.5.1. Set-up

To handle both the fidelity forced and mass-conserving cases at once, we write (4.1) and (4.2) in the form

$$(1-\lambda)u_{n+1} - e^{-\tau Q}u_n - w = \lambda\gamma_{n+1} \tag{4.47}$$

where $Q := A$ or $Q := \Delta$, $w := -\frac{1}{2}\lambda\mathbf{1} + F_\tau(A)f$ or $w := -\lambda\bar{u}\mathbf{1}$, and $\gamma_{n+1} := \beta_{n+1}$ or $\gamma_{n+1} := \beta_{n+1} - \overline{\beta_{n+1}}\mathbf{1}$ (all respectively).

Note 22. *In this section, Q, w, and γ_n will always denote the above quantities.*

We now solve the SDIE recurrence relation for the n^{th} term.

Proposition 4.5.1. *For $\lambda \in [0,1)$ the sequence generated by (4.47) is given by:*

$$\begin{aligned} u_n =& (1-\lambda)^{-n}e^{-n\tau Q}u_0 + \sum_{k=1}^{n}(1-\lambda)^{-k}e^{-(k-1)\tau Q}w \\ &+ \frac{\lambda}{1-\lambda}\sum_{k=1}^{n}(1-\lambda)^{-(n-k)}e^{-(n-k)\tau Q}\gamma_k \end{aligned} \tag{4.48}$$

Proof. We rearrange (4.47) as

$$(1-\lambda)u_{n+1} = e^{-\tau Q}u_n + w + \lambda\gamma_{n+1}.$$

We then check (4.48) inductively. The $n = 0$ case is trivial, and supposing (4.48) to be true for $n = m$ we have that

$$\begin{aligned} u_{m+1} &= (1-\lambda)^{-1}e^{-\tau Q}u_m + (1-\lambda)^{-1}w + \frac{\lambda}{1-\lambda}\gamma_{m+1} \\ &= (1-\lambda)^{-(m+1)}e^{-(m+1)\tau Q}u_0 + \sum_{k=1}^{m}(1-\lambda)^{-(k+1)}e^{-k\tau Q}w + (1-\lambda)^{-1}w \\ &\quad + \frac{\lambda}{1-\lambda}\sum_{k=1}^{m}(1-\lambda)^{-((m+1)-k)}e^{-((m+1)-k)\tau Q}\gamma_k + \frac{\lambda}{1-\lambda}\gamma_{m+1} \\ &= (1-\lambda)^{-(m+1)}e^{-(m+1)\tau Q}u_0 + \sum_{k=0}^{m}(1-\lambda)^{-(k+1)}e^{-k\tau Q}w \\ &\quad + \frac{\lambda}{1-\lambda}\sum_{k=1}^{m+1}(1-\lambda)^{-((m+1)-k)}e^{-((m+1)-k)\tau Q}\gamma_k \end{aligned}$$

completing the proof. □

We complete this set-up by exhibiting the asymptotics of (4.48). First, we define the limit relative to which we will consider asymptotics.

Definition 4.5.2. *We will consider 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 $\varepsilon > 0$. We will say, for real (matrix) valued g, that $g(\tau, n) = \mathcal{O}(\tau)$ if and only if $\operatorname{limsup} ||g(\tau, n)/\tau|| < \infty$ as $(\tau, n) \to (0, \infty)$ in $\{(\rho, m) \mid \rho > 0, m\rho - t \in [0, \rho)\}$ with the subspace topology induced by the standard topology on $(0, \infty) \times \mathbb{N}$. (Note that the choice of norm here is irrelevant since all norms on finite-dimensional real spaces are equivalent.)*

Theorem 4.5.3. *Let $t \geq 0$, $\varepsilon > 0$, $\mathcal{O}$ be as in Definition 4.5.2, $B := Q - \varepsilon^{-1} I$, F_t as in Definition 3.2.5, and $v := \varepsilon f - \frac{1}{2}\mathbf{1}$ in the fidelity forced case and $v := -\bar{u}\mathbf{1}$ in the mass-conserving case. Then:*

i. $(1-\lambda)^{-n} e^{-n\tau Q} u_0 = e^{-tB} u_0 + \mathcal{O}(\tau)$.

ii. $\sum_{k=1}^{n} (1-\lambda)^{-k} e^{-(k-1)\tau Q} w = \frac{1}{\varepsilon} F_t(B) v + \mathcal{O}(\tau)$.

iii. $\frac{\lambda}{1-\lambda} \sum_{k=1}^{n} (1-\lambda)^{-(n-k)} e^{-(n-k)\tau Q} \gamma_k = \lambda \sum_{k=1}^{n} e^{-(n-k)\tau B} \gamma_k + \mathcal{O}(\tau)$.

Hence by (4.48), the SDIE term obeys

$$u_n = e^{-tB} u_0 + \frac{1}{\varepsilon} F(B) v + \lambda \sum_{k=1}^{n} e^{-(n-k)\tau B} \gamma_k + \mathcal{O}(\tau). \tag{4.49}$$

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 $\mathcal{O}(\lambda)$, since $\lambda := \tau/\varepsilon$ and ε is fixed.

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

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

ii. We make the following claim:

$$\begin{aligned} \sum_{k=1}^{n} (1-\lambda)^{-k} e^{-(k-1)\tau Q} &= ((1-\lambda)I - e^{-\tau Q})^{-1} (I - (1-\lambda)^{-n} e^{-n\tau Q}) && (*) \\ &= ((1-\lambda)I - e^{-\tau Q})^{-1} (I - e^{-tB}) + \mathcal{O}(\tau) && (**) \\ &= ((1-\lambda)I - e^{-\tau Q})^{-1} B F_t(B) + \mathcal{O}(\tau) \end{aligned}$$

Towards showing $(*)$, note that if the sum is multiplied by $\mathcal{A} := (1-\lambda)I - e^{-\tau Q}$ then it telescopes to $I - (1-\lambda)^{-n} e^{-n\tau Q}$. Thus to show $(*)$, it suffices check

that as $\lambda \downarrow 0$, $\mathcal{A}$ is indeed invertible: let μ_k be the eigenvalues of Q and let $\mu_k' := \mu_k - \frac{1}{\varepsilon}$. Then $\mathcal{A}$ has eigenvalues:

$$1 - \lambda - e^{-\tau\mu_k} = 1 - \lambda - e^{-\lambda}e^{-\tau\mu_k'}$$

If $\mu_k' \leq 0$ then for $\lambda > 0$

$$e^{-\lambda}e^{-\tau\mu_k'} \geq e^{-\lambda} > 1 - \lambda$$

so the k^{th} eigenvalue of $\mathcal{A}$ is non-zero. If $\mu_k' > 0$ then $e^{-\varepsilon\mu_k'} < 1$ and $e(1-\lambda)^{1/\lambda} \to 1$ as $\lambda \downarrow 0$, so there exists $\lambda_k^* > 0$ such that for all $\lambda \in (0, \lambda_k^*)$

$$e(1-\lambda)^{1/\lambda} \in \left(e^{-\varepsilon\mu_k'}, 1\right]$$

and so for all such λ, the k^{th} eigenvalue of $\mathcal{A}$ is non-zero since $e^{\lambda}(1-\lambda) > e^{-\lambda\varepsilon\mu_k'}$. Hence for all $\lambda \in (0, \min_k \lambda_k^*)$, $\mathcal{A}$ is invertible.

To show $(**)$, note that by the proof of (i) it suffices to show that $\mathcal{A}^{-1}\mathcal{O}(\tau^2) = \mathcal{O}(\tau)$, i.e. that $\mathcal{A}\mathcal{O}(\tau) = \mathcal{O}(\tau^2)$. This follows since $\mathcal{A} = -\lambda I + (I - e^{-\tau Q})$, both terms of which are $\mathcal{O}(\tau)$.

Therefore, to show (ii) we seek to show that

$$BF_t(B)w = \frac{1}{\varepsilon}\mathcal{A}F_t(B)v + \mathcal{O}(\tau).$$

Noting that B and Q commute, and hence all the matrices here commute, it will therefore suffice to show that

$$\varepsilon Bw = \mathcal{A}v + \mathcal{O}(\tau).$$

Note that for all $x \in \mathcal{V}$

$$\varepsilon B\lambda x = \tau Q x - \lambda x$$

and so

$$\mathcal{A}x = -\lambda x + (I - e^{-\tau Q})x = \varepsilon B\lambda x + \mathcal{O}(\tau^2).$$

Finally, note that (since $F_\tau(A)f = \tau f + \mathcal{O}(\tau^2)$) $w = \lambda v + \mathcal{O}(\tau^2)$ in either case, so we have the desired result by the above.

iii. We consider the difference (recalling the bounds from Lemma 4.3.7)

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

as desired. □

4.5.2. Some functional analytic preamble

In the next subsection, we will consider the limit of (4.49). The key insight will be noticing that (4.49) strongly resembles a Riemann sum for the explicit integral form for the AC flows from Theorem 3.4.8. In this subsection, we will make some definitions to make this resemblance explicit, and prove some key convergence results.

Recalling that $B := Q - \varepsilon^{-1} I$, we define the piecewise constant function $z_\tau : [0, \infty) \to \mathcal{V}$,

$$z_\tau(s) := \begin{cases} e^{\tau B} \beta_1^{[\tau]}, & 0 \leq s \leq \tau, \\ e^{k\tau B} \beta_k^{[\tau]}, & (k-1)\tau < s \leq k\tau \text{ for } k \in \mathbb{N}, k \geq 2, \end{cases}$$

and the function

$$\zeta_\tau(s) := e^{-sB} z_\tau(s) = \begin{cases} e^{(\tau - s)B} \beta_1^{[\tau]}, & 0 \leq s \leq \tau, \\ e^{(k\tau - s)B} \beta_k^{[\tau]}, & (k-1)\tau < s \leq k\tau \text{ for } k \in \mathbb{N}, k \geq 2, \end{cases} \tag{4.50}$$

where the superscript $[\tau]$ is bookkeeping notation to keep track of the time-step governing u_n and β_n. Furthermore, define $\tilde{z}_\tau := z_\tau$ in the fidelity forced case and

$\tilde{z}_\tau := z_\tau - \overline{z_\tau}\mathbf{1}$ in the mass-conserving case. Finally, define $\tilde{\zeta}_\tau(s) := e^{-sB}\tilde{z}_\tau(s)$. We note a simple result.

Proposition 4.5.4.

$$\tilde{z}_\tau(s) = \begin{cases} e^{\tau B}\gamma_1^{[\tau]}, & 0 \le s \le \tau, \\ e^{k\tau B}\gamma_k^{[\tau]}, & (k-1)\tau < s \le k\tau \text{ for } k \in \mathbb{N}, k \ge 2, \end{cases}$$

and therefore

$$\tilde{\zeta}_\tau(s) = \begin{cases} e^{(\tau - s)B}\gamma_1^{[\tau]}, & 0 \le s \le \tau, \\ e^{(k\tau - s)B}\gamma_k^{[\tau]}, & (k-1)\tau < s \le k\tau \text{ for } k \in \mathbb{N}, k \ge 2. \end{cases}$$

4

Proof. In the fidelity forced case $\gamma = \beta$ so this is trivial. In the mass-conserving case since $B = \Delta - \frac{1}{\varepsilon}I$

$$\langle e^{k\tau B}\beta_k, \mathbf{1}\rangle_{\mathcal{V}} = \langle \beta_k, e^{k\tau B}\mathbf{1}\rangle_{\mathcal{V}} = e^{-k\tau/\varepsilon}\langle \beta_k, \mathbf{1}\rangle_{\mathcal{V}}$$

and therefore for $(k-1)\tau < s \le k\tau$

$$\overline{z_\tau(s)}\mathbf{1} = e^{k\tau B}\overline{\beta_k}\mathbf{1}$$

from which the result follows. The case for $k = 1$ is likewise. □

We note some important convergence results.

Proposition 4.5.5. *For any sequence $\tau_n' \to 0$[5] with $\tau_n' < \varepsilon$ for all n, there exists a function $z : [0,\infty) \to \mathcal{V}$ and a subsequence τ_n of τ_n' such that z_{τ_n} converges weakly to z in $L^2_{loc}([0,\infty);\mathcal{V})$ and z_{τ_n} weak*-converges to z in $L^\infty_{loc}([0,\infty);\mathcal{V})$. Furthermore, let $\tilde{z} := z$ in the fidelity forced case and $\tilde{z} := z - \bar{z}\mathbf{1}$ in the mass-conserving case. Then $\tilde{z}_{\tau_n}$ converges weakly to $\tilde{z}$ in $L^2_{loc}([0,\infty);\mathcal{V})$ and $\tilde{z}_{\tau_n}$ weak*-converges to $\tilde{z}$ in $L^\infty_{loc}([0,\infty);\mathcal{V})$.*

Proof. For $N \in \mathbb{N}$, consider $z_\tau|_{[0,N]}$. As the $\beta_k^{[\tau]} \in \mathcal{V}_{[-1,1]}$ for all k and τ by Lemma 4.3.7, we have for all $s \in [0,N]$ and $\tau < \varepsilon$

$$\left|\left|z_\tau(s)\right|\right|_{\mathcal{V}} \le \sup_{s' \in [0, N+\varepsilon]} \left|\left|e^{Bs'}\right|\right| \cdot ||\mathbf{1}||_{\mathcal{V}} \le e^{(N+\varepsilon)||B||} \cdot ||\mathbf{1}||_{\mathcal{V}}$$

where we have used that for $s \le N$ the corresponding $k\tau$ in the exponent of $z_\tau(s)$ is less than $N + \tau$, and that for $s' \ge 0$, $||e^{Bs'}|| \le e^{s'||B||}$. Therefore for $\tau < \varepsilon$ the $z_\tau|_{[0,N]}$ are uniformly bounded in s with respect to $||\cdot||_{\mathcal{V}}$ (and therefore with respect to $||\cdot||_\infty$, since all norms on $\mathcal{V}$ are equivalent), and hence they lie in a closed ball in $L^2([0,N];\mathcal{V})$ and in $L^\infty([0,N];\mathcal{V})$. By the Banach–Alaoglu theorem the former ball is weak-compact and the latter ball is weak*-compact. Hence for any $\tau_n' \downarrow 0$ there

[5] By this convergence, we just mean in the ordinary sense, not the sense of Definition 4.5.2. We are not fixing a t, and n is here just an index.

exists τ_n'' a subsequence of τ_n' and $z_1 \in L^2([0,N];\mathcal{V})$ and $z_2 \in L^\infty([0,N];\mathcal{V})$ such that

$$z_{\tau_n''}|_{[0,N]} \rightharpoonup z_1 \text{ in } L^2([0,N];\mathcal{V}),$$
$$z_{\tau_n''}|_{[0,N]} \rightharpoonup^* z_2 \text{ in } L^\infty([0,N];\mathcal{V}).$$

We claim that $z_1 = z_2$ a.e. on $[0,N]$. By the definitions of the weak and weak* topologies we have that for all $f \in L^2([0,N];\mathcal{V})$ and $g \in L^1([0,N];\mathcal{V})$

$$\int_0^N \langle z_{\tau_n''}(t), f(t)\rangle_{\mathcal{V}}\, dt \to \int_0^N \langle z_1(t), f(t)\rangle_{\mathcal{V}}\, dt,$$
$$\int_0^N \langle z_{\tau_n''}(t), g(t)\rangle_{\mathcal{V}}\, dt \to \int_0^N \langle z_2(t), g(t)\rangle_{\mathcal{V}}\, dt.$$

Hence for any $\mathcal{A} \subseteq [0,N]$ (measurable) and $i \in V$ define $f_{\mathcal{A},i}(t) := \chi_i$ if $t \in \mathcal{A}$ and $f_{\mathcal{A},i}(t) := \mathbf{0}$ otherwise. Then for all measurable $\mathcal{A} \subseteq [0,N]$, $f_{\mathcal{A},i} \in L^2([0,N];\mathcal{V}) \cap L^1([0,N];\mathcal{V})$ and so

$$\int_{\mathcal{A}} (z_1)_i(t) - (z_2)_i(t)\, dt = 0.$$

Hence $(z_1)_i = (z_2)_i$ a.e. for each $i \in V$, so $z_1 = z_2$ a.e. on $[0,N]$.

Next, we extend to $[0,\infty)$ by a "local-to-global" diagonal argument. First, we take $N = 1$: by the above argument we can choose a subsequence $\tau^{(1)}$ of τ' such that $z_{\tau_n^{(1)}}$ converges in both the weak topology on L^2 and the weak* topology on L^∞ to some z on $[0,1]$. Then to move from N to $N+1$ we likewise choose a subsequence $\tau^{(N+1)}$ of $\tau^{(N)}$ such that $z_{\tau_n^{(N+1)}}$ converges in both senses to z on $[0,N+1]$. Finally, define $\tau_n := \tau_n^{(n)}$. Then for all bounded $T \subseteq [0,\infty)$, we have $T \subseteq [0,M]$ for some $M \in \mathbb{N}$ and hence $z_{\tau_n}|_T$ is eventually a subsequence of $z_{\tau_n^{(M)}}|_T$ and so converges in both senses to $z|_T$.

Finally, both the identity map and the map $f \mapsto f - \bar{f}\mathbf{1}$ are continuous with respect to both topologies, so the result about $\tilde{z}$ immediately follows. □

Corollary 4.5.6. *From $z_{\tau_n} \rightharpoonup z$ (defined in Proposition 4.5.5) in $L^2_{loc}([0,\infty);\mathcal{V})$ we infer*:

A. $\zeta_{\tau_n} \rightharpoonup \zeta$ and $\tilde{\zeta}_{\tau_n} \rightharpoonup \tilde{\zeta}$, where $\zeta(s) := e^{-sB}z$ and $\tilde{\zeta}(s) := e^{-sB}\tilde{z}$ (for $\tilde{z}$ defined in Proposition 4.5.5), both in $L^2_{loc}([0,\infty);\mathcal{V})$.

B. For all $t \geq 0$,

$$\int_0^t z_{\tau_n}(s)\, ds \to \int_0^t z(s)\, ds \qquad \text{and} \qquad \int_0^t \tilde{z}_{\tau_n}(s)\, ds \to \int_0^t \tilde{z}(s)\, ds.$$

C. Replacing τ_n by an appropriate subsequence, we have strong convergence of

4

the Cesàro sums, i.e. for all bounded $T \subseteq [0,\infty)$

$$\frac{1}{N}\sum_{n=1}^{N} z_{\tau_n} \to z, \qquad \frac{1}{N}\sum_{n=1}^{N} \zeta_{\tau_n} \to \zeta,$$

$$\frac{1}{N}\sum_{n=1}^{N} \tilde{z}_{\tau_n} \to z, \qquad \frac{1}{N}\sum_{n=1}^{N} \tilde{\zeta}_{\tau_n} \to \zeta,$$

in $L^2(T;\mathcal{V})$ *as* $N \to \infty$.

And from $z_{\tau_n} \rightharpoonup^* z$ *in* $L^\infty_{loc}([0,\infty);\mathcal{V})$ *we infer*:

D. $\zeta_{\tau_n} \rightharpoonup^* \zeta$ *and* $\tilde{\zeta}_{\tau_n} \rightharpoonup^* \tilde{\zeta}$ *in* $L^\infty_{loc}([0,\infty);\mathcal{V})$.

Proof. All results regarding $\tilde{z}$ or $\tilde{\zeta}$ follow from the corresponding result regarding z or ζ since both the identity map and the map $f \mapsto f - \bar{f}\mathbf{1}$ are continuous with respect to all of these topologies. We therefore prove just the z or ζ parts.

Claim (A) follows since $f \mapsto e^{-sB}f$ (where s is the argument of f) is a continuous self-adjoint map on $L^2(T;\mathcal{V})$ for T bounded. Hence for all $f \in L^2(T;\mathcal{V})$,

$$(\zeta_{\tau_n}, f)_{s\in T} = (z_{\tau_n}, e^{-sB}f)_{s\in T} \to (z, e^{-sB}f)_{s\in T} = (\zeta, f)_{s\in T}. \qquad (*)$$

Claim (B) is a direct consequence of weak convergence. Claim (C) follows by the Banach–Saks theorem [1], which states that weak L^p convergence on a bounded interval entails strong convergence of Cesàro sums on that interval along an appropriate subsequence, and a "local-to-global" diagonal argument as in the above proof to extract a subsequence that works on all of $[0,\infty)$. Claim (D) follows since $f \mapsto e^{-sB}f$ is continuous on $L^\infty(T;\mathcal{V})$ and on $L^1(T;\mathcal{V})$, for T bounded, and for all $f \in L^\infty(T;\mathcal{V})$ and $g \in L^1(T;\mathcal{V})$

$$\int_T \langle e^{-sB}f(s), g(s)\rangle_{\mathcal{V}}\, ds = \int_T \langle f(s), e^{-sB}g(s)\rangle_{\mathcal{V}}\, ds$$

so the map is "self-adjoint" with respect to the pairing of L^∞ with L^1, and so (D) follows by the same argument as $(*)$. □

4.5.3. Convergence of the SDIE trajectories

We now turn to the question of convergence of the SDIE trajectories. Fixing $t \in [0,\infty)$ and $\varepsilon > 0$, we consider the limit as $\tau \downarrow 0$, $n \to \infty$ as in Definition 4.5.2. Taking τ to zero along the sequence τ_n from Proposition 4.5.5, we define $m_n(t) := \lceil t/\tau_n \rceil$, so that $m_n(t) \to \infty$ as $n \to \infty$, and $m_n(t)\tau_n - t \in [0,\tau_n)$. Thus $\tau_n, m_n(t)$ satisfy the conditions of the limit in Definition 4.5.2 sequentially. To reduce the clutter in our notation, we shall henceforth abbreviate $m_n(t)$ by m.

We thus define (for any $t \geq 0$) the pointwise limit of the SDIE trajectories

$$\hat{u}(t) := \lim_{n\to\infty} u_m^{\lceil \tau_n \rceil} \qquad (4.51)$$

when this limit exists (which we shall prove it does for all $t \geq 0$). By (4.49), we can rewrite this as:

$$\hat{u}(t) = e^{-tB}u_0 + \frac{1}{\varepsilon}F_t(B)v + \lim_{n\to\infty} \frac{\tau_n}{\varepsilon} \sum_{k=1}^{m} e^{-(m-k)\tau_n B}\gamma_k^{[\tau_n]}$$
$$= e^{-tB}u_0 + \frac{1}{\varepsilon}F_t(B)v + \frac{1}{\varepsilon} \lim_{n\to\infty} e^{-m\tau_n B} \int_0^{m\tau_n} \tilde{z}_{\tau_n}(s)\, ds.$$

We seek to show that the pair $(\hat{u}, \zeta)$ (where ζ is as in Corollary 4.5.6) solves (3.22) or (3.23) in the respective cases, i.e. $\hat{u}$ is an AC flow trajectory. We will do this by checking that $(\hat{u}, \zeta)$ satisfy the sufficient conditions given for (u, β) in Theorem 3.4.8. We will split this into two lemmas. First, we show all but one of the required conditions.

Proposition 4.5.7. *The pair $(\hat{u}, \zeta)$ obeys:*

(i) *For all $t \geq 0$, $\hat{u}(t)$ exists and is given by*

$$\hat{u}(t) = e^{-tB}u_0 + \frac{1}{\varepsilon}F_t(B)v + \frac{1}{\varepsilon}\int_0^t e^{-(t-s)B}\tilde{\zeta}(s)\, ds, \tag{4.52}$$

where v is as in Theorem 4.5.3 and $\tilde{\zeta}$ is as in Corollary 4.5.6.

(ii) *$\hat{u}(t) \in \mathcal{V}_{[0,1]}$ for all $t \geq 0$.*

(iii) *$\tilde{\zeta}$ is locally essentially bounded and locally integrable.*

Note 23. *By our choice of notation, these cover both cases of Theorem 3.4.8.*

Proof. (i) Note that $m\tau_n =: t + \eta_n$ where $\eta_n \in [0, \tau_n)$. Therefore

$$\lim_{n\to\infty} e^{-m\tau_n B} \int_0^{m\tau_n} \tilde{z}_{\tau_n}(s)\, ds$$
$$= \lim_{n\to\infty} e^{-\eta_n B}e^{-tB} \int_0^t \tilde{z}_{\tau_n}(s)\, ds + e^{-\eta_n B}e^{-tB} \int_t^{t+\eta_n} \tilde{z}_{\tau_n}(s)\, ds$$
$$= \lim_{n\to\infty} e^{-tB} \int_0^t \tilde{z}_{\tau_n}(s)\, ds + e^{-tB} \int_t^{t+\eta_n} \tilde{z}_{\tau_n}(s)\, ds \qquad (*)$$
$$= \lim_{n\to\infty} e^{-tB} \int_0^t \tilde{z}_{\tau_n}(s)\, ds \qquad (**)$$
$$= e^{-tB} \int_0^t \tilde{z}(s)\, ds \quad \text{by Corollary 4.5.6(B).}$$

To show line $(*)$, note that by the proof of Proposition 4.5.5, the $\tilde{z}_{\tau_n}|_{[0,\lfloor t+1 \rfloor]}$ lie in the continuous image of a closed ball in L^∞, so are uniformly bounded in n with respect to $||\cdot||_{\infty, t\in[0,\lfloor t+1 \rfloor]}$, and therefore the integral of $\tilde{z}_{\tau_n}$ over $[0, t]$ is bounded and over $[t, t + \eta_n]$ is $\mathcal{O}(\tau_n)$. Therefore, because $e^{-\eta_n B} =$

$I+\mathcal{O}(\tau_n)$, line $(*)$ follows. Line $(**)$ follows because $\tilde{z}_{\tau_n}(s)$ is bounded on $[t, t+\max_{n'} \eta_{n'}]$ uniformly in n by Proposition 4.5.5. Then (4.52) is an immediate consequence of the final line.

(ii) $\hat{u}(t)$ is a limit of SDIE iterates, each of which lies in $\mathcal{V}_{[0,1]}$.

(iii) Since, by Corollary 4.5.6(A), $\tilde{\zeta}$ is a weak limit of locally bounded and locally integrable functions, it must be locally essentially bounded and locally integrable.

□

Lastly, we check the subdifferential condition from Theorem 3.4.8.

Lemma 4.5.8. *$\zeta(t) \in \mathcal{B}(\hat{u}(t))$ for a.e. $t \geq 0$.*

We give two proofs of this result.

Proof (A). By Corollary 4.5.6(C), on each bounded $T \subseteq [0,\infty)$ ζ is the $L^2(T;\mathcal{V})$ limit of

$$S_N := \frac{1}{N}\sum_{n=1}^{N} \zeta_{\tau_n}$$

as $N \to \infty$. As L^2 convergence implies a.e. pointwise convergence along a subsequence, by a "local-to-global" diagonal argument there exists a sequence $N_k \to \infty$ such that for a.e. $t \geq 0$

$$\zeta(t) = \lim_{k\to\infty} \frac{1}{N_k}\sum_{n=1}^{N_k} \zeta_{\tau_n}(t).$$

Let $\eta_n := m\tau_n - t \in [0, \tau_n)$. Then by Lemma 4.3.7

$$\begin{aligned} ||\zeta_{\tau_n}(t) - \beta_m^{[\tau_n]}||_{\mathcal{V}} &= ||(e^{\eta_n B} - I)\beta_m^{[\tau_n]}||_{\mathcal{V}} \text{ by (4.50)} \\ &\leq \left(1 - e^{-\eta_n ||B||}\right) ||\mathbf{1}||_{\mathcal{V}} \\ &< (1 - e^{-\tau_n ||B||})||\mathbf{1}||_{\mathcal{V}} \\ &< \tau_n ||B||\, ||\mathbf{1}||_{\mathcal{V}}. \end{aligned} \tag{4.53}$$

Therefore for a.e. $t \geq 0$,

$$\left\| \zeta(t) - \frac{1}{N_k}\sum_{n=1}^{N_k} \beta_m^{[\tau_n]} \right\|_{\mathcal{V}} \leq \left\| \zeta(t) - \frac{1}{N_k}\sum_{n=1}^{N_k} \zeta_{\tau_n}(t) \right\|_{\mathcal{V}} + ||B||\, ||\mathbf{1}||_{\mathcal{V}} \frac{1}{N_k}\sum_{n=1}^{N_k} \tau_n \to 0$$

as $k \to \infty$ (since $\tau_n \to 0$ and the convergence of a sequence implies the convergence of its Cesàro sums to the same limit), so for a.e. $t \geq 0$

$$\zeta(t) = \lim_{k\to\infty} \frac{1}{N_k}\sum_{n=1}^{N_k} \beta_m^{[\tau_n]}. \tag{4.54}$$

Recall that as $n \to \infty$, $u_m^{[\tau_n]} \to \hat{u}(t)$ and $\beta_m^{[\tau_n]} \in \mathcal{B}\left(u_m^{[\tau_n]}\right)$. Suppose first that $\hat{u}_i(t) \in (0,1)$. Then we have some M such that for all $n > M$, $(u_m^{[\tau_n]})_i \in (0,1)$ and so $(\beta_m^{[\tau_n]})_i = 0$. Hence

$$\zeta_i(t) = \lim_{k\to\infty} \frac{1}{N_k}\left(\sum_{n=1}^{M} (\beta_m^{[\tau_n]})_i + \sum_{n=M+1}^{N_k} 0\right) = 0$$

as desired. Next suppose $\hat{u}_i(t) = 0$. Then we have some M such that for all $n > M$, $(u_n^{[\tau_n]})_i \in [0,1)$ and so $(\beta_n^{[\tau_n]})_i \geq 0$. Hence

$$\zeta_i(t) \geq \lim_{k\to\infty} \frac{1}{N_k}\left(\sum_{n=1}^{M} (\beta_m^{[\tau_n]})_i + \sum_{n=M+1}^{N_k} 0\right) = 0$$

as desired. Likewise for $\hat{u}_i(t) = 1$, $\zeta_i(t) \leq 0$. Hence we have $\zeta(t) \in \mathcal{B}(\hat{u}(t))$. □

Proof (B). Fix $i \in V$ and bounded $T \subseteq [0,\infty)$. For tidiness of notation, we define $x_n(t) := u_{\lfloor t/\tau_n \rfloor}^{[\tau_n]}$ and $x(t) := \hat{u}_i(t)$, and likewise $\xi_n(t) := (\beta_{\lfloor t/\tau_n \rfloor}^{[\tau_n]})_i$ and $\xi(t) := \zeta_i(t)$. Let

$$T_1 := \{t \in T \mid x(t) = 0\}, \quad T_2 := \{t \in T \mid x(t) \in (0,1)\}, \quad T_3 := \{t \in T \mid x(t) = 1\}.$$

Then it suffices to show that $\xi \geq 0$ a.e. on T_1, $\xi = 0$ a.e. on T_2, and $\xi \leq 0$ a.e. on T_3.

By Corollary 4.5.6(D) we have that $(\zeta_{\tau_n})_i \rightharpoonup^* \xi$ in $L^\infty_{loc}([0,\infty); \mathcal{V})$ and therefore $(\zeta_{\tau_n})_i \rightharpoonup^* \xi$ in $L^\infty(T, \mathbb{R})$, i.e. for all $f \in L^1(T, \mathbb{R})$, as $n \to \infty$

$$\int_T (\zeta_{\tau_n})_i(t) f(t)\, dt \to \int_T \xi(t) f(t)\, dt.$$

It follows from (4.53) that $(\zeta_{\tau_n})_i(t) = \xi_n(t) + \mathcal{O}(\tau_n)$, and so we infer that as $n \to \infty$

$$\int_T \xi_n(t) f(t)\, dt \to \int_T \xi(t) f(t)\, dt.$$

By (4.51) we have by definition that for all $t \in T_1$, $x_n(t) \to 0$. We define the increasing sequence of (measurable) sets $A_N := \{t \in T_1 \mid \forall n \geq N\ x_n(t) < 1/2\}$. Then by the pointwise convergence of the x_n, $T_1 = \bigcup_N A_N$. Suppose for contradiction that for some $X \subseteq T_1$ of positive measure, $\xi < 0$ on X. So there exists $\delta > 0$ and $Y \subseteq X$ of positive measure such that $\xi \leq -\delta$ on Y. As T_1 is the union of the A_N there exists $N \in \mathbb{N}$ such that $Y \cap A_N$ is of positive measure. Taking test function $f(t) = 1$ if $t \in \chi_{Y \cap A_N}$ and $f(t) = 0$ otherwise, we infer that as $n \to \infty$ (and μ the Lebesgue measure)

$$\int_{Y\cap A_N} \xi_n(t)\, dt \to \int_{Y\cap A_N} \xi(t)\, dt \leq -\delta\mu(Y \cap A_N) < 0$$

but since $\beta^{[\tau_n]}_{[t/\tau_n]} \in \mathcal{B}(u^{[\tau_n]}_{[t/\tau_n]})$ we have that if $t \in A_N$ then for all $n \geq N$, $\xi_n(t) \geq 0$, so this is a contradiction. Hence $\xi \geq 0$ a.e. on T_1. By the same argument, $\xi \leq 0$ a.e. on T_3.

Finally, for all $t \in T_2$, since $x_n(t) \to x(t)$, $x_n(t)$ is eventually in $(0,1)$. Define $B_N := \{t \in T_2 \mid \forall n \geq N\ x_n(t) \in (0,1)\}$, and note that $T_2 = \bigcup_N B_N$ and that for $t \in B_N$ and $n \geq N$, $\xi_n(t) = 0$ since $\beta^{[\tau_n]}_{[t/\tau_n]} \in \mathcal{B}(u^{[\tau_n]}_{[t/\tau_n]})$. Suppose for contradiction that for some $X \subseteq T_2$ of positive measure, $\xi \neq 0$ on X. Then WLOG there exists $\delta > 0$ and $Y \subseteq X$ of positive measure such that $\xi \geq \delta$ on Y. As before there exists $N \in \mathbb{N}$ such that $Y \cap B_N$ is of positive measure. Taking $f(t) = 1$ if $t \in \chi_{Y \cap B_N}$ and $f(t) = 0$ otherwise, we infer that as $n \to \infty$ (for $n \geq N$)

$$0 = \int_{Y \cap B_N} \xi_n(t)\, dt \to \int_{Y \cap B_N} \xi(t)\, dt \geq \delta \mu(Y \cap A_N) > 0$$

a contradiction. Therefore $\xi = 0$ a.e. on T_2. □

Note 24. *We thank Dr Carolin Kreisbeck for her suggestion of using weak* L^∞ convergence which led to the development of proof (B).*

Therefore, by Theorem 3.4.8, we have the following convergence result.

Theorem 4.5.9. *For any given $u_0 \in \mathcal{V}_{[0,1]} \setminus \{\mathbf{0}, \mathbf{1}\}$, $\varepsilon > 0$ and $\tau_n \downarrow 0$, there exists a subsequence τ'_n of τ_n with $\tau'_n < \varepsilon$ for all n, such that along this subsequence the semi-discrete iterates $(u^{[\tau'_n]}_m, \beta^{[\tau'_n]}_m)$ given by (4.1) (respectively (4.2)) with initial state u_0 converge to an AC solution. That is, there exists $(\hat{u}, \zeta)$ a solution to (3.22) (respectively (3.23)) with $\hat{u}(0) = u_0$, such that:*

a. $\beta^{[\tau'_n]}_{[\cdot/\tau'_n]}$ converges to ζ weakly in $L^2_{loc}([0,\infty); \mathcal{V})$ and weakly in $L^\infty_{loc}([0,\infty); \mathcal{V})$,*

b. for each $t \geq 0$ and for $m := \lceil t/\tau'_n \rceil$, $u^{[\tau'_n]}_m \to \hat{u}(t)$ as $n \to \infty$, and

c. there is a sequence $N_k \to \infty$ such that for almost every $t \geq 0$, $\frac{1}{N_k} \sum_{n=1}^{N_k} \beta^{[\tau'_n]}_m \to \zeta(t)$ as $k \to \infty$.

Note 25. *This result proves Theorem 3.4.10, i.e. the existence of AC solutions.*

Note 26. *For $u_0 = \mathbf{0}$ or $\mathbf{1}$, the $u^{[\tau_n]}_m \equiv u_0$ trivially converge but the $\beta^{[\tau_n]}_m$ need not converge.*

By the use of a basic topological spaces fact, we can eliminate the need to pass to a subsequence for the convergence of the $u^{[\tau_n]}_m$ to $\hat{u}$.

Corollary 4.5.10. *Let $u_0 \in \mathcal{V}_{[0,1]}$, $\varepsilon > 0$, and $\tau_n \downarrow 0$ with $\tau_n < \varepsilon$ for all n. Then for each $t \geq 0$, as $n \to \infty$, $u^{[\tau_n]}_{[t/\tau_n]} \to \hat{u}(t)$.*

Proof. Consider the following fact about topological spaces: if (X, ρ) is a topological space, $x_n, x \in X$, and every subsequence of x_n has a further subsequence converging to x in ρ, then $x_n \to x$ in ρ.[6]

Let $x_n := t \mapsto u^{[\tau_n]}_{[t/\tau_n]} \in (\mathcal{V}_{t\in[0,\infty)}, \rho)$ for ρ the topology of pointwise convergence, and let τ_{n_k} be any subsequence of τ_n. Then by the theorem there is a further subsequence $\tau_{n_{k_l}}$ such that $x_{n_{k_l}} \to \tilde{u}$ pointwise where $\tilde{u}$ is an AC solution with initial condition $\tilde{u}(0) = u_0$. By Theorem 3.4.9 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 further convergent subsequence with limit x, and hence by the above fact, $x_n \to x$ pointwise. □

4.5.4. Consequences of Theorem 4.5.9

Given the representation from Theorem 4.5.9 of the unique solutions to (3.22) and (3.23) as limits of SDIE approximations, we can deduce a number of properties of these AC flows.

We first verify that these flows monotonically decrease their respective Ginzburg–Landau energies, by considering the Lyapunov functionals for the SDIE schemes.

Proposition 4.5.11. *Let* $H_\tau(u) := \frac{1}{2\tau}H(u)$ *and* $H_{0,\tau}(u) := \frac{1}{2\tau}H_0(u)$ *. Then for* $u \in \mathcal{V}_{[0,1]}$

$$H_\tau(u) = \mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u) - \frac{1}{2}\langle \tilde{f}, M\tilde{f}\rangle_{\mathcal{V}} + \frac{1}{2}\tau\langle u, Q_\tau(Au - 2f)\rangle_{\mathcal{V}}$$
$$H_{0,\tau}(u) = \mathrm{GL}_\varepsilon(u) + \frac{1}{2}\tau\langle u, Q'_\tau u\rangle_{\mathcal{V}}$$

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

Proof. Note that GL_ε and H_0 are the $\mu = \mathbf{0}$ (and hence $f = \mathbf{0}$ and $A = \Delta$) cases of $\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}$ and H, and that in that case $Q'_\tau = \Delta Q_\tau = Q_\tau\Delta$. Hence it suffices to prove the theorem in the H cases.

Expanding and collecting terms in (3.9), we find that for $u \in \mathcal{V}_{[0,1]}$

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

Then by (4.40b) and recalling that $\lambda := \tau/\varepsilon$

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

[6] Suppose $x_n \nrightarrow x$. Then there exists $U \in \rho$ 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 subsequence converging to x.

To show the uniform convergence, note that (since $||A||$ is finite by Proposition 3.2.4) $||u||_{\mathcal{V}}$ and $||Au - 2f||_{\mathcal{V}}$ 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 (and thus $\xi_k \geq 0$ by Proposition 3.2.4), then Q_τ has corresponding eigenvalue

$$\tau^{-2}((1 - e^{-\tau\xi_k})/\xi_k - \tau) = \frac{1}{\tau^2 \xi_k}(1 - \tau\xi_k - e^{-\tau\xi_k}) \in \left[-\frac{1}{2}\xi_k, 0\right]$$

so $||Q_\tau|| \leq \frac{1}{2}||A||$.

Finally, it suffices to show that $H_\tau(u_\tau) - H_\tau(u) \to 0$, since

$$H_\tau(u_\tau) + \frac{1}{2}\langle \tilde{f}, M\tilde{f}\rangle_{\mathcal{V}} - \mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u) = H_\tau(u_\tau) - H_\tau(u) + \left(H_\tau(u) + \frac{1}{2}\langle \tilde{f}, M\tilde{f}\rangle_{\mathcal{V}} - \mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u)\right)$$

and the bracketed term converges to zero. But by the above expression for H_τ and $\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u)$ (and since $A(I + \tau Q_\tau)$ is self-adjoint)

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

which converges to zero since the right-hand entry in the inner product is bounded uniformly in τ. □

Theorem 4.5.12. *The fidelity forced AC trajectory u defined by Definition 3.4.1 has $\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u(t))$ monotonically decreasing in t. More precisely: for all $t > s \geq 0$,*

$$\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u(s)) - \mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u(t)) \geq \frac{1}{2(t-s)}\left|\left|u(s) - u(t)\right|\right|_{\mathcal{V}}^2. \tag{4.55}$$

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

$$\left|\left|u(s) - u(t)\right|\right|_{\mathcal{V}} \leq \sqrt{|t-s|}\sqrt{2\,\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u(0))}. \tag{4.56}$$

The mass-conserving AC trajectory u defined by Definition 3.4.2 has $\mathrm{GL}_\varepsilon(u(t))$ monotonically decreasing in t. More precisely: for all $t > s \geq 0$,

$$\mathrm{GL}_\varepsilon(u(s)) - \mathrm{GL}_\varepsilon(u(t)) \geq \frac{1}{2(t-s)}\left|\left|u(s) - u(t)\right|\right|_{\mathcal{V}}^2 \tag{4.57}$$

and therefore,

$$\left|\left|u(s) - u(t)\right|\right|_{\mathcal{V}} \leq \sqrt{|t-s|}\sqrt{2\,\mathrm{GL}_\varepsilon(u(0))}. \tag{4.58}$$

Proof. It suffices to prove the former case, the latter is likewise.

Let $t > s \geq 0$, $\ell := \lceil s/\tau_n \rceil$, and $m := \lceil t/\tau_n \rceil$. Note that therefore $\ell \leq m$. Next, note a standard inner product space fact: for all sequences $v_n \in \mathcal{V}$,

$$\sum_{n=1}^{N} ||v_n||_{\mathcal{V}}^2 = \frac{1}{N} \left\| \sum_{n=1}^{N} v_n \right\|_{\mathcal{V}}^2 + \frac{1}{N} \sum_{k<n} ||v_n - v_k||_{\mathcal{V}}^2 \geq \frac{1}{N} \left\| \sum_{n=1}^{N} v_n \right\|_{\mathcal{V}}^2, \tag{4.59}$$

which can be checked by expanding the inner products and collecting terms. Now by Theorems 3.4.9 (i.e., uniqueness of AC solutions) and 4.5.9, we have sequences $u_\ell^{[\tau_n]} \to u(s)$ and $u_m^{[\tau_n]} \to u(t)$. It follows that:

$$\begin{aligned}
&\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u(s)) - \mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u(t)) \\
&\quad = \lim_{n\to\infty} H_{\tau_n}\left(u_\ell^{[\tau_n]}\right) - H_{\tau_n}\left(u_m^{[\tau_n]}\right) && \text{by Proposition 4.5.11} \\
&\quad = \lim_{n\to\infty} \sum_{k=\ell}^{m-1} H_{\tau_n}\left(u_k^{[\tau_n]}\right) - H_{\tau_n}\left(u_{k+1}^{[\tau_n]}\right) \\
&\quad \geq \lim_{n\to\infty} \frac{1}{2\tau_n}\left(1 - \frac{\tau_n}{\varepsilon}\right) \sum_{k=\ell}^{m-1} \left\| u_{k+1}^{[\tau_n]} - u_k^{[\tau_n]} \right\|_{\mathcal{V}}^2 && \text{by (4.41)} \\
&\quad \geq \lim_{n\to\infty} \frac{1}{2\tau_n}\left(1 - \frac{\tau_n}{\varepsilon}\right) \frac{1}{m-\ell} \left\| u_m^{[\tau_n]} - u_\ell^{[\tau_n]} \right\|_{\mathcal{V}}^2 && \text{by (4.59)} \\
&\quad = \frac{1}{2(t-s)} \left\| u(s) - u(t) \right\|_{\mathcal{V}}^2 \geq 0
\end{aligned}$$

as desired, since $\tau_n(m - \ell) \to t - s$.

Finally, since $\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u) \geq 0$ for all $u \in \mathcal{V}$, it follows that

$$\begin{aligned}
\left\| u(s) - u(t) \right\|_{\mathcal{V}}^2 &\leq 2(t-s)(\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u(s)) - \mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u(t))) \\
&\leq 2(t-s)\,\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u(s)) \leq 2(t-s)\,\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u(0)).
\end{aligned}$$

□

Next, we prove the well-posedness of the fidelity forced AC flow.

Note 27. *We have yet to determine if the mass-conserving AC flow (3.23) is well-posed.*

Theorem 4.5.13. *Let $u_0, v_0 \in \mathcal{V}_{[0,1]}$ define fidelity forced AC trajectories u, v by Definition 3.4.1. Then, if $\xi_1 := \min \sigma(A)$, then*

$$||u(t) - v(t)||_{\mathcal{V}} \leq e^{-\xi_1 t} e^{t/\varepsilon} ||u_0 - v_0||_{\mathcal{V}}. \tag{4.60}$$

Proof. Fix $t \geq 0$ and let $m := \lceil t/\tau_n \rceil$. By Corollary 4.5.10, 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 (4.8):

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

and taking $n \to \infty$ gives (4.60). □

Finally, we derive some bounds on the ζ from Corollary 4.5.6 and thereby infer a Lipschitz condition on AC trajectories.

Lemma 4.5.14. *Let $\zeta(t)$ (as defined in Corollary 4.5.6) be given at a.e. $t \geq 0$ by (4.54), and let t be any such t.*

In the fidelity forced case

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

and in the mass-conserving case

$$\zeta(t) - \overline{\zeta(t)}\mathbf{1} \in \mathcal{V}_{[\bar{u}-1,\bar{u}]} \text{ and } \zeta(t) \in \mathcal{V}_{[-1,1]}.$$

Proof. The bounds on $\zeta(t)$ follow immediately from (4.54) and the bounds on β_{n+1} in Lemma 4.3.7. Finally, since $v \mapsto v - \bar{v}\mathbf{1}$ is linear and continuous, by (4.54)

$$\zeta(t) - \overline{\zeta(t)}\mathbf{1} = \lim_{k\to\infty} \frac{1}{N_k} \sum_{n=1}^{N_k} \left(\beta_m^{[\tau_n]} - \overline{\beta_m^{[\tau_n]}}\mathbf{1}\right).$$

and thus the bounds on $\zeta(t) - \overline{\zeta(t)}\mathbf{1}$ follow from (4.38). □

Theorem 4.5.15. *In both the fidelity forced and mass-conserving cases, the AC solution $(\hat{u}, \zeta)$ given by Theorem 4.5.9 has $\hat{u} \in C^{0,1}([0,\infty); \mathcal{V})$.*

Proof. Recall from (3.36) (in the proof of Theorem 3.4.8) that for $B := A - \frac{1}{\varepsilon}I$ or $B := \Delta - \frac{1}{\varepsilon}I$, $v := \varepsilon f - \frac{1}{2}\mathbf{1}$ or $v := -\bar{u}\mathbf{1}$, and $\gamma := \zeta$ or $\gamma := \zeta - \bar{\zeta}\mathbf{1}$, we have that since $(\hat{u}, \zeta)$ is an AC solution,

$$\hat{u}(t) = e^{-tB}\hat{u}(0) + \frac{1}{\varepsilon}F_t(B)v + \frac{1}{\varepsilon}\int_0^t e^{(s-t)B}\gamma(s)\,ds$$

and by Lemma 4.5.14, $\gamma(s) \in \mathcal{V}_{[-1,1]}$ at a.e. s in either case. Let $0 \leq t_1 < t_2$. Since $(\hat{u}, \zeta)$ solves either (3.22) or (3.23), and both of these equations are time-translation invariant, it follows that

$$\begin{aligned}\hat{u}(t_2) - \hat{u}(t_1) &= \left(e^{-(t_2-t_1)B} - I\right)\hat{u}(t_1) + \frac{1}{\varepsilon}F_{t_2-t_1}(Q)v + \frac{1}{\varepsilon}\int_0^{t_2-t_1} e^{(s-(t_2-t_1))B}\gamma(s)\,ds \\ &= \left(e^{-(t_2-t_1)B} - I\right)\hat{u}(t_1) + \frac{1}{\varepsilon}F_{t_2-t_1}(Q)v + \frac{1}{\varepsilon}\int_0^{t_2-t_1} e^{-sB}\gamma(t_2-t_1-s)\,ds.\end{aligned}$$

Let $B_s := (e^{-sB} - I)/s$ for $s > 0$. Then B_s commutes with B, is self-adjoint, and as $-B$ has largest eigenvalue less than or equal to $1/\varepsilon$ (by Proposition 3.2.4), we have $||B_s|| < (e^{s/\varepsilon} - 1)/s$, with RHS monotonically increasing in s for $s > 0$.[7] Furthermore, if η is an eigenvalue of B (and hence in $[-1/\varepsilon, ||A|| - 1/\varepsilon]$), then by

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

Definition 3.2.5 $F_s(Q)/s$ has corresponding eigenvalue $(1-e^{-s\eta})/s\eta$ (or 1 if $\eta = 0$). If $\eta \geq 0$, then this term is positive and less than or equal to 1. If $\eta < 0$, then the term has absolute value $(e^{s|\eta|}-1)/s|\eta|$, which is bounded above by $\varepsilon(e^{s/\varepsilon}-1)/s$ since $|\eta| \leq 1/\varepsilon$ in the $\eta < 0$ case, and $(e^{sx}-1)/sx$ is monotonically increasing in x for $x > 0$. Since that latter bound is greater than or equal to 1, we have that $||F_s(Q)/s|| \leq \varepsilon(e^{s/\varepsilon}-1)/s$ which is monotonically increasing in s.

Since v is constant, $\hat{u}$ is uniformly bounded in time, and γ is uniformly essentially bounded, let C be such that $||v||_\mathcal{V}, ||\hat{u}(t)||_\mathcal{V}, ||\gamma(s)||_\mathcal{V} < C$ for all $t \geq 0$ and a.e. $s \geq 0$. Then we have for $t_2 - t_1 < 1$

$$\begin{aligned}\frac{\left|\left|\hat{u}(t_2)-\hat{u}(t_1)\right|\right|_\mathcal{V}}{t_2-t_1} &\leq C||B_{t_2-t_1}|| + C\frac{1}{\varepsilon}||F_{t_2-t_1}(Q)/(t_2-t_1)|| + C\frac{1}{\varepsilon}\max_{s\in[0,t_2-t_1]}||e^{-sQ}|| \\ &\leq C\frac{e^{(t_2-t_1)/\varepsilon}-1}{t_2-t_1} + C\frac{e^{(t_2-t_1)/\varepsilon}-1}{t_2-t_1} + C\frac{1}{\varepsilon}e^{(t_2-t_1)/\varepsilon} \\ &\leq 2C(e^{1/\varepsilon}-1) + C\frac{1}{\varepsilon}e^{1/\varepsilon}\end{aligned}$$

with the last line since $\varepsilon(e^{s/\varepsilon}-1)/s$ is monotonically increasing in s, and for $t_2-t_1 \geq 1$ we simply have

$$\frac{\left|\left|\hat{u}(t_2)-\hat{u}(t_1)\right|\right|_\mathcal{V}}{t_2-t_1} \leq \left|\left|\hat{u}(t_2)-\hat{u}(t_1)\right|\right|_\mathcal{V} \leq ||\mathbf{1}||_\mathcal{V}$$

completing the proof. □

4.6. Conclusions and future work

In the previous chapter, we defined Allen–Cahn (AC) flow and the Merriman–Bence–Osher (MBO) scheme in a graph setting, and in particular defined a graph AC flow against the double-obstacle potential, including under either fidelity forcing or mass conservation constraints. We have proved that this double-obstacle AC flow, despite its definition requiring the introduction of subdifferential terms, has various desirable properties (including under either constraint). The first wave of these were proved in the last chapter direct from the definition, and included an explicit integral form and uniqueness of solutions. The second wave of these, including existence and Lipschitz regularity of solutions, and monotonic decrease of the Ginzburg–Landau energy along solutions, we proved in this chapter, by characterising the solutions as the limit of trajectories of the SDIE scheme for AC flow. Furthermore, we have demonstrated that the MBO scheme is a special case of this SDIE scheme, including under either constraint, and that therefore AC flow and the MBO scheme can be rigorously linked together.

A primary direction for future work could be the extension of this connection to the case of the *multi-class* AC flow and MBO scheme. The multi-class versions of these flows have received considerable attention for solving problems in which one seeks to classify data into multiple classes (i.e., more than just the binary "0" and "1" classes we consider in this thesis), often with mass conservation and/or fidelity

forcing, for example in Merkurjev *et al.* [8], Jacobs *et al.* [11], and Calder *et al.* [6]. However, applying the MBO thresholding in the presence of these constraints is a non-trivial task, inspiring for example the intriguing "auction dynamics" technique of [11]. In the above, the SDIE scheme yielded (for $\lambda < 1$) a convex relaxation of the MBO thresholding whose solutions converged to MBO solutions as $\lambda \uparrow 1$. If a similar result turns out to hold in the multi-class case, that could provide another method for applying this thresholding.

Another direction for future research could be to examine higher-order MBO schemes on graphs, such as those studied in the continuum in [15, 17]. Can these be understood as higher-order numerical schemes of AC flow in a like manner to the above? Moreover, can these lead to improved classification algorithms compared those based on the basic MBO scheme? (For details on classification algorithms using the basic MBO scheme, see the next chapter).

A final direction could be linking this work back up with the continuum. It seems plausible that the SDIE link between graph AC flow and the graph MBO scheme should also hold between continuum AC flow and the continuum MBO scheme. Furthermore, we can ask what happens to this SDIE link as the underlying graph converges to a continuum object, for example in the sense considered in García Trillos *et al.* [7].

Bibliography

[1] S. Banach and S. Saks. "Sur la convergence forte dans les champs L^p". In: *Studia Mathematica* 2.1 (1930), pp. 51–57. URL: http://eudml.org/doc/217263.

[2] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. 1st ed. Cambridge, UK: Cambridge University Press, 2004. ISBN: 9780521833783.

[3] Jeremy Budd and Yves van Gennip. "Graph Merriman–Bence–Osher as a SemiDiscrete Implicit Euler Scheme for Graph Allen–Cahn Flow". In: *SIAM Journal on Mathematical Analysis* 52 (Jan. 2020), pp. 4101–4139. DOI: 10.1137/19M1277394.

[4] Jeremy Budd and Yves van Gennip. "Mass-conserving diffusion-based dynamics on graphs". In: *European Journal of Applied Mathematics* (Apr. 2021), pp. 1–49. DOI: 10.1017/S0956792521000061.

[5] Jeremy Budd, Yves van Gennip, and Jonas Latz. "Classification and image processing with a semi-discrete scheme for fidelity forced Allen–Cahn on graphs". English. In: *GAMM Mitteilungen* 44.1 (2021), pp. 1–43. ISSN: 0936-7195. DOI: 10.1002/gamm.202100004.

[6] J. Calder et al. "Poisson Learning: Graph Based Semi-Supervised Learning At Very Low Label Rates". English. In: *Proceedings of the International Conference on Machine Learning*. 2020, pp. 8588–8598.

[7] Nicolás García Trillos et al. "Error Estimates for Spectral Convergence of the Graph Laplacian on Random Geometric Graphs Toward the Laplace–Beltrami Operator". In: *Foundations of Computational Mathematics* 20 (Jan. 2020), pp. 827–887. DOI: 10.1007/s10208-019-09436-w.

[8] Cristina Garcia-Cardona et al. "Multiclass Data Segmentation Using Diffuse Interface Methods on Graphs". In: vol. 36. 8. 2014, pp. 1600–1613. DOI: 10.1109/TPAMI.2014.2300478.

[9] Y. van Gennip et al. "Mean Curvature, Threshold Dynamics, and Phase Field Theory on Finite Graphs". In: *Milan Journal of Mathematics* 82 (2014), pp. 3–65.

[10] Yves van Gennip. "An MBO Scheme for Minimizing the Graph Ohta–Kawasaki Functional". In: *Journal of Nonlinear Science* 30.2 (Oct. 2020), pp. 2325–2373. DOI: 10.1007/s00332-018-9468-8.

[11] Matt Jacobs, Ekaterina Merkurjev, and Selim Esedoğlu. "Auction dynamics: A volume constrained MBO scheme". In: *Journal of Computational Physics* 354 (2018), pp. 288–310. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2017.10.036. URL: https://www.sciencedirect.com/science/article/pii/S0021999117308033.

[12] A. Juditsky. *Convex Optimization Lecture 1*. Accessed 20/08/2021. URL: http://www-ljk.imag.fr/membres/Anatoli.Iouditski/cours/convex/chapitre_1.pdf.

[13] Xiyang Luo and Andrea Bertozzi. "Convergence of the Graph Allen–Cahn Scheme". In: *Journal of Statistical Physics* 167 (May 2017), pp. 934–958. DOI: 10.1007/s10955-017-1772-4.

[14] Walter Rudin. *Functional analysis*. 2nd ed. International series in pure and applied mathematics. New York: McGraw-Hill, 1990, p. 424.

[15] Steven J. Ruuth. "Efficient Algorithms for Diffusion-Generated Motion by Mean Curvature". In: *Journal of Computational Physics* 144.2 (1998), pp. 603–625. ISSN: 0021-9991. DOI: 10.1006/jcph.1998.6025. URL: https://www.sciencedirect.com/science/article/pii/S0021999198960259.

[16] Steven J. Ruuth and B. Wetton. "A Simple Scheme for Volume-Preserving Motion by Mean Curvature". In: *Journal of Scientific Computing* 19 (2003), pp. 373–384.

[17] Alexander Zaitzeff, Selim Esedoğlu, and Krishna Garikipati. "Second order threshold dynamics schemes for two phase motion by mean curvature". In: *Journal of Computational Physics* 410 (2020), p. 109404. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2020.109404.

5

Applications in Image Segmentation

One day ladies will take their computers for walks in the park and tell each other, "My little computer said such a funny thing this morning!"

Alan Turing, quoted in *Alan Turing: The Enigma*

A major application of the graph flows defined in chapter 3 has been in image segmentation, the task of locating the "important" parts of an image. In this chapter, we shall describe how our SDIE scheme can be used as an algorithm for this task, along the way presenting some refinements compared to how this was previously done in a paper by Merkurjev, Kostić, and Bertozzi [22] with the MBO scheme. We then test this method on the "two cows" example considered by [22] (and also Bertozzi and Flenner [6]), as well as two related examples. We find that whilst in this example the SDIE scheme does not improve on its MBO special case, our other refinements lead to a substantially improved segmentation compared to those of [6] and [22].

Parts of this chapter have been published in *GAMM Mitteilungen* 44 (2021) [9]. Jonas Latz contributed significantly to this chapter.

5.1. Introduction

In this chapter, we will be investigating how the tools from the last chapter can be applied in *image segmentation*, i.e. the task of locating the "important" parts of an image. For example, in an image of a cancer patient the "important" part to be located might be the pixels corresponding to a tumour or tumours. This is an example of a classification problem, i.e. a problem where one wishes to assign a label to every member of a set of individuals (in this case, the set of pixels in an image) in such a way that individuals which are relevantly "similar" get assigned the same label. In classification problems, one also has *a priori* information, in the form of some subset of already labelled individuals. The task then is to propagate those *a priori* labels to the whole set.

In this chapter, we will restrict ourselves to *binary* classification problems, i.e. problems in which there are only two labels, and the key idea will be to encode the individuals as a graph and use the fidelity forced SDIE scheme to propagate the labels.

5

5.1.1. Background

In the continuum, a major class of techniques for classification problems relies upon the minimisation of total variation (TV), e.g. the famous Mumford–Shah [24] and Chan–Vese [11] algorithms. These methods are linked to Ginzburg–Landau methods by the fact that the (continuum) Ginzburg–Landau functional Γ-converges to TV [20, 23] (a result that continues to hold in the graph context [16]). This motivated a common technique of minimising the Ginzburg–Landau functional in place of TV, e.g. in Esedoğlu and Tsai [14] two-class Chan–Vese segmentation was implemented by replacing TV with the Ginzburg–Landau functional; the resulting energy was minimised by using a fidelity forced MBO scheme.

Inspired by this continuum work, in Bertozzi and Flenner [6] a method for graph classification was introduced based on minimising the Ginzburg–Landau functional on a graph by evolving the graph Allen–Cahn (AC) equation. 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 labelling function which, due to the influence of a double-well potential (e.g. $W(x) = x^2(x-1)^2$) 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 Merkurjev, Kostić, and Bertozzi [22] an alternative method was introduced: a graph MBO scheme with fidelity forcing, inspired by the use of the original MBO scheme [4] to approximate motion by mean curvature in the continuum. Heuristically, this MBO scheme was expected to behave similarly to the graph AC flow as the thresholding step resembles a "hard" version of the "soft" double-well potential nonlinearity in the AC flow. We have rigorously supported this heuristic argument in the previous chapter.

5.2. The SDIE scheme as a classification algorithm

In the last chapter, we showed that the fidelity forced SDIE scheme (4.1) generalises the MBO scheme into a family of schemes, all of the same computational cost as the MBO scheme (excluding the negligible extra cost incurred performing a piecewise linear thresholding instead of a hard thresholding), and which as $\tau \downarrow 0$ become increasingly more accurate approximations of the AC trajectories. As was described in the above background, graph AC and MBO trajectories can be deployed as classification algorithms, as was originally done in work by Bertozzi and co-authors in [6, 22]. In this chapter, we will investigate whether the use of the SDIE scheme can significantly improve on the use of the MBO scheme to segment the "two cows" images from [6, 22]. We will also investigate other refinements to the method of [22].

5.2.1. Groundwork

In this section, we will describe the framework for applying graph dynamics to classification problems, following [6, 22].

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 set of pixels in the image, and x is the greyscale/RGB/etc. values at each pixel. Furthermore, we have labelled 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 from Definition 3.2.3 of $M := \mathrm{diag}(\mu)$ 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 labelled) to have feature vectors that are "close together". What this means in practice will depend on the application, e.g. [28] incorporates texture into the features and [6, 10] give other options.

Next, we construct the weights on the graph by deciding on the edge set E (e.g. $E = V^2 \setminus \{(i,i) \mid i \in V\}$) and for each $ij \in E$ computing $\omega_{ij} := \Omega(z_i, z_j)$ (and for $ij \notin E$, $\omega_{ij} := 0$) for some similarity function Ω. There are a number of standard choices for the similarity function Ω, see e.g. [6, 7, 31, 33]. The similarity function we will use in this thesis is the Gaussian function:

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

where $||\cdot||_F$ denotes the Frobenius norm.

Note 28. *The above described process for constructing a graph upon (image) data does not exhaust all methods used in the literature. For example, there has been recent interest in using deep learning methods to construct graphs upon data, see e.g. de Vriendt* et al. [29].

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 *normalised* (a.k.a. random walk) graph Laplacian, i.e. we will henceforth

take $r = 1$ and so $\Delta = I - D^{-1}\omega$. We will also consider the *symmetric normalised* Laplacian $\Delta_s := I - D^{-1/2}\omega D^{-1/2}$, though this does not fit into the schema for the graph Laplacian given in chapter 2 (extending the theory from the previous two chapters to the symmetric normalised Laplacian case is a topic for future research). This normalisation 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], normalisation vastly improves these segmentation properties. As that figure looked at the symmetric normalised Laplacian, we include Fig. 5.1 to show the difference between the symmetric normalised and the random walk Laplacian.

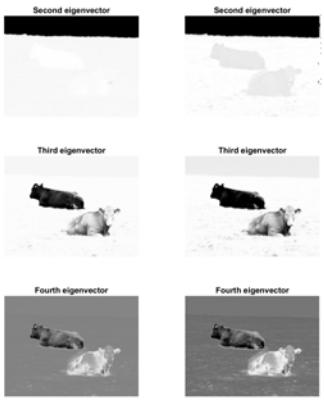


Figure 5.1: Second, third, and fourth eigenvectors of the random walk Laplacian (left) and symmetric normalised Laplacian (right) for the graph built on one of the "two cows" images from Example 5.3.1, computed using Algorithm 1.

5.2.2. The basic classification algorithm

For some time step $0 < \tau \leq \varepsilon$ note that from (3.7),

$$\mathcal{S}_\tau u = e^{-\tau A}u + b$$

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

Then the basic approach of the classification algorithm can be summarised as follows.

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 $\omega_{ij} := \Omega(z_i, z_j)$, a from this compute A and b.
4. From some initial condition u_0, compute the SDIE sequence u_n until it meets a stopping condition at some $n = N$.

5. **Output:** u_N thresholded to lie in $\mathcal{V}_{\{0,1\}}$, i.e. we output $\Theta(u_N)$ where Θ is as in (3.1).

Unfortunately, as written this algorithm cannot be feasibly run. The chief obstacle is that in many applications ω, and therefore A, are 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.

5.2.3. Matrix compression and approximate SVDs

Since A is Δ plus a diagonal matrix, the latter of which we can therefore easily store in memory, our first challenge will be to compress Δ into something we can store in memory. Following [6, 22], we employ the Nyström extension [15, 25]. 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 the matrices $\omega_{VX} := \omega(V, X)$ (i.e. $\omega_{VX} := (\omega_{ij})_{i\in V, j\in X}$) and $\omega_{XX} := \omega(X, X)$ and then the Nyström extension is the approximation:

$$\omega \approx \omega_{VX}\omega_{XX}^{-1}\omega_{VX}^T. \tag{5.1}$$

Note 29. *This approximation avoids having to compute the full matrix ω, which in many applications is too large to store in memory. However, a word of warning: there is no* a priori *guarantee that ω_{XX} will be well-conditioned or even invertible. For example, if there were no edges between elements of X then ω_{XX} would be the zero matrix. Therefore, the use of this approximation induces some constraints on one's construction of the graph edge weights and one's choice of K. Fortunately, since ω_{XX} is by construction small, one can compute its condition number and thereby check whether or not invertibility is in fact an issue for a given generated ω_{XX}.*

We next compute an approximation for the degree vector d and degree matrix $D := \mathrm{diag}(d)$ of our graph

$$d \approx \hat{d} := \omega_{VX}\omega_{XX}^{-1}\omega_{VX}^T\mathbf{1}, \qquad D \approx \hat{D} := \mathrm{diag}\left(\hat{d}\right).$$

We thus approximately normalise ω

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

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

Following [6, 22], we next compute an approximate eigendecomposition of $\tilde{\omega}$. We here diverge from the method of [6, 22]. 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.[1] Whilst this clearly does not prevent the method

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

of [6, 22] 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 Bebendorf and Kunis [3] for computing a singular value decomposition (SVD) from an adaptive cross approximation (ACA) (see [3] for a definition of ACA), which was recently recommended for the Nyström decomposition of graph Laplacians in Alfke *et al.* [1].

First, we compute the so-called "thin" QR decomposition (see [17, Theorem 5.2.2])

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

where $Q \in \mathbb{R}^{|V|\times K}$ has orthonormal columns, 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 the orthogonal matrix of eigenvectors of $R\omega_{XX}^{-1}R^T$ and $\Sigma \in \mathbb{R}^{K\times K}$ is the diagonal matrix of the corresponding eigenvalues. 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 $U_s := Q\Phi$ has orthonormal columns. This gives an approximate eigendecomposition of the symmetric normalised 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 $\Lambda := I_K - \Sigma$, 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 $U_1 := \hat{D}^{-1/2}U_s$ and $U_2 := \hat{D}^{1/2}U_s$. As in [3], it is easy to see that the computational complexity of this approach is $\mathcal{O}(K|V|)$ in space and $\mathcal{O}(K^2|V| + K^3)$ in time. We summarise this all as Algorithm 1.

5.2.4. Numerical examination of the matrix compression methods

We consider the accuracy of our Nyström-QR approach for the compression of the symmetric normalised Laplacian[2] Δ_s built on the image in Fig. 5.2, containing $|V| = 6400$ pixels, which is small enough for us to compute the true value of Δ_s to high accuracy. For $K \in \{50, 100, \ldots, 500\}$, we compare the rank K approximation $U_s\Lambda U_s^T$ with the true Δ_s in terms of the relative Frobenius distance, i.e. $||U_s\Lambda U_s^T - \Delta_s||_F / ||\Delta_s||_F$. Moreover, we compare these errors to the errors incurred by other low-rank approximations of Δ_s, namely the Nyström method used in [6,

[2]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 or Δ_s, inspired by Bebendorf and Kunis [3].

function NyströmQR($ij \mapsto \omega_{ij}, V, Z, K$) // Computes U_1, Λ, and U_2, where $\Delta \approx U_1 \Lambda U_2^T$ 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, i.e. $(\tilde{\omega}_{VX})_{ij} = \hat{d}_i^{-1/2}(\omega_{VX})_{ij}$
  $[Q, R] = \texttt{thin\_qr}(\tilde{\omega}_{VX})$ // 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

22], the Halko–Martinsson–Tropp (HMT) method[3] [18], and the optimal rank K approximation of Δ_s with respect to $||\cdot||_F$, which (by the Eckart–Young theorem [13]; see also [17, 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 also measure the complexity of the methods, via the runtimes of MATLAB R2019a implementations of them on the set-up as in section 5.3.2.

We report the relative Frobenius distance in Fig. 5.3. As the Nyström (and HMT) methods are randomised, we repeat the experiments 1000 times and plot the mean (relative Frobenius) error in the far-left figure and the standard deviations of the errors in the centre-right figure. To expose the difference between the methods for $K \geq 100$, we plot the percentage increase of the other mean errors above the SVD error (i.e. $[\text{mean error}/\text{error}_{\text{SVD}} - 1] \times 100\%$) and show this percentage in the centre-left figure. In the far-right figure, we compare the average runtime of the algorithms. 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

[3] The HMT results serve only to give an additional benchmark for the Nyström methods: HMT requires matrix-vector-products with the full Δ_s, which was infeasible for us in applications due to the size of the matrix. A topic for future work will be to investigate the use of [1, Algorithm 3], which potentially allows feasible computation of an HMT-like SVD of Δ_s.

Figure 5.2: The 80×80 image on which the Laplacian Δ_s is constructed to test the low-rank approximations.

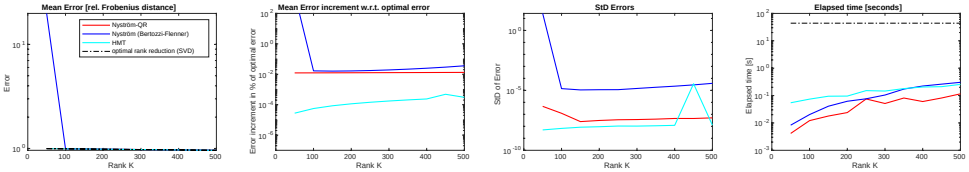


Figure 5.3: Error of the low-rank approximations of Δ_s in terms of their relative Frobenius distance to the true Δ_s (far-left), percentage increase above the optimal error reached with the SVD (centre-left), and standard deviations (StD) of the errors (centre-right). Timings of the methods in seconds (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.

from [6, 22]: it is faster, more accurate, and the mean error does not blow up 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 in K, suggesting 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 error ranges from 8.87×10^{-7} $(K = 50)$ to 5.00×10^{-8} $(K = 500)$. By contrast, for the Nyström method from [6, 22] we see much more random variation. The bump in the HMT error at $K = 450$ is an outlier that is present in many repeats of this experiment.

5.2.5. Interlude: an analysis of the method from [22]

In [22], the authors approximated $\mathcal{S}_\tau u$ by a semi-implicit Euler method for fidelity forced diffusion. That is, since $\mathcal{S}_\tau u$ 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 [22] approximate trajectories of this ODE by a semi-implicit Euler method with time step $\delta t > 0$ (such that $\tau/\delta t \in \mathbb{N}$), i.e. $v_0 := u$ and

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

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

$$\begin{aligned} v_{k+1} &= (I + \delta t\Delta)^{-1}\left(v_k - \delta tM(v_k - \tilde{f})\right) \\ &= (I + \delta t\Delta)^{-1}(I - \delta tM)\, v_k + \delta t\,(I + \delta t\Delta)^{-1} f. \end{aligned} \tag{5.2}$$

This Euler method then corresponds to the approximation $\mathcal{S}_\tau u \approx v_{\tau/\delta t}$. To compute (5.2), they use the Nyström decomposition to compute the leading eigenvectors and eigenvalues of Δ.

Note 30. *In fact, in* [22] *the authors use* Δ_s *not* Δ*. It makes no difference to this analysis which Laplacian is used.*

Given $\Delta \approx U_1 \Lambda U_2^T$, i.e. the SVD of a low-rank matrix which approximates Δ, the authors approximate (5.2) by

$$\begin{aligned} \hat{v}_{k+1} &= U_1(I_K + \delta t\Lambda)^{-1}U_2^T(\hat{v}_k - \delta tM(\hat{v}_k - \tilde{f})) \\ &= U_1(I_K + \delta t\Lambda)^{-1}U_2^T(I - \delta tM)\hat{v}_k + \delta tU_1(I_K + \delta t\Lambda)^{-1}U_2^T f. \end{aligned} \tag{5.3}$$

The final approximation for $\mathcal{S}_\tau u$ in [22] is then computed by setting $\hat{v}_0 = u$ and $\mathcal{S}_\tau u \approx \hat{v}_{\tau/\delta t}$.

Note 31. *The substitution of* $U_1(I_K + \delta t\Lambda)^{-1}U_2^T$ *for* $(I + \delta t\Delta)^{-1}$ *incurs an* $\mathcal{O}(1)$ *error. In particular, if* $\delta t < \min\left\{||\Lambda||^{-1}, ||\Delta||^{-1}\right\}$ *then*

$$\begin{aligned} U_1(I_K + \delta t\Lambda)^{-1}U_2^T &= U_1\left(I_K - \delta t\Lambda + \delta t^2\Lambda^2 - + \cdots\right)U_2^T \\ &= U_1U_2^T - \delta tU_1\Lambda U_2^T + \delta t^2U_1\Lambda^2U_2^T - + \cdots \\ &\approx U_1U_2^T - I + \left(I - \delta t\Delta + \delta t^2\Delta^2 - + \cdots\right) \\ &= U_1U_2^T - I + (I + \delta t\Delta)^{-1} \end{aligned}$$

where $U_1U_2^T$ *is the projection onto the singular vectors used in the low-rank approximation, and so the error incurred arises from replacing the identity with this projection.*

Both (5.2) and (5.3) are of the form

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

By induction, such a scheme has k^{th} term

$$v_k = \mathcal{A}^k v_0 + \sum_{r=0}^{k-1} \mathcal{A}^r g = \mathcal{A}^k v_0 + (\mathcal{A} - I)^{-1}(\mathcal{A}^k - I)g. \tag{5.4}$$

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

$$\begin{aligned}
&\mathcal{S}_\tau u \\
&\approx \left[(I+\delta t\Delta)^{-1}(I-\delta tM)\right]^k u \qquad (5.5)\\
&+\left[(I+\delta t\Delta)^{-1}(I-\delta tM)-I\right]^{-1}\left(\left[(I+\delta t\Delta)^{-1}(I-\delta tM)\right]^k - I\right)\delta t\,(I+\delta t\Delta)^{-1} f \\
&\approx \left[U_1(I_K+\delta t\Lambda)^{-1}U_2^T(I-\delta tM)\right]^k u \qquad (5.6)\\
&+\left[U_1(I_K+\delta t\Lambda)^{-1}U_2^T(I-\delta tM)-I\right]^{-1} \\
&\quad\left(\left[U_1(I_K+\delta t\Lambda)^{-1}U_2^T(I-\delta tM)\right]^k - I\right)\delta t U_1(I_K+\delta t\Lambda)^{-1}U_2^T f.
\end{aligned}$$

To see what these approximations are doing, note that with respect to the limit $\delta t \to 0$,

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

and note the Lie product formula [19, Theorem 2.11] (with respect to the limit $k \to \infty$)

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

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

$$\begin{aligned}
\left[(I+\delta t\Delta)^{-1}(I-\delta tM)\right]^k &= \left[e^{-\delta t\Delta}e^{-\delta tM} + \mathcal{O}(\delta t^2)\right]^k \\
&= \left[e^{-\delta t\Delta}e^{-\delta tM}\right]^k + \mathcal{O}(k\delta t^2) \\
&= \left(e^{-\tau(\Delta+M)} + \mathcal{O}(k\delta t^2)\right) + \mathcal{O}(k\delta t^2) \\
&= e^{-\tau A} + \mathcal{O}(\delta t)
\end{aligned} \qquad (5.7)$$

and the second term in (5.5) becomes

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

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

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

is the commutation error. Hence the overall error for (5.5) is $\mathcal{O}(\delta t)$. The error for (5.6) 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 (3.7) 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 (5.7). By (5.4) and (5.7), we can write the Euler approximation for the integral term as

$$\int_0^\tau e^{-tA} f \, dt \approx \delta t \sum_{r=0}^{k-1} \left[(I + \delta t \Delta)^{-1} (I - \delta t M) \right]^r (I + \delta t \Delta)^{-1} f \tag{5.8}$$

$$\approx \delta t \sum_{r=0}^{k-1} \left[U_1 (I_K + \delta t \Lambda)^{-1} U_2^T (I - \delta t M) \right]^r U_1 (I_K + \delta t \Lambda)^{-1} U_2^T f. \tag{5.9}$$

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

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

applying the reciprocation elementwise. Therefore, we can rewrite (5.8) as

$$\begin{aligned}
b &:= \int_0^\tau e^{-tA} f \, dt \\
&\approx \delta t \sum_{r=0}^{k-1} \left[(I + \delta t \Delta)^{-1} (I - \delta t M) \right]^{r+1} \left((\mathbf{1} - \delta t \mu)^{-1} \odot f \right) \\
&= \delta t \sum_{r=1}^{k} \left(e^{-r\delta t A} \left((\mathbf{1} - \delta t \mu)^{-1} \odot f \right) + \mathcal{O}(r \delta t^2) \right) \qquad \text{by (5.7) and relabelling } r \\
&= \left(\delta t \sum_{r=1}^{k} e^{-r\delta t A} \left((\mathbf{1} - \delta t \mu)^{-1} \odot f \right) \right) + \mathcal{O}(\tau^2 \delta t)
\end{aligned}$$

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

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

Likewise, we can rewrite (5.9) 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 (5.8) to (5.9) has incurred an extra error from the spectrum truncation alongside the quadrature and Lie product formula errors.

The key takeaway from these calculations concerns (5.7). That equation shows that the Euler approximation for the $e^{-\tau A}$ term is in fact an approximation of an approximation. That is, it approximates a Lie product formula approximation for $e^{-\tau A}$. This motivates the method we shall explore in the next subsection of cutting out the middleman and using a matrix exponential formula directly, and furthermore using a formula which is more accurate than the Lie product formula. We have also shown how the [22] Euler method approximation for b can be written as a form of quadrature, motivating our investigation in the next subsection into other quadrature methods as potential improvements for computing b.

5.2.6. Computing the SDIE scheme: a Strang formula method

To compute the iterates of our SDIE scheme, we will need to compute an approximation for $\mathcal{S}_\tau u_n$. We saw in (5.7) in the previous subsection that the [22] semi-implicit Euler method works by approximating a Lie product formula approximation of $e^{-\tau A}$, which incurs an error which is linear in the Euler time-step (plus a spectrum truncation error). Therefore we propose as an improvement a scheme that directly employs the superior Strang formula[4] 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;[5] by expressing b as a solution to the ODE from (3.6) with initial condition $\mathbf{0}$, and using the Euler method from [22] with a very small time step (or a higher-order ODE solver);[6] or by computing the closed form solution for b directly using the Woodbury identity [30]. We therefore improve on the accuracy of computing $\mathcal{S}_\tau u$ at low cost.

The Strang formula for matrix exponentials [27] is given, for $k > 0$ a parameter and $\mathcal{O}$ relative to the limit $k \to \infty$, by

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

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 $\delta t := \tau/k$)

$$\begin{aligned} e^{-\tau A}u &= \left(e^{-\frac{1}{2}\tau/kM} e^{-\tau/k\Delta} e^{-\frac{1}{2}\tau/kM}\right)^k u + \mathcal{O}(k^{-2}) \\ &= \left(e^{-\frac{1}{2}\delta t M}\left(I + U_1(e^{-\delta t\Lambda} - I_K)U_2^T\right) e^{-\frac{1}{2}\delta t M}\right)^k u + E + \mathcal{O}(\delta t^2) \\ &=: v_k + E + \mathcal{O}(\delta t^2) \end{aligned} \tag{5.10}$$

where E is a spectrum truncation error.[7]

[4] We owe the suggestion to use this formula to Arieh Iserles, who also suggested to us the Yoshida method that we consider below.

[5] We again thank Arieh Iserles for also making this suggestion.

[6] 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.

[7] Specifically, $E = (e^{-\frac{1}{2}\delta t M} e^{-\delta t\Delta} e^{-\frac{1}{2}\delta t M})^k u - (e^{-\frac{1}{2}\delta t M}(I + U_1(e^{-\delta t\Lambda} - I_K)U_2^T)e^{-\frac{1}{2}\delta t M})^k u$ is incurred by the spectrum truncation in the middle of the latter term.

Note 32. *Here we have used that:*

$$\begin{aligned} e^{-U_1 \Lambda U_2^T} &= I - U_1 \Lambda U_2^T + \frac{1}{2} U_1 \Lambda^2 U_2^T - + \cdots \\ &= I + U_1 \left(-\Lambda + \frac{1}{2} \Lambda^2 - + \cdots \right) U_2^T \\ &= I + U_1 (e^{-\Lambda} - I_K) U_2^T. \end{aligned}$$

This works because (recalling the notation of section 5.2.3) $U_2^T U_1 = U_S^T \hat{D}^{1/2} \hat{D}^{-1/2} U_S = U_S^T U_S$ *and* $U_S^T U_S = \Phi^T Q^T Q \Phi = I_K$ *because* Q *has orthonormal columns and* Φ *is orthogonal. Note that therefore we have here avoided needing to use the* [22] *approximation of* I *by* $U_1 U_2^T$ *(see Note 31) and hence avoided the error that approximation incurs.*

Furthermore, we can compute the term v_k by setting $v_0 := u$, and defining v_r for $r \in \{1, ..., k\}$ iteratively by

$$\begin{aligned} v_{r+1} &= e^{-\delta t M} v_r + e^{-\frac{1}{2}\delta t M} U_1 (e^{-\delta t \Lambda} - I_K) U_2^T e^{-\frac{1}{2}\delta t M} v_r \\ &= a_1(\delta t) \odot v_r + a_3(\delta t) \odot \left(U_1 \left(a_2(\delta t) \odot \left(U_2^T \left(a_3(\delta t) \odot v_r \right) \right) \right) \right) \\ &=: S(\delta t) v_r \end{aligned} \tag{5.11}$$

where $\odot$ is the Hadamard (i.e. elementwise) product, $a_1(\delta t) := \exp(-\delta t \mu)$, $a_2(\delta t) := \exp(-\delta t \operatorname{diag}(\Lambda)) - \mathbf{1}_K$, and $a_3(\delta t) := \exp(-\frac{1}{2}\delta t \mu)$ 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 Fig. 5.5, we verify on a simple image that this method has quadratic error (plus a spectrum truncation error) and outperforms the [22] Euler method. Moreover, (5.11) is (to leading order) as fast as (5.3) (i.e., a step of the [22] Euler method). This is because by defining $\tilde{a}_1 := \mathbf{1} - \delta t \mu$ and $\tilde{a}_2 := (\mathbf{1}_K + \delta t \operatorname{diag}(\Lambda))^{-1}$ (applying the reciprocation elementwise), we can rewrite (5.3) 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 (5.11) and (5.3) involve two $\mathcal{O}(NK)$ matrix multiplications and the vectors in (5.11) and (5.3) and are at most N-dimensional, hence the Hadamard products in (5.11) and (5.3) are all at most $\mathcal{O}(N)$ and so are not the leading order contributions to the computation time.

At the cost of extra $\mathcal{O}(NK)$ matrix multiplications, one can employ the method of Yoshida [32] to increase the order of the (non-spectrum-truncation) error by 2. If we set $\alpha_0 := -\sqrt[3]{2}/(2 - \sqrt[3]{2})$ and $\alpha_1 := 1/(2 - \sqrt[3]{2})$ then from the map $S(t)$ from (5.11) we can define the map

$$Y(\delta t) := S(\alpha_1 \delta t) \circ S(\alpha_0 \delta t) \circ S(\alpha_1 \delta t)$$

which gives $Y^k(\delta t) u = e^{-\tau A} u + \mathcal{O}(\delta t^4)$ plus a spectrum truncation error.[8] However, as can be seen in Fig. 5.5(c,d), the spectrum truncation error can make negligible any gain from using the Yoshida method over the Strang formula.

[8] This method can be extended to give higher-order formulae of any even order [32], but consideration of those formulae is beyond the scope of this thesis.

It remains to compute an approximation for $b := A^{-1}(I - e^{-\tau A})f$. A straightforward application of calculus shows that b can be rewritten as the integral

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

which we can approximate via a quadrature, e.g. applying the midpoint or Simpson's rules respectively we get

$$b = \tau e^{-\frac{1}{2}\tau A} f + \mathcal{O}(\tau^3) = \frac{1}{6}\tau(I + 4e^{-\frac{1}{2}\tau A} + e^{-\tau A})f + \mathcal{O}(\tau^5) \tag{5.12}$$

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 $h := \tau/(2m)$ 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 $k_b = 2m$ (i.e. if the Simpson subdivision equals the Strang/Yoshida step number) can be approximated using the above tools by

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

where one can choose $w_r := S^r(h)f$ with quadrature error $E_1 = \mathcal{O}(\tau^2 h)$ or choose $w_r := Y^r(h)f$ with quadrature error $E_1 = \mathcal{O}(\tau^2 h^3)$, and where E_2 is the spectrum truncation error. Finally, we can also let MATLAB compute whatever quadrature it thinks is best 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 (3.7), b is the fidelity forced diffusion of $\mathbf{0}$ at time τ, i.e.

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

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

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

via the semi-implicit Euler method from [22]. Since we only need to compute b once we can choose a small time step, i.e. 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 [22] notes this ODE is stiff, which we found causes standard solvers such as `ode45` (i.e. Dormand–Prince-(4, 5) [12]) 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 $b = A^{-1} g$. Towards this, we consider the Woodbury identity [30]

$$(\Psi + \tilde{U}_1 \Xi \tilde{U}_2^T)^{-1} = \Psi^{-1} - \Psi^{-1}\tilde{U}_1 \left(\Xi^{-1} + \tilde{U}_2^T \Psi^{-1} \tilde{U}_1\right)^{-1} \tilde{U}_2^T \Psi^{-1}$$

where $\Psi \in \mathbb{R}^{N\times N}$ and $\Xi \in \mathbb{R}^{K\times K}$ are invertible, and $\tilde{U}_1, \tilde{U}_2 \in \mathbb{R}^{N\times K}$. Then given this identity and our approximation $\Delta \approx U_1 \Lambda U_2^T$, we approximate

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

where $\Psi := I + M$, superscript $+$ denotes the Moore–Penrose pseudoinverse (see [17, §5.5.2]), and $\Sigma := I_K - \Lambda$ (note that this approach does involve approximating I by $U_1 U_2^T$). Then

$$b = \psi \odot (g - U_1 x)$$

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

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

where we define $\psi \odot U_1$ as columnwise Hadamard multiplication, i.e. $(\psi \odot U_1)_{ij} := \psi_i (U_1)_{ij}$.

We compare the accuracy of these approximations of b in Table 5.1, and observe that no method is hands-down superior. Table 5.1 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.

5.2.7. Numerical examination of the methods for computing the SDIE scheme

In this section, we will build our graphs on a 80×80 image and a 40×40 image of the form displayed in Fig. 5.4, which are 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 Fig. 5.5 we investigate the accuracy of the Strang formula and Yoshida method vs. the [22] 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 $K = |V|$ (i.e. 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,[9] and that the Yoshida

[9] In the rank-reduced case, the $\mathcal{O}(1)$ improvement of the Strang/Yoshida methods over the Euler method derives from the latter making an approximation of I that incurs an $\mathcal{O}(1)$ error, whilst the former both avoid this. See Notes 31 and 32 for details.

Algorithm 2 The SDIE scheme via a Strang formula method.

function SDIE(μ, f, U_1, Λ, U_2, τ, ε, u_0, k, k_b, K, δ) // 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
    $\psi = (1 + \mu)^{-1}$
    $\Sigma = I_K - \Lambda$
    $(-\Sigma^+ + U_2^T(\psi \odot U_1))x = U_2^T(\psi \odot g)$ // Solving the linear system for x
    $b = \psi \odot (g - U_1 x)$
  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**
      $v = a_1 \odot v + a_3 \odot \left(U_1\left(a_2 \odot \left(U_2^T\left(a_3 \odot v\right)\right)\right)\right)$ // 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 (4.6), the piecewise linear thresholding
    $n = n + 1$
  end while
  return u_n
end function

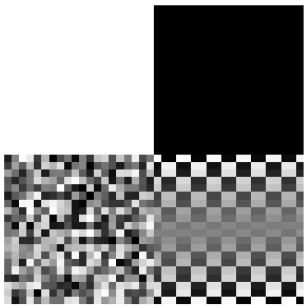
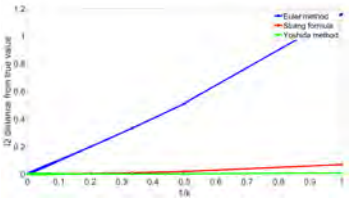


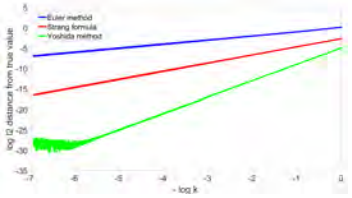
Figure 5.4: The 80×80 image (the 40×40 image is similar) that we build one of our graphs upon, using feature vectors as described in section 5.3.2.

method is more accurate than the Strang formula, but only barely in the rank-reduced case (with the difference in the rank-reduced case plotted in Fig. 5.5(d)). Furthermore, the log-log gradients of the Strang formula error and the Yoshida method error (excluding the outliers for small k values and the outliers caused by errors from reaching machine precision) in Fig. 5.5(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.

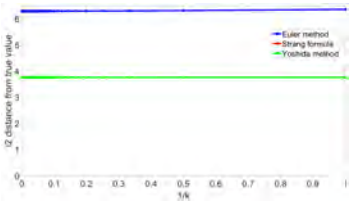
Next, in Table 5.1 we compare the accuracy of the different methods for computing b. We take Z as the left two quadrants of the image, $\mu = \chi_Z$, $\tilde{f}$ as equal to the image on Z, and $k_b = 1000$ in the Strang formula/Yoshida method approximations for $e^{-tA}f$ and in the [22] Euler scheme. 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 5.3.1), we have observed that (interestingly) the [22] Euler method yields the best segmentation. A topic for future research can be whether this is generally true for large matrices.



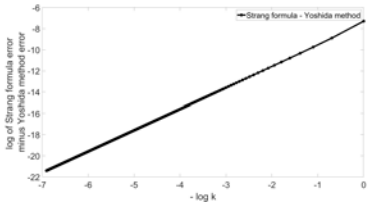
(a) Plot of ℓ^2 errors vs. $1/k$.



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



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



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

Figure 5.5: 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 Fig. 5.4. For (a) and (b), $K = |V|$. For (c) and (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.

5.3. Applications in image processing

5.3.1. Examples

We consider three examples, all using images of cows from the Microsoft Research Cambridge Object Recognition Image Database.[10] Some of these images have been used before by [6, 22] to illustrate and test graph-based segmentation algorithms.

Example 5.3.1 (Two cows)**.** *We first introduce the "two cows" example familiar from* [6, 22]*. We take the image of two cows in the top left of Fig. 5.6 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 Fig. 5.6, 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, i.e. the reference data and the image are tensors of size $480 \times 640 \times 3$.*

We will use Example 5.3.1 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 the non-MBO SDIE case ($\tau < \varepsilon$) and MBO SDIE case ($\tau = \varepsilon$);

[10] Available at `https://www.microsoft.com/en-us/research/project/image-understanding/`, accessed 17 June 2021.

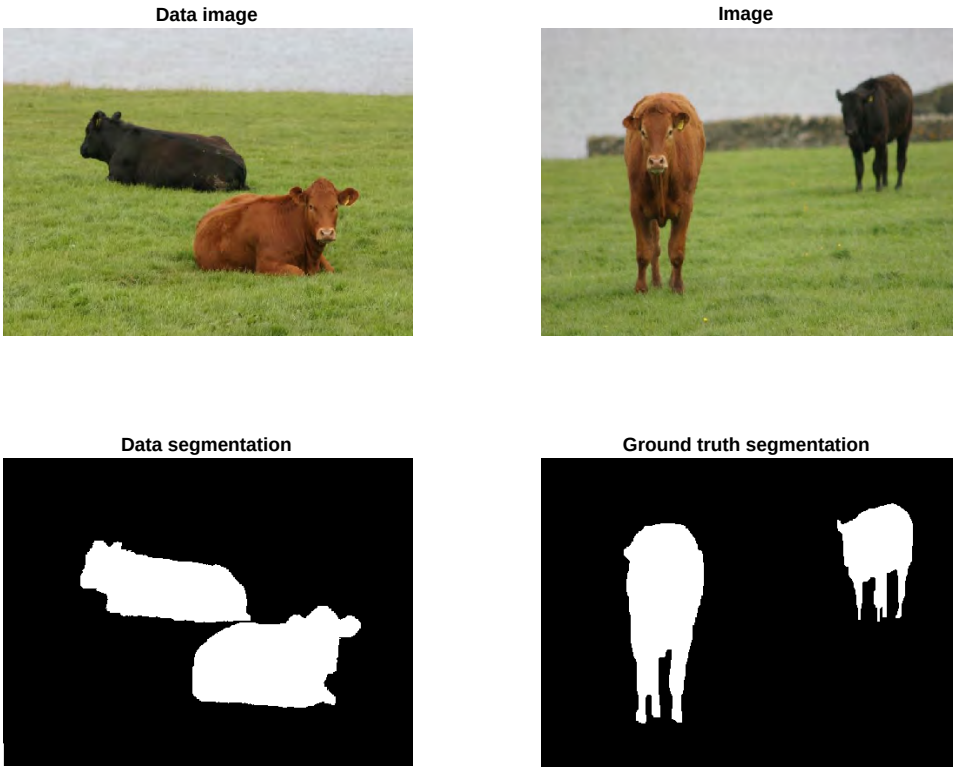


Figure 5.6: Two cows: the reference data image, the image to be segmented, the reference $\tilde{f}$ (which is a segmentation of the reference data image), and the ground truth segmentation associated to Example 5.3.1. Both segmentations were drawn by hand by the authors.

Method	**Relative ℓ^2 error for $\tau = 0.5$**			**Relative ℓ^2 error for $\tau = 4$**		
$\lvert V\rvert, K$	1600,1600	1600,40	6400,40	1600,1600	1600,40	6400,40
Euler method [22]	$1.43 \cdot 10^{-4}$	0.495	0.411	$2.88 \cdot 10^{-4}$	0.207	0.172
Woodbury identity	$2.29 \cdot 10^{-8}$	0.575	0.461	$1.04 \cdot 10^{-7}$	**0.197**	**0.154**
Midpoint rule (5.12)	0.028	**0.129**	**0.111**	0.428	0.608	0.611
Simpson's rule via Strang formula	$2.59 \cdot 10^{-9}$	0.134	0.114	$5.85 \cdot 10^{-7}$	0.512	0.483
MATLAB `integrate` via Strang formula	n/a	0.134	0.114	n/a	0.512	0.483
Simpson's rule via Yoshida method	$\mathbf{8.38 \cdot 10^{-14}}$	0.134	0.114	$\mathbf{9.34 \cdot 10^{-12}}$	0.512	0.483
MATLAB `integrate` via Yoshida method	n/a	0.134	0.114	n/a	0.512	0.483

Table 5.1: Comparison of the relative ℓ^2 errors from the methods for approximating b on the image from Fig. 5.4. We did not compute `integrate` for $K = 1600$ as it ran too slowly. Bold entries indicate the smallest error in that column. Simpson's rule was computed with $m = 500$.

5

- compare different normalisations of the graph Laplacian, i.e. the symmetric vs. random walk normalisation;
- investigate the influence of the Nyström-QR approximation of the graph Laplacian (i.e., Algorithm 1) in terms of the rank K; and
- quantify the inherent uncertainty in the computational strategy induced by the randomised Nyström approximation.

Example 5.3.2 (Greyscale)**.** *This example is the greyscale version of Example 5.3.1. Hence, we map the images in Fig. 5.6 to greyscale using* `rgb2gray`*. We show the greyscale images in Fig. 5.7. We use the same segmentation of the reference data image as in Example 5.3.1. The greyscale images are matrices of size* 480×640*.*

Figure 5.7: Two cows greyscale: the reference data image and the image to be segmented associated to Example 5.3.2. Note that the reference $\tilde{f}$ and the ground truth segmentation are identical to those in Fig. 5.6.

The greyscale image is much harder to segment than the RGB image, as there is no clear colour separation. With Example 5.3.2, we aim to illustrate the performance of the SDIE scheme in a harder segmentation task.

Example 5.3.3 (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 Fig. 5.8. Again, we shall separate the cows from the background. As reference data, we use the reference data image and labels as in Example 5.3.1. 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* $480 \times 2560 \times 3$*.*

Figure 5.8: Many cows: the image to be segmented associated to Example 5.3.3. Note that the reference data image and labels are identical to those in Fig. 5.6 (left).

With Example 5.3.3, 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 33. *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 thesis but is explored for the* [22] *MBO segmentation algorithm and related methods in* [26, Figure 4]*.*

5.3.2. Set-up

Algorithms We here use the Nyström-QR method (Algorithm 1) to compute the rank K approximation to the Laplacian, and we use the [22] semi-implicit Euler method (with time step τ/k_b) to compute b (as we found that in practice, for the above examples, this worked better than the quadrature and Woodbury identity methods).

Feature vectors Let $\mathcal{N}(i)$ denote the 3×3 neighbourhood 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 $J \in \{R, G, B\}$ (i.e. 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 $z_i := (z_i^R, z_i^G, z_i^B) \in \mathbb{R}^{3\times3\times3}$, which we reshaped (using `reshape`) so that $z \in \mathbb{R}^{|V|\times27}$.

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, i.e. 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 (e.g. the grass) getting labelled 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 34. *It is not an improvement to initialise 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. 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* [22] *nor for* [6].

Fidelity parameter We followed [22] 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.

5.3.3. The "two cows" example

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

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 Fig. 5.9. Moreover, we show the progression of the algorithms in Fig. 5.11. The

(a) Reference data image, image to be segmented, and image masked with ground truth segmentation.

(b) $\tau = 0.001 \ll 0.003 = \varepsilon$. Relative segmentation error: 1.6416%, elapsed time: 13.0 sec.

(c) $\tau = 0.0025 < 0.003 = \varepsilon$. Relative segmentation error: 1.9424%, elapsed time: 4.9 sec.

(d) $\tau = \varepsilon = 0.003$ (following [22]). Relative segmentation error: 1.5378%, elapsed time: 2.4 sec.

Figure 5.9: 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 Fig. 5.6. 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.

parameters not given in the captions of Fig. 5.9 are $\hat{\mu} = 30$, $\sigma = 35$, $k_b = 1$, $k = 1$, $\delta = 10^{-10}$, and $K = 70$.

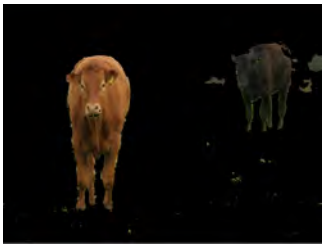
Note 35. *The regime $\tau > \varepsilon$ is not of much interest since, by* [8, Remark 4.8] mutatis mutandis*, in this regime the SDIE scheme has non-unique solution for the update, of which one is just the MBO solution.*

Comparing the results in Fig. 5.9, 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 Fig. 5.10 we compare the result of Fig. 5.9(d) (our best segmentation) with the results of the analogous experiments in [6, 22]. 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, 22] do. Note that in the case of Fig. 5.9(d) we are using the MBO scheme, so the only differences between our method and [22] are the Nyström-QR method, the Strang formula method, and the reference that we hand-drew (see Fig. 5.6, bottom-left), which is slightly different from both the reference used in [6, Figure 4.6] and the reference used in [22, Figure 2(e)].

We measure the computational cost of the SDIE scheme through the measured runtime of the respective algorithm. We note from Fig. 5.9 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 Fig. 5.11, where the SDIE scheme needs many more steps to satisfy the termination criterion. At least for this example,

(a) Segmentation from [6, Figure 4.6] (b) Segmentation from [22, Figure 2(f)]



(c) Segmentation from Fig. 5.9(d)

Figure 5.10: Comparison of our segmentation (using the set-up in Fig. 5.9(d)) with the analogous segmentations from the previous literature [6, 22], both reproduced with permission from SIAM and the authors. Note that unfortunately in reproduction the colour balances and aspect ratios have become slightly inconsistent, but we can still make qualitative comparisons.

the non-MBO SDIE scheme is less efficient than the MBO scheme. Thus, in the following sections we focus on the MBO case.

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 discretisation steps k_b and k in the Euler method and the Strang formula respectively, and the choice of normalisation 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 normalised Laplacian Δ_s.

We plot total runtimes (i.e. 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 Fig. 5.12. 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. Moving on to the errors, we observe that increasing k_b and k also does not increase the accuracy of the overall segmentation. Finally, we see that the symmetric normalised Laplacian incurs consistently

(a) $\tau = 0.001 \ll 0.003 = \varepsilon$.

(b) $\tau = 0.0025 < 0.003 = \varepsilon$.

(c) $\tau = \varepsilon = 0.003$ (following [22]).

Figure 5.11: 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 Fig. 5.9. The intermediary rows (showing the current segmentation u_n and the image masked by that segmentation) 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, i.e. the state satisfying the termination criterion, which correspond to the subfigures in Fig. 5.9. For layout reasons, we have squashed the figure in (a).

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 normalised Laplacian seem to deteriorate for large K, though it is not clear to us as to why. The random walk Laplacian has diametric properties in this regard: the segmentations are only reliably accurate when K is reasonably large.

Uncertainty in the segmentation

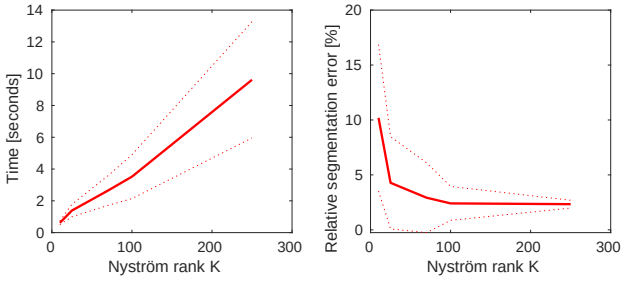
Due to the randomised Nyström approximation, our approximation of the SDIE scheme is inherently stochastic. Therefore, the segmentations that the algorithm returns are realisations of random variables. We now briefly study these random variables, especially with regard to K. We show pixelwise mean and standard deviations of the binary labels in each of the left two columns of the four subfigures of Fig. 5.13. 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 in the previous subsubsection.

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 normalisation of the Laplacian has an effect: the symmetric normalised Laplacian leads to less variation than the random walk Laplacian. Third, the parameters k_b and k appear to have no major effect for the values 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 colour 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.

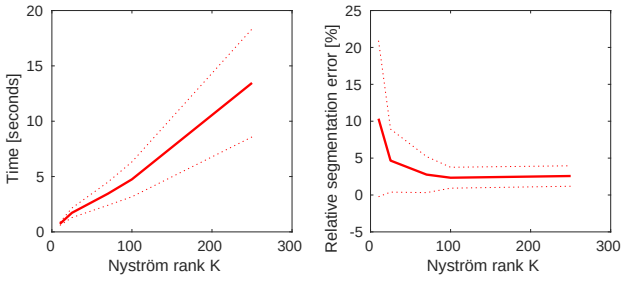
This is of course not a rigorous Bayesian uncertainty quantification, as for instance is given in [5, 26] for graph-based learning. However the use of stochastic algorithms for inference tasks, and the use of their output as a method of uncertainty quantification, has for instance been motivated by [21].

5.3.4. The greyscale example

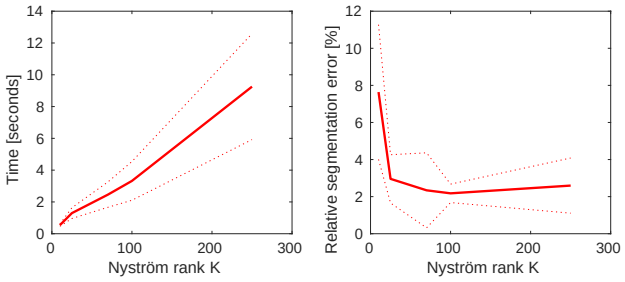
We now move on to Example 5.3.2, the greyscale problem. We will especially use this example to study the influence of the parameters $\hat{\mu}$ and σ. The parameter $\hat{\mu} > 0$ determines the strength of the fidelity term in the AC flow. From a statistical point of view, we can understand a choice of $\hat{\mu}$ as an assumption on the statistical precision (i.e. the inverse of the variance of the noise) of the reference $\tilde{f}$ (see [5, Section 3.3] for details). Thus, a small $\hat{\mu}$ should lead to a stronger regularisation 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



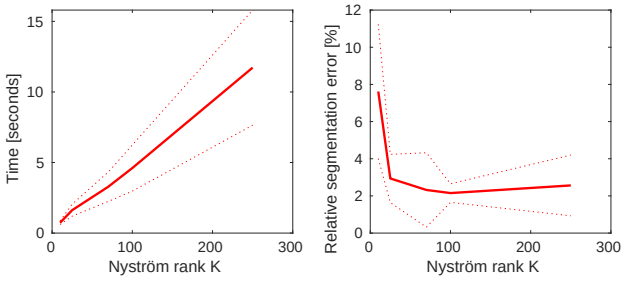
(a) $k_b = 1$, $k = 1$, random walk Laplacian.



(b) $k_b = 10$, $k = 5$, random walk Laplacian.



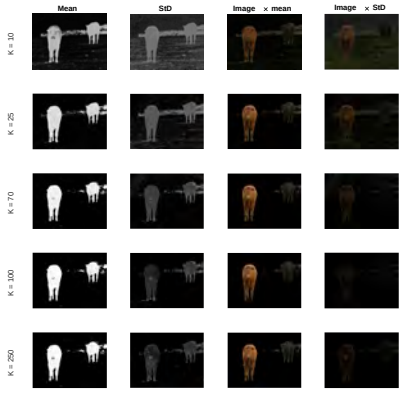
(c) $k_b = 1$, $k = 1$, symmetric normalised Laplacian.



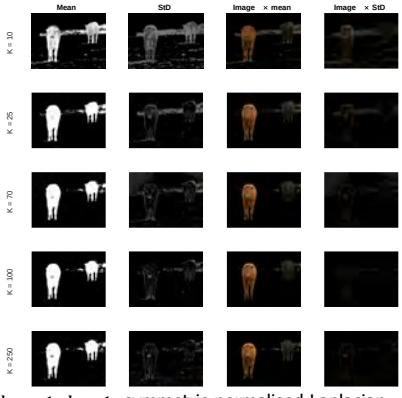
(d) $k_b = 10$, $k = 5$, symmetric normalised Laplacian.

Figure 5.12: Errors and timings of 100 independent segmentations of the two cows image (Example 5.3.1) with the MBO SDIE scheme. The solid lines represent the means averaged over 100 runs, the dotted lines show means $\pm$ standard deviations.

(a) $k_b = 1$, $k = 1$, random walk Laplacian.



(b) $k_b = 10$, $k = 5$, random walk Laplacian.

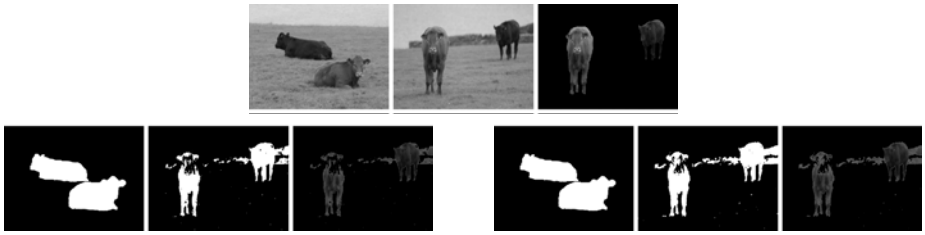


(c) $k_b = 1$, $k = 1$, symmetric normalised Laplacian.

(d) $k_b = 10$, $k = 5$, symmetric normalised Laplacian.

Figure 5.13: Mean, standard deviation, and images weighted by mean and standard deviation of 100 independent segmentations of Example 5.3.1 with the MBO SDIE scheme, with set-up as in Section 5.3.3.

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 instead of using the Nyström method this sparsity can be exploited using Rayleigh–Chebyshev [2] methods as in [22], or MATLAB sparse matrix algorithms as in [6], but this lies beyond the scope of this thesis. If σ is too large then the graph structure no longer reflects the features of the image.

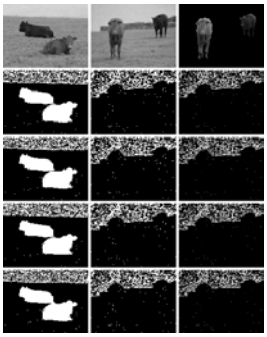


(a) $\hat{\mu} = 150, \sigma = 10^3$, error = 5.54 %, time = 6.6 sec. (b) $\hat{\mu} = 150, \sigma = 35$, error = 5.8226 %, time = 12.9 sec.

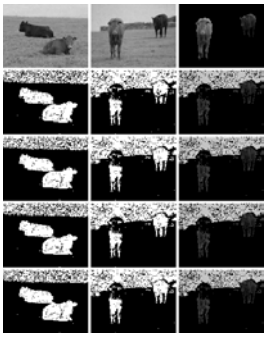
Figure 5.14: MBO SDIE segmentations for the greyscale segmentation task. In the centred top figure, we show the reference data image, the image to be segmented, and the image masked with the ground truth segmentation, cf. Fig. 5.6. The other figures show the reference $\tilde{f}$, the segmentation returned by the algorithm, and the original images masked with the segmentation.

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 5.3.3) we use the random walk Laplacian. We will qualitatively study single realisations 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 Fig. 5.14. Moreover, we give a comprehensive overview of all tests and the progression of the SDIE scheme in Figs 5.15 and 5.16. We observe that this segmentation problem is indeed considerably harder than the "two cows" problem, as we anticipated after stating Example 5.3.2. The difference in shade between the left cow and the wall is less visible than in Example 5.3.1, and the left cow's snout is less identifiable as part of the cow. Thus, the segmentation errors we incur are about 3 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'.

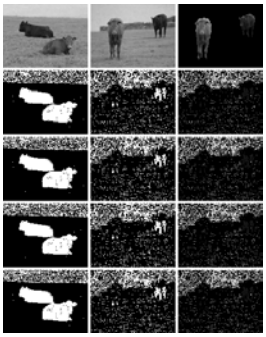
Given σ in $\{35, 10^3\}$, the strength of the fidelity term $\hat{\mu}$ has a significant impact on the result, as well. Indeed, for $\hat{\mu} = 50$ the algorithm does not find any segmentation. For $\hat{\mu} \geq 100$, we get more practical segmentations. Interestingly, for $\hat{\mu} = 150$ and $\hat{\mu} = 200$ 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 refer-



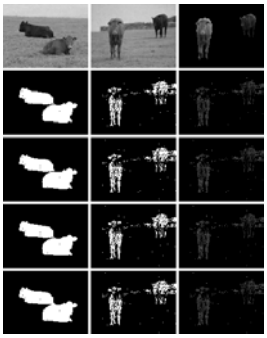
(a) $\hat{\mu} = 50, \sigma = 2$,
error: 25.2741%, time: 15.4 sec.



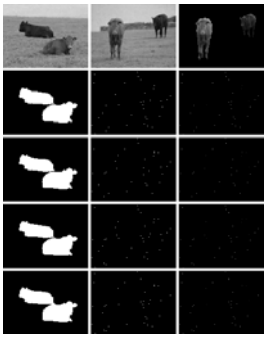
(b) $\hat{\mu} = 100, \sigma = 2$,
error: 27.098%, time: 15.6 sec.



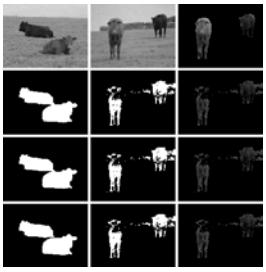
(c) $\hat{\mu} = 150, \sigma = 2$,
error: 28.0225%, time: 14.7 sec.



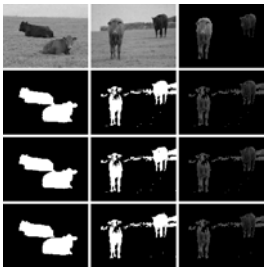
(d) $\hat{\mu} = 200, \sigma = 2$,
error: 8.013%, time: 15.1 sec.



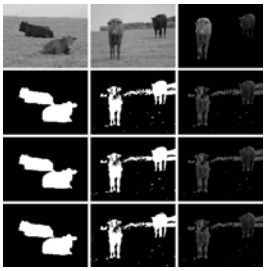
(e) $\hat{\mu} = 50, \sigma = 35$,
error: 13.0137%, time: 14.5 sec.



(f) $\hat{\mu} = 100, \sigma = 35$,
error: 5.9385%, time: 6.7 sec.

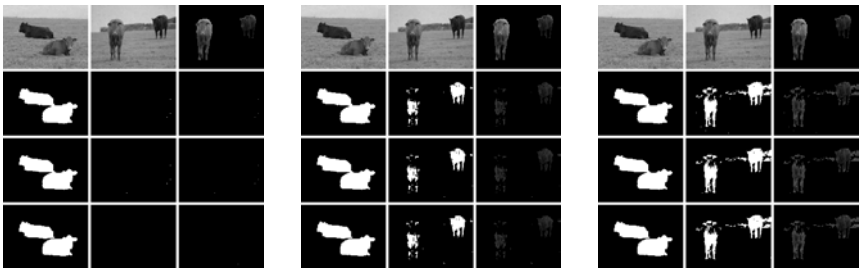


(g) $\hat{\mu} = 150, \sigma = 35$,
error: 5.8226%, time: 12.9 sec.



(h) $\hat{\mu} = 200, \sigma = 35$,
error: 6.2301%, time: 6.7 sec.

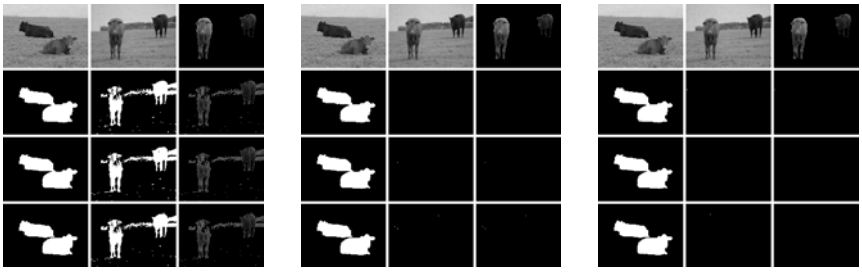
Figure 5.15: Progression of the MBO SDIE scheme for the *greyscale* segmentation task. In each subfigure: The first row shows the reference data, the image to be segmented, and the ground truth segmentation. The middle rows, showing the reshaped label vector u_n and the image masked by the label, each represent one iteration of the considered algorithm, to be read from top to bottom. The last row gives the state returned by the scheme, i.e. the state satisfying the termination criterion.



(a) $\hat{\mu} = 50, \sigma = 10^3$, error: 12.8148%, time: 6.7 sec.

(b) $\hat{\mu} = 100, \sigma = 10^3$, error: 6.5934%, time: 6.6 sec.

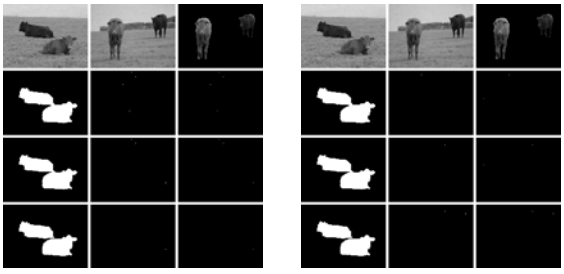
(c) $\hat{\mu} = 150, \sigma = 10^3$, error: 5.54%, time: 6.6 sec.



(d) $\hat{\mu} = 200, \sigma = 10^3$, error: 6.3376%, time: 6.6 sec.

(e) $\hat{\mu} = 50, \sigma = 10^4$, error: 12.8089%, time: 6.5 sec.

(f) $\hat{\mu} = 100, \sigma = 10^4$, error: 12.8092%, time: 6.6 sec.



(g) $\hat{\mu} = 150, \sigma = 10^4$, error: 12.8122%, time: 6.5 sec.

(h) $\hat{\mu} = 2000, \sigma = 10^4$, error: 12.8171%, time: 6.4 sec.

Figure 5.16: Continuation of Fig. 5.15.

ence is less precise than for larger $\hat{\mu}$, leading us (due to the smoothing effect of the Ginzburg–Landau regularisation) 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: typical was between 6 and 7 seconds, but a number of cases took twice that time. By doing an additional step of the MBO scheme before converging, the times for $\sigma = 2$ as well as $(\hat{\mu}, \sigma) = (50, 35)$ were significantly increased. We also see a larger runtime in the case of $(\hat{\mu}, \sigma) = (150, 35)$.

5.3.5. The "many cows" example

(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 5.17: Segmentations via 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, for comparison with the bottom row, which shows the reference $\tilde{f}$, the segmentation returned by the respective algorithm, and the original image masked with the segmentation.

We finally study the "many cows" example, i.e. Example 5.3.3. The main differences to the earlier examples are the larger size of the image that is to be segmented (see Fig. 5.8) and the variety of the features 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 would require 17.6 TB of memory to be constructed in MATLAB R2019a, if we were constructing the full matrix and not using the Nyström-QR method to compress it. Furthermore, 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, e.g. 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 Fig. 5.17(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 Fig. 5.17(b,c): 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 recognised 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.

5.4. Conclusion

We have here demonstrated how to use the SDIE scheme to solve classification problems. We have furthermore provided an algorithm (Algorithms 1 and 2) 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 [22] algorithm for graph MBO with fidelity forcing: it implements the Nyström extension via a QR decomposition (Algorithm 1) and it replaces the Euler discretisation of the diffusion step with a computation based on the Strang formula for matrix exponentials (see (5.11)). 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 stable than the Nyström method used in previous literature [6, 22], and it is less conceptually troubling than that method, as it does not involve taking the square root of a matrix which is never positive semi-definite.

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 whilst 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, 22]. We furthermore investigated empirically various properties of this numerical method and the role of different parameters. In particular:

- We found that the symmetric normalised Laplacian incurred consistently low segmentation error when approximated to a low rank, whilst 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 scalabity, and thus very low-rank approximations, we recommend using the symmetric normalised Laplacian.

- We investigated empirically the uncertainty inherited from the randomisation in the Nyström extension. We found that the rank reduction and the normalisation 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, e.g. 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 5.3.2 and 5.3.3, it was particularly important to tune this parameter to get an accurate segmentation.

Bibliography

[1] Dominik Alfke et al. "NFFT Meets Krylov Methods: Fast Matrix-Vector Products for the Graph Laplacian of Fully Connected Networks". In: *Frontiers in Applied Mathematics and Statistics* 4 (2018), p. 61. issn: 2297-4687. doi: 10.3389/fams.2018.00061. url: https://www.frontiersin.org/article/10.3389/fams.2018.00061.

[2] Christopher R. Anderson. "A Rayleigh-Chebyshev Procedure for Finding the Smallest Eigenvalues and Associated Eigenvectors of Large Sparse Hermitian Matrices". In: *J. Comput. Phys.* 229.19 (Sept. 2010), pp. 7477–7487. issn: 0021-9991. doi: 10.1016/j.jcp.2010.06.030.

[3] M. Bebendorf and S. Kunis. "Recompression techniques for adaptive cross approximation". In: *Journal of Integral Equations and Applications* 21.3 (2009), pp. 331–357. doi: 10.1216/JIE-2009-21-3-331.

[4] J. Bence, B. Merriman, and S. Osher. "Diffusion generated motion by mean curvature". In: *CAM Report, 92-18, Department of Mathematics, University of California, Los Angeles* (1992).

[5] A. Bertozzi et al. "Uncertainty Quantification in Graph-Based Classification of High Dimensional Data". In: *SIAM/ASA J. Uncertain. Quantification* 6 (2018), pp. 568–595.

[6] Andrea Bertozzi and Arjuna Flenner. "Diffuse Interface Models on Graphs for Classification of High Dimensional Data". In: *Multiscale Modeling Simulation* 10 (July 2012), pp. 1090–1118. doi: 10.1137/11083109X.

[7] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. "A Review of Image Denoising Algorithms, with a New One". In: *SIAM Journal on Multiscale Modeling and Simulation* 4.2 (Jan. 2005), pp. 490–530. doi: 10.1137/040616024.

[8] Jeremy Budd and Yves van Gennip. "Graph Merriman–Bence–Osher as a SemiDiscrete Implicit Euler Scheme for Graph Allen–Cahn Flow". In: *SIAM Journal on Mathematical Analysis* 52 (Jan. 2020), pp. 4101–4139. doi: 10.1137/19M1277394.

[9] Jeremy Budd, Yves van Gennip, and Jonas Latz. "Classification and image processing with a semi-discrete scheme for fidelity forced Allen–Cahn on graphs". English. In: *GAMM Mitteilungen* 44.1 (2021), pp. 1–43. issn: 0936-7195. doi: 10.1002/gamm.202100004.

[10] Luca Calatroni et al. "Graph Clustering, Variational Image Segmentation Methods and Hough Transform Scale Detection for Object Measurement in Images". In: *Journal of Mathematical Imaging and Vision* 57 (Feb. 2017), pp. 269–291. doi: 10.1007/s10851-016-0678-0.

[11] T.F. Chan and L.A. Vese. "Active contours without edges". In: *IEEE Transactions on Image Processing* 10.2 (2001), pp. 266–277. doi: 10.1109/83.902291.

[12] J.R. Dormand and P.J. Prince. "A family of embedded Runge-Kutta formulae". In: *Journal of Computational and Applied Mathematics* 6.1 (1980), pp. 19–26. issn: 0377-0427. doi: 10.1016/0771-050X(80)90013-3. url: https://www.sciencedirect.com/science/article/pii/0771050X80900133.

[13] Carl Eckart and Gale Young. "The approximation of one matrix by another of lower rank". In: *Psychometrika* 1.3 (1936), pp. 211–218. url: https://EconPapers.repec.org/RePEc:spr:psycho:v:1:y:1936:i:3:p:211-218.

[14] Selim Esedoglu and Yen-Hsi Richard Tsai. "Threshold dynamics for the piecewise constant Mumford–Shah functional". In: *Journal of Computational Physics* 211.1 (2006), pp. 367–384. issn: 0021-9991. doi: 10.1016/j.jcp.2005.05.027. url: https://www.sciencedirect.com/science/article/pii/S0021999105002792.

[15] Charless Fowlkes et al. "Spectral Grouping Using the Nyström Method". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 26.2 (Jan. 2004), pp. 214–225. issn: 0162-8828. doi: 10.1109/TPAMI.2004.1262185.

[16] Yves van Gennip and Andrea L. Bertozzi. *Gamma-convergence of graph Ginzburg-Landau functionals*. 2018. arXiv: 1204.5220 [math.AP].

[17] Gene H. Golub and Charles F. van Loan. *Matrix Computations*. 4th ed. Baltimore: Johns Hopkins University Press, 2013. isbn: 9781421407944. url: http://www.cs.cornell.edu/cv/GVL4/golubandvanloan.htm.

[18] N. Halko, P. Martinsson, and J. Tropp. "Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions". In: *SIAM Rev.* 53 (2011), pp. 217–288.

[19] Brian C. Hall. *Lie Groups, Lie Algebras, and Representations*. 2nd ed. Vol. 222. Graduate Texts in Mathematics. New York: Springer International Publishing, 2015. isbn: 978-3-319-13466-6. doi: 10.1007/978-3-319-13467-3.

[20] Robert V. Kohn and Peter Sternberg. "Local minimisers and singular perturbations". In: *Proceedings of the Royal Society of Edinburgh: Section A Mathematics* 111.1-2 (1989), pp. 69–84. doi: 10.1017/S0308210500025026.

[21] Stephan Mandt, Matthew D. Hoffman, and David M. Blei. "Stochastic Gradient Descent as Approximate Bayesian Inference". In: *J. Mach. Learn. Res.* 18.1 (Jan. 2017), pp. 4873–4907. issn: 1532-4435.

[22] Ekaterina Merkurjev, Tijana Kostić, and Andrea Bertozzi. "An MBO Scheme on Graphs for Classification and Image Processing". In: *SIAM Journal on Imaging Sciences* 6 (Oct. 2013), pp. 1903–1910. doi: 10.1137/120886935.

[23] Luciano Modica and Stefano Mortola. "Un esempio di Γ^--convergenza". In: *Bollettino della Unione Matematica Italiana. Series V. B* 14 (Jan. 1977), pp. 285–299.

[24] David Mumford and Jayant Shah. "Optimal approximations by piecewise smooth functions and associated variational problems". In: *Communications on Pure and Applied Mathematics* 42.5 (1989), pp. 577–685. doi: 10.1002/cpa.3160420503. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.3160420503. url: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160420503.

[25] E. J. Nyström. "Über Die Praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben". In: *Acta Mathematica* 54.none (1930), pp. 185–204. doi: 10.1007/BF02547521.

[26] Yiling Qiao et al. "Uncertainty quantification for semi-supervised multi-class classification in image processing and ego-motion analysis of body-worn videos". In: *Electronic Imaging* 2019.11 (2019), pp. 264-1-264–7. issn: 2470-1173. doi: doi:10.2352/ISSN.2470-1173.2019.11.IPAS-264. url: https://www.ingentaconnect.com/content/ist/ei/2019/00002019/00000011/art00015.

[27] Gilbert Strang. "On the Construction and Comparison of Difference Schemes". In: *SIAM Journal on Numerical Analysis* 5.3 (1968), pp. 506–517. issn: 00361429. url: http://www.jstor.org/stable/2949700.

[28] Manik Varma and Andrew Zisserman. "A Statistical Approach to Texture Classification from Single Images". In: *International Journal of Computer Vision* 62 (Apr. 2005), pp. 61–81. doi: 10.1007/s11263-005-4635-4.

[29] Marianne de Vriendt, Philip Sellars, and Angelica I Aviles-Rivero. "The GraphNet Zoo: An All-in-One Graph Based Deep Semi-supervised Framework for Medical Image Classification". In: *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Graphs in Biomedical Image Analysis*. Springer, 2020, pp. 187–197.

[30] Max A. Woodbury. "Inverting modified matrices". In: *Department of Statistics, Princeton University* (1950).

[31] Leonid Yaroslavsky. *Digital Picture Processing*. 1st ed. Vol. 9. Springer Series in Information Sciences. Berlin: Springer-Verlag, 1985. isbn: 978-3-642-81931-5. doi: 10.1007/978-3-642-81929-2.

[32] Haruo Yoshida. "Construction of higher order symplectic integrators". In: *Physics Letters A* 150.5 (1990), pp. 262–268. issn: 0375-9601. doi: 10.1016/0375-9601(90)90092-3. url: https://www.sciencedirect.com/science/article/pii/0375960190900923.

[33] Lihi Zelnik-Manor and Pietro Perona. "Self-Tuning Spectral Clustering". In: *Proceedings of the 17th International Conference on Neural Information Processing Systems*. NIPS'04. Vancouver, British Columbia, Canada: MIT Press, 2004, pp. 1601–1608.

6

Joint Reconstruction-Segmentation on Graphs

Real data is messy. [...] It's all very, very noisy out there. Very hard to spot the tune. Like a piano in the next room, it's playing your song, but unfortunately it's out of whack, some of the strings are missing, and the pianist is tone deaf and drunk — I mean, the noise! Impossible!

Tom Stoppard, *Arcadia*

In practice, image segmentation tasks also require the image to be reconstructed *from noisy, distorted, or incomplete observations. A recent approach for solving such tasks is to perform this reconstruction jointly with the segmentation, using each to guide the other. Past work on this has employed relatively simple segmentation algorithms, such as the Chan–Vese algorithm* [9]. *In this chapter, we present a foundation for performing joint reconstruction-segmentation using the graph-based segmentation method of chapter 5. We consider the joint reconstruction-segmentation task as a minimisation problem, which we attempt to solve via an iterative minimisation scheme which alternately refines the reconstruction (using the current segmentation) and refines the segmentation (using the current reconstruction). As in chapter 5, complications arise due to the large size of the matrices involved, and we*

Parts of this chapter to appear in J. Budd, Y. van Gennip, J. Latz, S. Parisotto, and C.-B. Schönlieb, "Joint reconstruction-segmentation on graphs", *in preparation*. Jonas Latz and Simone Parisotto contributed significantly to this chapter.

show how these complications can be managed. We then exhibit some simple tests, applying the scheme to a noised version of the "two cows" example from chapter 5. Finally, we discuss a number of directions for future work building on this foundation.

6.1. Introduction

In practice, we often do not observe images directly, but rather observe data that is noisy, may have pieces missing, and/or may be a transform of the true image (e.g. in a CT or MRI scan). In such cases, one must "reconstruct" the image from these observations. Hence, when we seek to segment an image, we will frequently need to also reconstruct it. That is, in practice a segmentation task is often a *reconstruction*-segmentation task. Joint reconstruction-segmentation is a powerful approach for solving such tasks. It performs the reconstruction and segmentation together, using each to guide the other, with the goal of improving the quality of the reconstruction/segmentation compared to performing the tasks in sequence. In this chapter, we will develop a foundation for incorporating the graph-based segmentation method of chapter 5 into this technique.

6.1.1. Background

In the last chapter, we examined the task of image segmentation. *Image reconstruction* is another fundamental task in image processing, perhaps the most fundamental. The general setting for image reconstruction is that we have some observations y of an image x^*, which are related via

$$y = \mathcal{T}(x^*) + e \tag{6.1}$$

where $\mathcal{T}$ is the *forward model*, typically a linear map, and e is an error term (e.g. a Gaussian random variable). For example, in MRI imaging the map $\mathcal{T}$ is a sampling of the Fourier transform [11]. Even given y, $\mathcal{T}$, and the distribution of e, solving this equation for x^* is in general an ill-posed problem. A key approach to solving such problems, deriving from pioneering work by Tikhonov [24] and Phillips [19] in the 1960s, has been to solve a variational problem of the form

$$\operatorname*{argmin}_x R(x) + \mu D(\mathcal{T}(x), y) \tag{6.2}$$

where R is a *regulariser*, which encodes *a priori* information about the solution x, and D is a distance term which enforces fidelity to our observed data and encodes information about the error e. A considerable amount of work has been devoted to the choice of regulariser R. To give a whistle-stop tour: Tikhonov used an ℓ^2 norm, which was superseded by Rudin, Osher, and Fatemi's [22] use of total variation in the 1990s, which recently inspired more sophisticated regularisers such as total generalised variation [16], and finally the cutting-edge approach is to use data-driven *learned regularisers* such as those discussed in Arridge *et al.* [3]. Solving (6.2) can be computationally challenging. Some examples of approaches include: Euler–Lagrange methods [5] (and the references therein), duality-based methods [8, 16], split-Bregman methods [12], accelerated proximal gradient methods [7, 27], alternating direction method of multipliers (ADMM) methods [25, 26], and data-driven optimisation methods (see [3, §4.9] for a detailed overview).

Joint reconstruction-segmentation lies in the middle of two different approaches to reconstruction-segmentation, i.e. the task of obtaining a segmentation u of an image x^*, from observations y of the form (6.1).

At one extreme is the traditional *sequential* approach, in which one first reconstructs an image $x \approx x^*$ from y via the above methods, and then one performs a segmentation of x. The drawback of this method is that it doesn't make full use of all the resources at hand, since the reconstruction step is blind to any segmentation-relevant information. Furthermore the reconstruction methods might induce a loss of contrast [23] that might interfere with achieving the best segmentation.

At the other extreme is the *end-to-end* approach, in which one collects training data $\{(y_n, u_n)\}$ of pairs of observations and the corresponding ground truth segmentations, and then using this training data one learns (e.g., via deep learning[1]) a map that sends y to u. The drawbacks of this approach are that it forgoes entirely reconstructing an explicit x^*, and that this map may be somewhat of a "black box", i.e. it may be hard to explain why it outputs the segmentation that it does or to prove theoretical guarantees.

Joint reconstruction-segmentation (a.k.a. simultaneous reconstruction and segmentation) lies between these extremes, seeking to perform the reconstruction and segmentation simultaneously, using each to guide the other. It was first proposed by Ramlau and Ring [21] in 2007 for Computed Tomography (CT) imaging, with related methods later developed for other medical imaging tasks (for an overview, see [10, §2.4] and the references therein). The methods employed in this initial work were extremely varied, and largely application-driven. An extensive theoretical overview of *task-adapted reconstruction*, of which reconstruction-segmentation is a special case, was developed in Adler *et al.* [1]. That work also found that joint reconstruction-segmentation outperformed both the sequential and end-to-end approaches with respect to segmentation accuracy. In recent work by Corona *et al.* [10] this method was enhanced using Bregman iteration methods, and a number of theoretical guarantees were proved about this enhanced scheme. However, these approaches have mostly relied on Mumford–Shah or Chan–Vese methods to perform the segmentation.[2] In this chapter, we will demonstrate how to incorporate the graph-based segmentation methods discussed in the previous chapter into this joint reconstruction-segmentation technique.

6.1.2. Groundwork

We express the reconstruction-segmentation task that we shall be attempting to solve as follows.

Problem 6.1.1. *Let $x^* : Y \to \mathbb{R}^\ell$ be the image to be reconstructed and segmented, $y = \mathcal{T}(x^*) + e$ be* observed data *where $\mathcal{T} \in C^1$ is the* forward model *and e is some random variable which describes observation error, and let $x_d : Z \to \mathbb{R}^\ell$ be an already reconstructed and segmented* reference image *with* a priori *segmentation $\tilde{f} : Z \to \{0, 1\}$. Given y, $\mathcal{T}$, x_d, and $\tilde{f}$: reconstruct $x \approx x^*$ and find $u : Y \cup Z \to \{0, 1\}$ such that $u|_Y$ segments x and $u|_Z$ is close to $\tilde{f}$.*

We incorporate this into a graph framework, as in section 5.2.1. We will define our vertex set to be $V := Y \cup Z$ and our edge set by $ij \in E$ if and only if $i \neq j$.

[1] See Goodfellow, Bengio, and Courville [14] for an overview of deep learning.
[2] See Mumford and Shah [17] and Chan and Vese [9] for details on these methods.

We will also define $N := |Y|$ and $N_d := |Z|$. Then, given a candidate reconstruction $x : Y \to \mathbb{R}^\ell$, we define the weights on the edges of this graph by defining *feature vectors* $z : Y \to \mathbb{R}^q$ and $z_d : Z \to \mathbb{R}^q$, according to linear maps $z := \mathcal{F}(x)$ and $z_d := \mathcal{F}_d(x_d)$ where $\mathcal{F}$ and $\mathcal{F}_d$ will be fixed and known. Since x_d and $\mathcal{F}_d$ are given, we can hereafter treat z_d as given. We will then define the edge weights via $\omega = \Omega(z, z_d)$, where $\Omega(z, z_d)$ is given by (for $\mathbf{z} := (z, z_d)$)

$$\Omega_{ij}(z, z_d) := e^{-\frac{||\mathbf{z}_i - \mathbf{z}_j||_F^2}{q\sigma^2}} \tag{6.3}$$

where $||\cdot||_F$ denotes the Frobenius norm. The q in the denominator averages over the q components of $\mathbf{z}$ so as to make a parameter choice for σ generalise better.

Note 36. *The feature vectors z and z_d are here defined so that z does not depend on x_d and* vice versa*. This is a simplification, since x and x_d might be different parts of the same image and hence one might want z to partially depend on x_d. However, this simplification greatly aids in the following analysis, and in computation, as it means that the edge weights between vertices of Z can be considered fixed and given.*

We shall be using the Ginzburg–Landau energy from (3.9) to measure the adequacy of our segmentation, and following the previous chapter we will be taking $r = 1$. Over the course of the reconstruction, the candidate image x and therefore the weight matrix ω will be updated, and so it will be useful to reconsider $\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}$ as a function of both u and ω. A simple calculation gives that when $r = 1$

$$\begin{aligned}\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u, \omega) = &\frac{1}{2}\sum_{i,j\in V}\omega_{ij}(u_i - u_j)^2 + \frac{1}{2\varepsilon}\sum_{i,j\in V}\omega_{ij}(W(u_i) + W(u_j)) \\ &+ \frac{1}{4}\sum_{i,j\in V}\omega_{ij}\left(\mu_i(u_i - \tilde{f}_i)^2 + \mu_j(u_j - \tilde{f}_j)^2\right)\end{aligned} \tag{6.4}$$

where $\mu \in \mathcal{V}\setminus\{\mathbf{0}\}$ is supported on Z. Note that this is linear in ω. We can therefore define $G : \mathcal{V} \to \mathbb{R}^{V\times V}$ given by

$$(G(u))_{ij} = \frac{1}{2}\left((u_i - u_j)^2 + \frac{1}{\varepsilon}(W(u_i) + W(u_j)) + \frac{1}{2}(\mu_i(u_i - \tilde{f}_i)^2 + \mu_j(u_j - \tilde{f}_j)^2)\right) \tag{6.5}$$

such that

$$\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u, \omega) = \mathrm{tr}(G(u)^T\omega) =: \langle G(u), \omega\rangle_F$$

where $\langle\cdot,\cdot\rangle_F$ denotes the Frobenius inner product. Furthermore, note that if $v_i := \frac{1}{2}u_i^2 + \frac{1}{2\varepsilon}W(u_i) + \frac{1}{4}\mu_i(u_i - \tilde{f}_i)^2$, then

$$G(u) = -uu^T + v\mathbf{1}^T + \mathbf{1}v^T.$$

6.1.3. The iPiano method

In order to update our reconstruction in (6.9a), it will be necessary to solve problems of the form

$$\operatorname*{argmin}_x R(x) + \mathscr{F}(x)$$

where R is convex (and proper and lower semi-continuous) but non-smooth, and $\mathscr{F}$ is smooth but non-convex. We will follow Ochs *et al.* [18, Algorithm 4] and use the iPiano method to solve such problems. This finds such a minimiser via the following iterative scheme.

Let x'_0 be some initial guess, and let $\xi \in [0,1)$ be fixed. Then given x'_k,

$$x'_{k+1} := \operatorname{prox}_{\theta_k R}(x'_k - \theta_k \nabla_x \mathscr{F}(x'_k) + \xi(x'_k - x'_{k-1}))^3 \tag{6.6}$$

where $\theta_k < 2(1-\xi)L_k^{-1}$, and L_k is the least value in $\{bL_{k-1}, abL_{k-1}, a^2bL_{k-1}, \ldots\}$ such that

$$\mathscr{F}(x'_{k+1}) \leq \mathscr{F}(x'_k) + \langle \nabla_x \mathscr{F}(x'_k), x'_{k+1} - x'_k \rangle_F + \frac{1}{2} L_k ||x'_{k+1} - x'_k||_F^2 \tag{6.7}$$

for some fixed $a > 1$ and $b \geq 1$. Note that since x'_{k+1} depends on L_k, to implement this involves *backtracking*. That is, we compute a candidate x'_{k+1} with the smallest candidate L_k, check (6.7), and if that fails we repeat with the next candidate L_k. This process to pick L_k must eventually terminate, since if L is our candidate L_k, then as $L \to \infty$, $\theta_k \to 0$, and so our candidate x'_{k+1} tends to $x'_k + \xi(x'_k - x'_{k-1})$. Since $\mathscr{F}$ is smooth, there exists an L such that for all x' with $||x' - x'_k||_F \leq 2\xi||x'_k - x'_{k-1}||_F$,

$$\mathscr{F}(x') \leq \mathscr{F}(x'_k) + \langle \nabla_x \mathscr{F}(x'_k), x' - x'_k \rangle_F + \frac{1}{2} L ||x' - x'_k||_F^2.$$

In practice, we will terminate the iPiano scheme if $L_k > 10^{10}$, and output the current x'_k as the minimiser.

Note that to use this method, we need to able to compute both $\nabla_x \mathscr{F}$ and $\mathscr{F}$.

6.2. The joint reconstruction-segmentation scheme

We consider the joint minimisation problem

$$\min_{x \in \mathbb{R}^{N \times \ell}, u \in \mathcal{V}} R(x) + \alpha ||\mathcal{T}(x) - y||_F^2 + \beta \operatorname{GL}_{\varepsilon,\mu,\tilde{f}}(u, \Omega(\mathcal{F}(x), z_d)) \tag{6.8}$$

where R is a convex regulariser. Given the number of moving parts in this, we consider an iterative scheme to approach solutions (where $\alpha, \beta, \eta_n, \nu_n$ are parameters):

$$x_{n+1} = \operatorname*{argmin}_{x \in \mathbb{R}^{N \times \ell}} R(x) + \alpha ||\mathcal{T}(x) - y||_F^2 + \beta \operatorname{GL}_{\varepsilon,\mu,\tilde{f}}(u_n, \Omega(\mathcal{F}(x), z_d)) + \eta_n ||x - x_n||_F^2, \tag{6.9a}$$

$$u_{n+1} = \operatorname*{argmin}_{u \in \mathcal{V}} \beta \operatorname{GL}_{\varepsilon,\mu,\tilde{f}}(u, \Omega(\mathcal{F}(x_{n+1}), z_d)) + \nu_n ||u|_Y - u_n|_Y||_2^2. \tag{6.9b}$$

[3] Recall that if f is proper, lower semi-continuous, and convex, then $\operatorname{prox}_f(y) := \operatorname{argmin}_x f(x) + \frac{1}{2}||x - y||_2^2$.

We can understand this scheme intuitively as iterating the following steps:

I. Given the current segmentation, update the reconstruction using the segmentation energy as part of the regulariser and the previous reconstruction as a momentum term.

II. Given the current reconstruction, update the segmentation using the previous segmentation of the image to be reconstructed as a momentum term.

It is important to discuss what (6.9a) is doing in a bit more detail. There are two goals we might have for solving this joint reconstruction-segmentation problem. One is to achieve both a more accurate reconstruction and more accurate segmentation than could be achieved by doing these tasks in sequence. However, a second goal is simply to achieve a more accurate segmentation of the image, without interest in the reconstruction.

In the latter case, it will be advantageous for our candidate reconstructions x_n to have higher contrast than the ground truth x^*, and in particular for the difference between the pixels belonging to different segments to be exaggerated. This behaviour is encouraged by the middle term in (6.9a). That is the Ginzburg–Landau term penalises x via penalising ω_{ij} being large (i.e. $||\boldsymbol{z}_i - \boldsymbol{z}_j||_F$ being small) when $(G(u_n))_{ij}$ is large. And one of the main ways for $(G(u_n))_{ij}$ to be large is for i and j to be put in different segments by u_n.[4] That is, minimising that term rewards images where the pixels put into different segments by u_n have more distinct features.

Therefore, the β parameter will govern how strong this segmentation-driven increase in contrast is. If one desires an accurate reconstruction as well as segmentation, one will therefore want β to be lower than if one desires only an accurate segmentation.

6.2.1. Initialisation

The simplest choice for the initial condition x_0 would be to choose it to be the (Moore–Penrose) pseudoinverse of $\mathcal{T}$ (see [13, §5.5.2]) applied to y, i.e. $x_0 := \mathcal{T}^+(y)$. However, in practice we have found that this can be too poorly structured to give a good initial segmentation, and furthermore it can be highly sensitive to small changes/errors in y and $\mathcal{T}$ [13, §5.5.3] and doesn't generalise to non-linear $\mathcal{T}$. Therefore, an alternative initialisation would be to initialise with some cheap reconstruction $x_0 := \mathrm{Recon}(y, \mathcal{T})$. Then the initial segmentation u_0 is constructed by applying the SDIE scheme to x_0 as in the previous chapter.

[4] The other ways are for u_n to send i or j to a non-binary value; for i in Z, for $(u_n)_i$ to disagree with $\tilde{f}_i$; and for $j \in Z$, for $(u_n)_j$ to disagree with $\tilde{f}_j$.

6.2.2. Solving (6.9b)

We solve (6.9b) via the SDIE scheme. We can rewrite the functional that is to be minimised as

$$\begin{aligned}&\mathrm{GL}_\varepsilon(u,\Omega(\mathcal{F}(x_{n+1}),z_d))+\frac{1}{2}\sum_{i\in Z}\mu_i(u_i-\tilde{f}_i)^2+\frac{1}{2}\frac{2\nu_n}{\beta}||u_Y-(u_n)_Y||_2^2\\&=\mathrm{GL}_\varepsilon(u,\Omega(\mathcal{F}(x_{n+1}),z_d))+\frac{1}{2}\sum_{i\in V}\mu'_i(u_i-\tilde{f}'_i)^2\end{aligned}$$

where $\mu' = \mu + 2\nu_n\beta^{-1}\chi_Y$ and $\tilde{f}' = \tilde{f} + u_n \odot \chi_Y$. We then solve this using the algorithm from the last chapter, with μ' and $\tilde{f}'$ in place of that chapter's "μ" and "$\tilde{f}$". Conceptually, this uses the previous segmentation as a reference, with our confidence in those previous labels encoded by the momentum parameter ν_n scaled by β.

6.3. Solving (6.9a)

To solve (6.9a), we first rewrite it. Define $G_n := G(u_n)$ and v_n by $(v_n)_i := \frac{1}{2}(u_n)_i^2 + \frac{1}{2\varepsilon}W((u_n)_i) + \frac{1}{4}\mu_i((u_n)_i - \tilde{f}_i)^2$, so that

$$G_n = -u_n u_n^T + v_n\mathbf{1}^T + \mathbf{1}v_n^T. \tag{6.10}$$

Then since ω_{ZZ} depends only on z_d, and is hence constant in n, the Ginzburg–Landau energy term can be rewritten

$$\langle G_n, \Omega(\mathcal{F}(x), z_d)\rangle_F \simeq \langle (G_n)_{YY}, \Omega_{YY}(\mathcal{F}(x))\rangle_F + 2\langle (G_n)_{YZ}, \Omega_{YZ}(\mathcal{F}(x), z_d)\rangle_F$$

so we seek to minimise via iPiano the energy:

$$R(x)+\underbrace{\underbrace{\beta\langle (G_n)_{YY}, \Omega_{YY}(\mathcal{F}(x))\rangle_F}_{=:\mathscr{F}_1(\mathcal{F}(x))} + \underbrace{2\beta\langle (G_n)_{YZ}, \Omega_{YZ}(\mathcal{F}(x), z_d)\rangle_F}_{=:\mathscr{F}_2(\mathcal{F}(x))} + \underbrace{\alpha||\mathcal{T}(x)-y||_F^2+\eta_n||x-x_n||_F^2}_{=:\mathscr{F}_3(x)}}_{=:\mathscr{F}(x)}.$$

To this end, we need to be able to compute $\nabla_x\mathscr{F}(x)$ and $\mathscr{F}(x)$.

6.3.1. Computing the gradient

Recall that the features z are given by $z := \mathcal{F}(x)$. Then

$$\nabla_x\mathscr{F}(x) = \nabla_x\mathscr{F}_1(\mathcal{F}(x))+\nabla_x\mathscr{F}_2(\mathcal{F}(x))+\nabla_x\mathscr{F}_3(x) = \mathcal{F}^*(\nabla_z\mathscr{F}_1(z))+\mathcal{F}^*(\nabla_z\mathscr{F}_2(z))+\nabla_x\mathscr{F}_3(x)$$

where $\mathcal{F}^*$ is the adjoint of $\mathcal{F}$ with respect to $\langle\cdot,\cdot\rangle_F$. We compute each term in turn.

Computing $\nabla_z \mathscr{F}_1(z)$

Expanding around z

$$\begin{aligned}\mathscr{F}_1(z+\delta z) &= \beta\langle((G_n)_{YY}, \Omega_{YY}(z+\delta z))\rangle_F \\ &= \beta\left\langle((G_n)_{YY}, \Omega_{YY}(z) + [\langle\nabla_z\Omega_{ij}(z), \delta z\rangle_F]_{i,j\in Y}\right\rangle_F + o(\delta z) \\ &= \mathscr{F}_1(z) + \beta\sum_{i,j\in Y}(G_n)_{ij}\sum_{l\in Y, r=1}^{q}(\nabla_z\Omega_{ij}(z))_{lr}\delta z_{lr} + o(\delta z)\end{aligned}$$

and thus for all $l \in Y$ and $r \in \{1, \ldots, q\}$

$$(\nabla_z \mathscr{F}_1(z))_{lr} = \beta\sum_{i,j\in Y}(G_n)_{ij}(\nabla_z\Omega_{ij}(z))_{lr}.$$

Now, since $(\Omega_{YY})_{ij}(z) = e^{-||z_i - z_j||_2^2/q\sigma^2}$ for $i \neq j$ and $(\Omega_{YY})_{ii}(z) = 0$, we have

$$\begin{aligned}\nabla_{z_{lr}}(\Omega_{YY})_{ij}(z) &= \frac{1}{q\sigma^2}\begin{Bmatrix}0, & l \notin \{i,j\}, \\ 2(\Omega_{YY})_{il}(z)(z_{ir} - z_{lr}), & j = l, \\ 2(\Omega_{YY})_{lj}(z)(z_{lr} - z_{jr}), & i = l\end{Bmatrix} \\ &= \frac{2}{q\sigma^2}(\Omega_{YY})_{ij}(z)(z_{ir} - z_{jr})(\delta_{jl} - \delta_{il}).\end{aligned}$$

Therefore

$$(\nabla_z \mathscr{F}_1(z))_{lr} = \frac{2\beta}{q\sigma^2}\sum_{i,j\in Y}(G_n)_{ij}\Omega_{ij}(z)(z_{ir} - z_{jr})(\delta_{jl} - \delta_{il}).$$

Hence, letting $\mathcal{A}(z) := (G_n)_{YY} \odot \Omega_{YY}(z)$, we get

$$\begin{aligned}(\nabla_z \mathscr{F}_1(z))_{lr} &= \frac{2\beta}{q\sigma^2}\sum_{i,j\in Y}\mathcal{A}_{ij}(z)(z_{ir} - z_{jr})(\delta_{jl} - \delta_{il}) \\ &= \frac{4\beta}{q\sigma^2}\left(\sum_{j\in Y}\mathcal{A}_{lj}(z)z_{jr} - z_{lr}\sum_{j\in Y}\mathcal{A}_{lj}(z)\right) \quad \text{since } \mathcal{A}(z) \text{ is symmetric}\end{aligned}$$

and therefore

$$\nabla_z \mathscr{F}_1(z) = \frac{4\beta}{q\sigma^2}\left(\mathcal{A}(z)z - (\mathcal{A}(z)\mathbf{1}_N) \odot z\right). \tag{6.11}$$

Computing $\nabla_z \mathscr{F}_2(z)$

By a similar argument as the above, for all $l \in Y$ and $r \in \{1, \ldots, q\}$

$$(\nabla_z \mathscr{F}_2(z))_{lr} = -\frac{4\beta}{q\sigma^2}\sum_{i\in Y, j\in Z}((G_n)_{ij})(\Omega_{YZ})_{ij}(z, z_d)(z_{ir} - (z_d)_{jr})\delta_{il}.$$

Hence, letting $\mathcal{B}(z) := (G_n)_{YZ} \odot \Omega_{YZ}(z, z_d)$, we get

$$\begin{aligned}(\nabla_z \mathscr{F}_2(z))_{lr} &= -\frac{4\beta}{q\sigma^2} \sum_{i\in Y, j\in Z} \mathcal{B}_{ij}(z)(z_{ir} - (z_d)_{jr})\delta_{il} \\ &= -\frac{4\beta}{q\sigma^2}\left(z_{lr} \sum_{j\in Z} \mathcal{B}_{lj}(z) - \sum_{j\in Z} \mathcal{B}_{lj}(z_d)_{jr} \right)\end{aligned}$$

and we therefore arrive at a similar formula to before

$$\nabla_z \mathscr{F}_2(z) = \frac{4\beta}{q\sigma^2}\left(\mathcal{B}(z) z_d - (\mathcal{B}(z)\mathbf{1}_{N_d}) \odot z\right). \tag{6.12}$$

Computing $\nabla_x \mathscr{F}_3(x)$

Recall that

$$\mathscr{F}_3(x) := \alpha ||\mathcal{T}(x) - y||_F^2 + \eta_n ||x - x_n||_F^2$$

The gradient of the latter term is simply

$$2\eta_n(x - x_n).$$

For the former term, since $\mathcal{T}$ is assumed to be $\mathcal{C}^1$, for all x there is a linear map D_x such that

$$\mathcal{T}(x + h) = \mathcal{T}(x) + D_x(h) + o(h).$$

Therefore

$$\begin{aligned}||\mathcal{T}(x + h) - y||_F^2 &= ||\mathcal{T}(x) + D_x(h) - y||_F^2 + o(h) \\ &= ||\mathcal{T}(x) - y||_F^2 + 2\langle D_x(h), \mathcal{T}(x) - y\rangle_F + o(h)\end{aligned}$$

and so the gradient of the former term is $2\alpha D_x^*(\mathcal{T}(x) - y))$. Hence

$$\nabla_x \mathscr{F}_3(x) = 2\alpha D_x^*(\mathcal{T}(x) - y) + 2\eta_n(x - x_n). \tag{6.13}$$

Note that if $\mathcal{T}$ is linear then $D_x = \mathcal{T}$ for all x.

The full gradient

Tying this all together, recalling $z := \mathcal{F}(x)$, we get

$$\begin{aligned}\nabla_x \mathscr{F}(x) &= \frac{4\beta}{q\sigma^2}\mathcal{F}^*\left(\mathcal{A}(z)z + \mathcal{B}(z)z_d - (\mathcal{A}(z)\mathbf{1}_N + \mathcal{B}(z)\mathbf{1}_{N_d}) \odot z\right) \\ &\quad + 2\alpha D_x^*(\mathcal{T}x - y) + 2\eta_n(x - x_n) \\ &= \frac{4\beta}{q\sigma^2}\mathcal{F}^*\left(\mathcal{C}(z)\begin{pmatrix} z \\ z_d \end{pmatrix} - (\mathcal{C}(z)\mathbf{1}_{N+N_d}) \odot z\right) \\ &\quad + 2\alpha D_x^*(\mathcal{T}(x) - y) + 2\eta_n(x - x_n)\end{aligned} \tag{6.14}$$

where $\mathcal{C}(z) := (G_n)_{YV} \odot \Omega_{ZV}(z, z_d)$. To compute (6.14), we need to compute matrix-vector products of the form $\mathcal{C}(z)v$.

Recalling (6.10), it follows that

$$(G_n)_{ZV} = -(u_n)_Y u_n^T + (v_n)_Y \mathbf{1}_V^T + \mathbf{1}_Y v_n^T.$$

Then we observe an neat linear algebra result[5]

$$((-u_n|_Y u_n^T + v_n|_Y \mathbf{1}_V^T + \mathbf{1}_Y v_n^T) \odot A)v = -u_n|_Y \odot (A(u_n \odot v)) + v_n|_Y \odot (Av) + A(v_n \odot v)$$

where in this case $A = \Omega_{YV}(z, z_d)$. Hence it suffices to be able to compute terms of the form $\Omega_{YV}(z, z_d)v$. But via the Nyström extension (5.1) we have

$$\Omega_{YV}(\mathcal{F}(x), z_d)v \approx \left(\Omega_{VX}(\mathcal{F}(x), z_d)\Omega_{XX}^{-1}(\mathcal{F}(x), z_d)\Omega_{XV}(\mathcal{F}(x), z_d)v\right)|_Y \tag{6.15}$$

where $X \subseteq V$ is some interpolation set, so we can compute matrix-vector products quickly.

6.3.2. Computing the objective function

The hard part is computing $\mathscr{F}_1$ and $\mathscr{F}_2$. This is equivalent to computing the full Frobenuis inner product

$$\langle G_n, \Omega(\mathcal{F}(x), z_d)\rangle_F.$$

Recall that

$$G_n = -u_n u_n^T + v_n \mathbf{1}^T + \mathbf{1} v_n^T.$$

Thus

$$\begin{aligned}\langle G_n, \Omega(\mathcal{F}(x), z_d)\rangle_F &= -\langle u_n, \Omega(\mathcal{F}(x), z_d)u_n\rangle_F + \langle v_n, \Omega(\mathcal{F}(x), z_d)\mathbf{1}\rangle_F + \langle \mathbf{1}, \Omega(\mathcal{F}(x), z_d)v_n\rangle_F \\ &= -\langle u_n, \Omega(\mathcal{F}(x), z_d)u_n\rangle_F + 2\langle v_n, \Omega(\mathcal{F}(x), z_d)\mathbf{1}\rangle_F\end{aligned}$$

and as in (6.15) we can use the Nyström extension to compute approximations to these matrix-vector products quickly.

6.4. The full algorithm for (6.9)

In this section, we combine the above ingredients together into a sequence of algorithms. We begin with Algorithm 3, implementing the iPiano update used to solve (6.9a). We finally give the algorithm for (6.9) in Algorithm 5.

6.5. Linearising (6.9a)

In the test we will shortly describe, we found that (6.9a) was a computational bottleneck. We will therefore also consider a simplification of (6.9a). Recall from section 6.3 that the objective functional of (6.9a) can be rewritten

$$R(x) + \mathscr{F}_1(\mathcal{F}(x)) + \mathscr{F}_2(\mathcal{F}(x)) + \alpha||\mathcal{T}(x) - y||_F^2 + \eta_n||x - x_n||_F^2.$$

[5] To see this, observe that in suffix notation, for $i \in Y$ and $j \in V$, the LHS is $(-(u_n)_i(u_n)_j + (v_n)_i + (v_n)_j)A_{ij}v_j$ and the RHS is $-(u_n)_i A_{ij}((u_n)_j v_j) + (v_n)_i A_{ij} v_j + A_{ij}((v_n)_j v_j)$.

Algorithm 3 Implementation of a step of iPiano for (6.9a).

function iPiano($x', x, z_d, u, v, \mathcal{F}, \mathcal{T}, D^*, y, R, \alpha, \beta, q, \sigma, \eta, \delta x', L, \xi, a, b, V, Y, Z, K$) // Computes a step of the iPiano scheme for (6.9a), outputting the iPiano update x'_{k+1} of x'_k for x'_k equal to the input x', as well as the value of L such that (6.7) is satisfied.
  $\mathscr{F}_{old} = \texttt{Feval}(x', x, z_d, u, v, \mathcal{F}, \mathcal{T}, y, \alpha, \beta, \sigma, \eta, V, Y, Z, K)$ // `Feval` defined in Algorithm 4
  $z' = \mathcal{F}(x')$
  $w_1 = \texttt{CzProd}(z', (z, z_d), u, v, \sigma, V, Y, Z, K)$ // `CzProd` defined in Algorithm 4
  $w_2 = \texttt{CzProd}(z', \mathbf{1}_V, u, v, \sigma, V, Y, Z, K)$
  $g_1 = \frac{4\beta}{q\sigma^2}\mathcal{F}^*(w_1 - w_2 \odot z')$
  $g_2 = 2\alpha D^*_{x'}(\mathcal{T}(x') - y) + 2\eta_n(x' - x)$
  $g = g_1 + g_2$ // $g \approx \nabla_x \mathscr{F}(x')$ as in (6.14)
  check $= 0$ // Check variable for condition (6.7)
  while check $= 0$ and $L < 10^{10}$ **do** // Backtracking loop
    $\theta = 1.99(1 - \xi)/L$ // As an example of how to pick θ
    $p = \text{prox}_{\theta R}(x' - \theta g + \xi \delta x')$ // prox computed using FISTA [6]
    $\mathscr{F}_{new} = \texttt{Feval}(p, x, z_d, u, v, \mathcal{F}, \mathcal{T}, y, \alpha, \beta, \sigma, \eta, V, Y, Z, K)$
    if $\mathscr{F}_{new} \leq \mathscr{F}_{old} + \langle g, p - x' \rangle_F + \frac{1}{2}L||p - x'||_F^2$ **then**
      check $= 1$
    else
      $L = aL$
    end if
  end while
  return p, L
end function

Algorithm 4 Definitions of the `Feval` and `CzProd` functions used in Algorithm 3.

function Feval($x, x_n, z_d, u, v, \mathcal{F}, \mathcal{T}, y, \alpha, \beta, \sigma, \eta, V, Y, Z, K$) // Approximates $\mathscr{F}(x)$ as in section 6.3.2.
  $z = \mathcal{F}(x)$
  $F_1 = -\langle u, \texttt{OmegaProd}(u, z, z_d, \sigma, V, Y, Z, K)\rangle_F$ // `OmegaProd` defined below
  $F_2 = 2\langle v, \texttt{OmegaProd}(\mathbf{1}_V, z, z_d, \sigma, V, Y, Z, K)\rangle_F$
  $F_3 = \alpha||\mathcal{T}(x) - y||_F^2 + \eta||x - x_n||_F^2$
  return $F_1 + F_2 + F_3$
end function
function CzProd($z, v, u, v_n, \sigma, V, Y, Z, K$) // Approximates $\mathcal{C}(z)v$ as in the end of section 6.3.1.
  $A : w \mapsto (\texttt{OmegaProd}(w, z, z_d, \sigma, V, Y, Z, K))|_Y$
  $w_1 = -u|_Y \odot A(u \odot v)$
  $w_2 = v_n|_Y \odot A(v)$
  $w_3 = A(v_n \odot v)$
  return $w_1 + w_2 + w_3$
end function
function OmegaProd($v, z, z_d, \sigma, V, Y, Z, K$) // Approximates $\Omega(z, z_d)v$ via the Nyström extension as in (6.15).
  $\omega : ij \mapsto \Omega_{ij}(z, z_d, \sigma)$ // Defined as in (6.3)
  $X_1 = \texttt{random\_subset}(Y, K/2)$ // A random subset of Y of size $K/2$
  $X_2 = \texttt{random\_subset}(Z, K/2)$ // A random subset of Z of size $K/2$
  $X = X_1 \cup X_2$
  $\omega_{XX} = \omega(X, X)$
  $\omega_{VX} = \omega(V, X)$
  $\omega_{XX}v' = \omega_{VX}^T v$ // Solving the linear system for v'
  return $\omega_{VX}v'$
end function

Algorithm 5 Graph-based joint reconstruction-segmentation algorithm implementing (6.9).

function JointReconSeg($y, \mathcal{T}, D^*, z_d, \tilde{f}, V, Y, Z, \mathcal{F}, \sigma, R, W, \alpha, \beta, \eta_n, \nu_n, \tau, \varepsilon, \mu, a, b, L_0, \xi, K, N$) // Computes the first N iterates of (6.9).
  $x_0 = \texttt{cheap\_reconstruction}(y, \mathcal{T})$ // Initial cheap reconstruction
  $u_0 = \texttt{SDIE\_seg}(\mathcal{F}(x_0), z_d, \mu, \tilde{f}, \tau, \varepsilon, \sigma)$ // Initial SDIE segmentation (see chapter 5) on the graph built from the features $(\mathcal{F}(x_0), z_d)$
  for $n = 0$ to $N - 1$ **do** // The iterations of (6.9)
    $x_0' = x_n$
    $v_n = \frac{1}{2}(u_n)^2 + \frac{1}{2\varepsilon}W(u_n) + \frac{1}{4}\mu \odot (u_n - \tilde{f})^2$ // Squaring elementwise
    $k = 0$
    while iPiano stopping condition not met **do**
      $\delta x' = 0$
      if $k > 0$ **then**
        $L_k = bL_{k-1}$
        $\delta x' = x_k' - x_{k-1}'$
      end if
      $[x_{k+1}', L_k] = \texttt{iPiano}(x_k', x_n, z_d, u_n, v_n, \mathcal{F}, \mathcal{T}, D^*, y, R, \alpha, \beta, \sigma, \eta_n, \delta x', L_k \xi, a, b, V, Y, K)$
      $k = k + 1$
    end while
    $x_{n+1} = x_k'$ // Solving (6.9a)
    $\mu' = \mu + 2\nu_n\beta^{-1}\chi_Y$
    $\tilde{f}' = \tilde{f} + u_n \odot \chi_Y$
    $u_{n+1} = \texttt{SDIE\_seg}(\mathcal{F}(x_{n+1}), z_d, \mu', \tilde{f}', \tau, \varepsilon, \sigma)$ // Solving (6.9b)
  end for
  return $\{x_n\}_{n=0}^N, \{u_n\}_{n=0}^N$
end function

Let us assume that our candidate minimisers are close to x_n (which will become a more accurate assumption the larger η_n is). Then we shall simplify this functional via the linearisation

$$\begin{aligned}\mathscr{F}_1(z) + \mathscr{F}_2(z) &\approx \mathscr{F}_1(z_n) + \mathscr{F}_2(z_n) + \langle x - x_n, \mathcal{F}^*\left(\nabla_z \mathscr{F}_1(z_n) + \nabla_z \mathscr{F}_2(z_n)\right)\rangle \\ &\simeq \langle x, g_n\rangle\end{aligned}$$

where $z := \mathcal{F}(x)$, $z_n := \mathcal{F}(x_n)$, and $g_n := \mathcal{F}^*\left(\nabla_z \mathscr{F}_1(z_n) + \nabla_z \mathscr{F}_2(z_n)\right)$. Given this assumption it follows that the linearisation of the objective function of (6.9a) is

$$R(x) + \langle x, g_n\rangle_F + \alpha||\mathcal{T}(x) - y||_F^2 + \eta_n||x - x_n||_F^2$$

which is equivalent to

$$R(x) + \alpha||\mathcal{T}(x) - y||_F^2 + \eta_n||x - \tilde{x}_n||_F^2 \tag{6.16}$$

where $\tilde{x}_n := x_n - \frac{1}{2}\eta_n^{-1} g_n$.

Note 37. *If we define $R_n(x) := R(x) + \eta_n||x - \tilde{x}_n||_F^2$ then minimising (6.16) is the same as solving*

$$\operatorname*{argmin}_x R_n(x) + \alpha||\mathcal{T}(x) - y||_F^2$$

which is of the form of a standard variational image reconstruction problem (6.2), which we discussed in section 6.1.1. In this thesis we will continue to solve this problem via iPiano, but given its standard form we seek to improve on this in future work by using the standard methods from the literature (a subset of which we listed in section 6.1.1).

Thus, our difficulty is reduced to computing $\tilde{x}_n$, which requires computing g_n. Defining $\mathcal{C}_n := (G_n)_{YZ} \odot \Omega_{YZ}(z_n, z_d)$, by considering (6.14) we get that

$$g_n = \frac{4\beta}{q\sigma^2}\mathcal{F}^*\left(\mathcal{C}_n\begin{pmatrix} z_n \\ z_d\end{pmatrix} - (\mathcal{C}_n \mathbf{1}_{N+N_d}) \odot z_n\right)$$

which we can compute via the methods described at the end of section 6.3.1.

We describe the joint reconstruction-segmentation scheme using this linearised (6.9a) in Algorithm 6.

6.6. A simple denoising-segmentation test

To demonstrate this scheme in action, we will apply it to a noised version of the "two cows" example (i.e. Example 5.3.1) from the previous chapter. We will exhibit two parameter set-ups, which demonstrate different behaviour.

6.6.1. The example

Example 6.6.1 (Noised two cows)**.** *We take the reference data Z and reference labels $\tilde{f}$ as in Example 5.3.1. The true image x^* to be reconstructed and segmented*

Algorithm 6 Graph-based joint reconstruction-segmentation algorithm implementing (6.9) using the linearised (6.9a).

function JointReconSeg2($y, \mathcal{T}, D^*, z_d, \tilde{f}, V, Y, Z, \mathcal{F}, \sigma, R, W, \alpha, \beta, \eta_n, \nu_n, \tau, \varepsilon, \mu, a, b, L_0, \xi, K, N$) // Computes the first N iterates of (6.9) using the linearised (6.9a).
  $x_0 = \texttt{cheap\_reconstruction}(y, \mathcal{T})$ // Initial cheap reconstruction
  $u_0 = \texttt{SDIE\_seg}(\mathcal{F}(x_0), z_d, \mu, \tilde{f}, \tau, \varepsilon, \sigma)$ // Initial SDIE segmentation
  for $n = 0$ to $N - 1$ **do** // The iterations of (6.9)
    $v_n = \frac{1}{2}(u_n)^2 + \frac{1}{2\varepsilon}W(u_n) + \frac{1}{4}\mu \odot (u_n - \tilde{f})^2$ // Squaring elementwise
    $z_n = \mathcal{F}(x_n)$
    $w_1 = \texttt{CzProd}(z_n, (z_n, z_d), u_n, v_n, \sigma, V, Y, Z, K)$ // See Algorithm 4
    $w_2 = \texttt{CzProd}(z_n, \mathbf{1}_V, u_n, v_n, \sigma, V, Y, Z, K)$
    $g_n = \frac{4\beta}{q\sigma^2}\mathcal{F}^*(w_1 - w_2 \odot z_n)$
    $\tilde{x}_n = x_n - \frac{1}{2}\eta_n^{-1} g_n$
    $x'_0 = x_n$
    $k = 0$
    while iPiano stopping condition not met **do**
      $\delta x'_k = 0$
      if $k > 0$ **then**
        $L_k = bL_{k-1}$
        $\delta x'_k = x'_k - x'_{k-1}$
      end if
      $\mathscr{F}_{old} = \alpha||\mathcal{T}(x'_k) - y||_F^2 + \eta_n||x'_k - \tilde{x}_n||_F^2$
      $\nabla\mathscr{F} = 2\alpha D^*_{x'_k}(\mathcal{T}(x'_k) - y) + 2\eta_n(x'_k - \tilde{x}_n)$
      check $= 0$ // Check variable for condition (6.7)
      while check $= 0$ and $L_k < 10^{10}$ **do** // Backtracking loop
        $\theta = 1.99(1 - \xi)/L_k$
        $x'_{k+1} = \text{prox}_{\theta R}(x'_k - \theta\nabla\mathscr{F} + \xi\delta x'_k)$ // Computed using FISTA [6]
        $\mathscr{F}_{new} = \alpha||\mathcal{T}(x'_{k+1}) - y||_F^2 + \eta_n||x'_{k+1} - \tilde{x}_n||_F^2$
        if $\mathscr{F}_{new} \leq \mathscr{F}_{old} + \langle\nabla\mathscr{F}, x'_{k+1} - x'_k\rangle_F + \frac{1}{2}L_k||x'_{k+1} - x'_k||_F^2$ **then**
          check $= 1$
        else
          $L_k = aL_k$
        end if
      end while
      $k = k + 1$
    end while
    $x_{n+1} = x'_k$ // Solving the linearised (6.9a)
    $\mu' = \mu + 2\nu_n\beta^{-1}\chi_Y$
    $\tilde{f}' = \tilde{f} + u_n \odot \chi_Y$
    $u_{n+1} = \texttt{SDIE\_seg}(\mathcal{F}(x_{n+1}), z_d, \mu', \tilde{f}', \tau, \varepsilon, \sigma)$ // Solving (6.9b)
  end for
  return $\{x_n\}_{n=0}^N, \{u_n\}_{n=0}^N$
end function

will be also be as in that example. Finally, we take the observed data y (see Fig. 6.1) to be x^* plus Gaussian noise with mean zero and standard deviation 0.6, created using `imnoise`. We generate y once and use the same y throughout this section. Since this example is a denoising with no further transformations, $\mathcal{T}$ is the identity.

Figure 6.1: The observed data y for Example 6.6.1.

Note that the noise level in this example corresponds to a significant amount of noise, as the mean value of x^* is 0.5116. This noise rate was chosen to be a stress test of the method.

6.6.2. Set-up

Parameters

We took $\alpha = 50$, $\eta_n = 2\beta \times 2^n$, and $\sigma = 3$. For the SDIE scheme for (6.9b), we took $\tau = \varepsilon = 0.003$, $\mu = 50\chi_Z$, and stopping condition parameter $\delta = 10^{-10}$. For the iPiano scheme for (6.9a), we took $\xi = 0.45$, $L_0 = 1000$, $a = 2$, $b = 1$, and $\alpha_k = 0.099/L_k$. For all matrix compressions, we took $K = 70$.

Next, we took as regulariser R the Huber-TV [15] function:

$$R(x) = 10 \sum_{i \in Y} \begin{cases} ||(\nabla x)_i||_2 - 0.005, & \text{if } ||(\nabla x)_i||_2 > 0.01, \\ ||(\nabla x)_i||_2^2 / 0.02, & \text{if } ||(\nabla x)_i||_2 \leq 0.01, \end{cases}$$

where ∇x is the vector of forward finite differences between x at the pixel i and at its neighbouring pixels, i.e. for each j either directly above/below or directly left/right of i, $(\nabla x)_i$ has a component with the value $x_j - x_i$. This choice of regulariser was fairly arbitrary; we have not yet explored the impact of the choice of regulariser on the behaviour of this scheme.

Finally, we considered two cases for β and ν_n:

(I) $\beta = 0.05$ and $\nu_n = 0.25 \times (1.3)^n$.

(II) $\beta = 0.0001$ and $\nu_n = 20 \times (1.3)^n$.

Initialisation

The initial reconstruction x_0 was computed via a standard TV-based (i.e., Rudin–Osher–Fatemi [22]) denoising, with fidelity term 0.1. That is

$$x_0 = \operatorname*{argmin}_x \mathrm{TV}(x) + 0.1||x - y||_F^2$$

where $\mathrm{TV}(x) := \sum_{i \in Y} ||(\nabla x)_i||_2$ for ∇x defined as above.

The initial segmentation u_0 of x_0 was computed via the SDIE scheme with the above parameters and with initial state $0.48\chi_Y + \tilde{f}$.

The feature map and its adjoint

In the above, we needed to be able to quickly compute $\mathcal{F}$ and $\mathcal{F}^*$. This constrains somewhat our choice of features.

As a simple choice we define $\mathcal{F}$ as follows. For each pixel $i \in Y$, suppose we have a map $\mathcal{N}_i : \{1, ..., k\} \to Y$ which defines the k "neighbours" of i in Y (in the sense of the image, not the graph) and we likewise have a kernel $\mathcal{K} : \{1, ..., k\} \to \mathbb{R}$. Then for each channel $s \in \{1, ..., \ell\}$ of x, $i \in Y$, and $p \in \{1, ..., k\}$, we define $\mathcal{G}(x^s) := z^s \in \mathbb{R}^{N \times k}$, where z^s is given by

$$z^s_{ip} := \mathcal{K}(p) x^s_{\mathcal{N}_i(p)}.$$

Then $z = \mathcal{F}(x) \in \mathbb{R}^{N \times k\ell}$ is defined by $z = \begin{pmatrix} z^1 & z^2 & \dots & z^\ell \end{pmatrix}$. We can then derive the adjoint of $\mathcal{F}$ with respect to $\langle \cdot, \cdot \rangle_F$, i.e. the map $\mathcal{F}^* : \mathbb{R}^{N \times q} \to \mathbb{R}^{N \times \ell}$ such that for all $x \in \mathbb{R}^{N \times \ell}$ and $w \in \mathbb{R}^{N \times q}$

$$\langle \mathcal{F}(x), w \rangle_F = \langle x, \mathcal{F}^*(w) \rangle_F.$$

Write $w = \begin{pmatrix} w^1 & w^2 & \dots & w^\ell \end{pmatrix}$, where $w^s \in \mathbb{R}^{N \times k}$. Then since

$$\begin{aligned} \langle \mathcal{G}(x^s), w^s \rangle_F &= \sum_{i \in Y, p=1}^{k} x^s_{\mathcal{N}_i(p)} \mathcal{K}(p) w^s_{ip} \\ &= \sum_{j \in Y} x^s_j \left(\sum_{\{(i,p) | \mathcal{N}_i(p) = j\}} \mathcal{K}(p) w^s_{ip} \right) \end{aligned}$$

it follows that $\mathcal{G}$ has adjoint $\mathcal{G}^* : \mathbb{R}^{N \times k} \to \mathbb{R}^N$ given by

$$(\mathcal{G}^*(w^s))_j = \sum_{\{(i,p) | \mathcal{N}_i(p) = j\}} \mathcal{K}(p) w^s_{ip}.$$

Furthermore, by construction $\mathcal{F}^*(w) = \begin{pmatrix} \mathcal{G}^*(w^1) & \mathcal{G}^*(w^2) & \dots & \mathcal{G}^*(w^\ell) \end{pmatrix}$.

In particular, we took $k = 9$, with the neighbours of pixel i corresponding to the 3×3 square centred on i, and $\mathcal{K}$ being 9 multiplied by a 3×3 Gaussian kernel with standard deviation 1, centred on the centre of that square. Since the image is RGB and hence $\ell = 3$, it follows that $q = 27$.

6.6.3. Results for parameter set-up (I)

We first examine the behaviour of the scheme for the parameter set-up (I). The bottle-neck in terms of run time was found to be in solving (6.9a). We exhibit two strategies to alleviate this: truncating the number of iterations of the iPiano scheme, and using the linearised scheme from section 6.5. We give here results for three cases: truncation to a single step, truncation to five steps, and the linearised case.

All run times are for MATLAB R2019a implementations performed on a ASUS ZenBook with Intel® Core™ i5-8265U CPU @ 1.60GHz and 8.00 GB RAM.

Single-step iPiano

Figure 6.2: Results after 30 iterations of (6.9) using the full (6.9a) with iPiano restricted to a single step, and using parameter set-up (I).

We give results for iPiano truncated to a single step in Fig. 6.2. In that figure, the top left is x_0, i.e. the TV-denoised y. Bottom left is the ground truth masked with u_0, i.e. the SDIE segmentation of x_0. The middle column are the reconstruction at iteration 30 and the ground truth masked with the segmentation at iteration 30. The top-right shows the most accurate reconstruction, which occurred at iteration 7 and had relative error 0.1805 to the ground truth. Finally, the bottom right shows the ground truth masked with the most accurate segmentation, which occurred at iteration 25 and was 94.7998% accurate (it should be noted that the segmentation at iteration 8 was already 94.7161% accurate).

It took around 20 minutes to compute all 30 iterations.

Five-step iPiano

We give results for iPiano truncated to five steps in Fig. 6.3. In that figure, the top left is x_0, i.e. the TV-denoised y. Bottom left is the ground truth masked with

Figure 6.3: Results after 30 iterations of (6.9) using the full (6.9a) with iPiano restricted to five steps, and using parameter set-up (I).

u_0, i.e. the SDIE segmentation of x_0. The middle column are the reconstruction at iteration 30 and the ground truth masked with the segmentation at iteration 30. The top-right shows the most accurate reconstruction, which occurred at iteration 4 and had relative error 0.1867 to the ground truth. Finally, the bottom right shows the ground truth masked with the most accurate segmentation, which occurred at iteration 20 and was 93.6670% accurate.

It took around 40 minutes to compute all 30 iterations.

Linearised (6.9a)

After the initial segmentation, Fig. 6.4 shows $\tilde{x}_0$ (see (6.16) and the line following) and the corresponding x_1. As can be observed, this parameter set-up has induced an extreme segmentation-driven contrast. The scheme then breaks down, as when we try to use the Nyström method to compute the next segmentation, we find that the submatrix ω_{XX} has infinite condition number.

Discussion

We notice a number of important features of these results. First and foremost, it seems that the reconstruction and segmentation do not converge to the ground truths with this parameter set-up. As was previously discussed, for the reconstruction this is not unexpected and is not necessarily undesired. That is, the increasing inaccuracy of later reconstructions derives largely from the increasing levels of segmentation-driven contrast. As can be seen by the relative iterations at which the best segmentation is attained vs. the best reconstruction, this increase in contrast does for a time lead to better segmentations. However, at a certain point this stops being the case, and the accuracy of the segmentation drops as it settles on a

(a) The image $\tilde{x}_0$, i.e. segmentation-driven adjustment of the initial reconstruction.

(b) The first reconstruction x_1, i.e. the minimiser of (6.16) for $n = 0$.

Figure 6.4: After the initial segmentation, the segmentation-driven adjustment $\tilde{x}_0$ of x_0 and the corresponding first reconstruction x_1, using the linearised (6.9a) from section 6.5, and using parameter set-up (I).

constant segmentation. One possible explanation for this is that it is because of the fact that, due to the increase in contrast, the edge weights both between vertices in Y and Y and between vertices in Y and Z are on average lower in the graph generated by x_{30} compared to the graph generated by x_0.[6] This will cause there to be less penalisation of the $u = (\mathbf{0}_Y, \tilde{f})$ candidate, potentially explaining why we see something close to it eventually dominate.

Secondly, and perhaps more worryingly, increased iterations of iPiano seem to result in less accurate reconstructions and segmentations, and a greatly increased run time. Inspecting the values of the objective functional in (6.9a) during the iPiano iterations, we discovered that it frequently increased. This suggests that the problem is that our approximations of the gradient in section 6.3.1 are insufficiently accurate, and these errors in the gradient are compounding. We tested this hypothesis by scaling down the image to the point that we could compute the exact gradient, and found that in this scaled-down case the objective functional in (6.9a) monotonically decreased during the iPiano iterations.

Finally, we observed that in the linearised case if $\tilde{x}$ has too much segmentation-driven contrast, then the segmentation method breaks down as the weight matrix becomes singular. Therefore we must not have β/η_n too large in the linearised scheme.

6.6.4. Results for parameter set-up (II)

We now examine the behaviour of the scheme for the parameter set-up (II). We found for this set-up that the iPiano scheme could be run to convergence. We give here results for two cases: solving (6.9) with the full (6.9a), and solving (6.9) with the linearised (6.9a).

[6] For example, in the single-step case the average edge weight between the first 1000 vertices in Y and themselves fell from 0.9970 to 0.9789, and the average weight between the first 1000 vertices in Y and the first 1000 vertices of Z fell from 0.9950 to 0.9841.

The full scheme

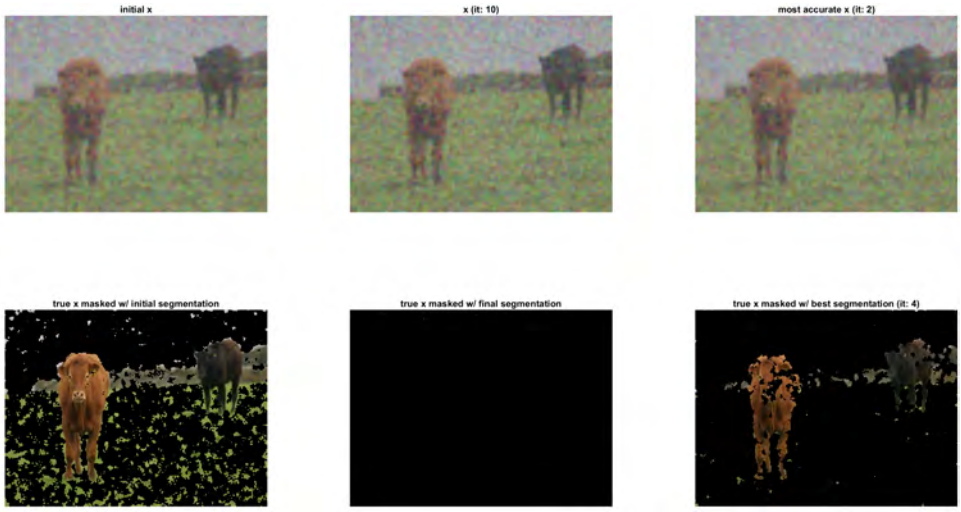


Figure 6.5: Results after 10 iterations of (6.9) using the full (6.9a), and using parameter set-up (II).

We show the results of using the full scheme in Fig. 6.5. In that figure, the top left is x_0, i.e. the TV-denoised y. Bottom left is the ground truth masked with u_0, i.e. the SDIE segmentation of x_0. The middle column are the reconstruction at iteration 10 and the ground truth masked with the segmentation at iteration 10. The top-right shows the most accurate reconstruction, which occurred at iteration 2 and had relative error 0.2145 to the ground truth. Finally, the bottom right shows the ground truth masked with the most accurate segmentation, which occurred at iteration 4 and was 94.8597% accurate.

It took around 30 minutes to compute all 10 iterations.

The linearised scheme

We show the results of using the linearised scheme in Fig. 6.6. In that figure, the top left is x_0, i.e. the TV-denoised y. Bottom left is the ground truth masked with u_0, i.e. the SDIE segmentation of x_0. The middle column are the reconstruction at iteration 10 and the ground truth masked with the segmentation at iteration 10. The top-right shows the most accurate reconstruction, which occurred at iteration 3 and had relative error 0.2014 to the ground truth. Finally, the bottom right shows the ground truth masked with the most accurate segmentation, which occurred at iteration 5 and was 94.8538% accurate.

It took around 10 minutes to compute all 10 iterations.

Discussion

Much of what we observe here is similar to before, however there are some notable differences. First with this parameter set-up, the linearised scheme works and gives results which are of equal quality to the full scheme. Second, the iPiano scheme

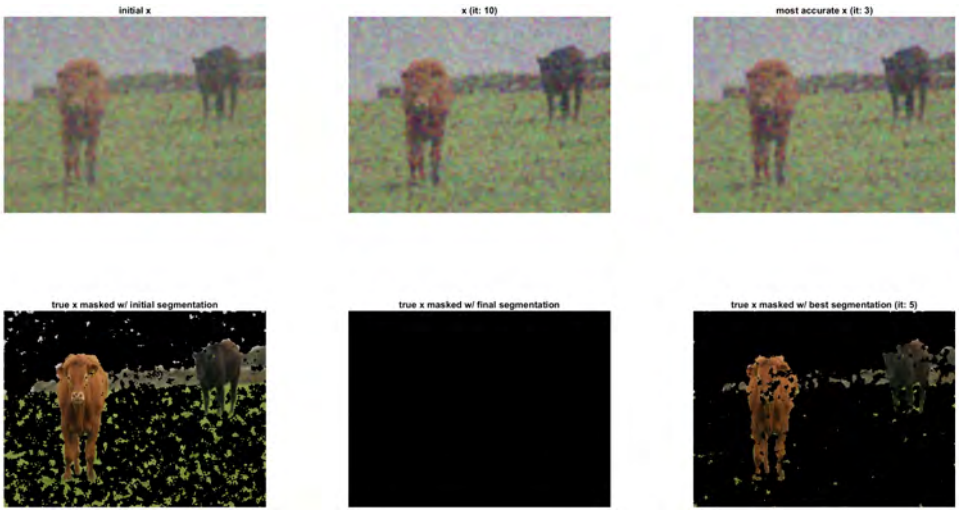


Figure 6.6: Results after 10 iterations of (6.9) using the linearised (6.9a), and using parameter set-up (II).

ran significantly faster with this parameter set-up, such that we could run it to convergence, which took around 20 iPiano iterations for the full scheme (recall that for parameter set-up (I) we truncated to at most 5 iPiano iterations) and less than 10 iPiano iterations for the linearised scheme. Third, we found that the iPiano scheme roughly monotonically decreased the energy for the full scheme, and for the linearised scheme always monotonically decreased the energy, reinforcing our above theory that the issues with iPiano in set-up (I) were caused by errors in the computation of the gradient. Finally, this time the reconstruction converges without going overboard on the contrast, though the converged reconstruction is still worse than the best reconstruction, and the best reconstructions are worse than for set-up (I) (though the best segmentations are better).

6.7. Conclusions and directions for future work

In this chapter, we have outlined a framework for joint reconstruction-segmentation on graphs, understood as an iterative scheme. As in our earlier segmentation results, complications arise because of the large size of the matrices involved, requiring some careful analysis to reduce the problem to that of computing a number of matrix-vector products which can be done via the Nyström method. We presented an algorithm for this scheme in Algorithm 5. We furthermore considered a simplified scheme, where by linearising the segmentation-driven regularisation in (6.9a) we reduce that problem to a standard image reconstruction problem. We presented an algorithm for this simplified scheme in Algorithm 6. We then performed some basic numerical experiments with a highly noised version of the "two cows" example from the previous chapter.

However, this work is still in early days, based on as-yet unpublished material,

and there are many directions for future work.

First, as we saw above the scheme is highly influenced by parameter choices, and there are a large number of parameters that require tuning. It will therefore be important to investigate how to tune these parameters in a straightforward and reliable way, in order to get high-quality results. Moreover, we have made a number of choices above regarding our regulariser, our initial reconstruction method, and the use of iPiano to solve problems like (6.9a), and we seek to investigate the impact of those choices and identify any potential improvements.

Second, we also saw that in neither of the parameter cases we looked at did the scheme converge to the best reconstruction-segmentation. Rather, the segmentation converged to something close to $(\mathbf{0}_Y, \tilde{f})$, and the reconstruction would improve for a few iterations before declining in quality, in the former case significantly as the segmentation-driven contrast took over. This might be solvable by better parameter choices, or by the parameters adapting appropriately over the course of the scheme. Alternatively, we might have to develop some smart stopping condition that can identify when the scheme has done the best it can.

Third, it would seem that the gradient of the energy in (6.9a) is not being approximated sufficiently accurately by the Nyström method described in section 6.3.1. Future work should therefore investigate ways to improve this approximation. One avenue might be to consider the NFFT method of Alfke *et al.* [2] as an alternative method for computing the required matrix-vector products. Another avenue will be to consider how we can reduce the dimension of our problem, for example by reconstructing the image as a collection of sub-images, which we could also connect to only a subset of our reference data, and thereby form a smaller graph. However such an approach of splitting up our dataset might cause us to lose accuracy, and might also lead to the sub-images being inconsistently reconstructed or segmented resulting in a very patchwork final reconstruction-segmentation.

Fourth, one modification that could improve the quality of the scheme greatly is to incorporate the "human-in-the-loop" idea from Qiao *et al.* [20]. That is, in our experiments above the first few iterations did a great job clearing away a lot of the noise in the initial segmentation, but then got stuck with a segmentation accuracy in the low-to-mid 90s%, not able to identify the misclassified grass, wall, and parts of the cow's face. This in turn in set-up (I) led to artefacts from that misclassification getting included in the reconstruction. A human-in-the-loop could identify this, pause the scheme, spend a few minutes in an image editor tidying things up (erasing the wall, most of the grass, and filling in the face) before handing the tidied up segmentation back to the scheme to continue working with.

Finally, on the theoretical side we want to investigate the convergence properties of this scheme, making use of the theory from Attouch *et al.* [4]. In particular, we want to understand the conditions under which the iterative scheme (6.9) converges to a minimiser of (6.8).

Bibliography

[1] Jonas Adler et al. *Task adapted reconstruction for inverse problems*. 2018. arXiv: 1809.00948 [cs.CV].

[2] Dominik Alfke et al. "NFFT Meets Krylov Methods: Fast Matrix-Vector Products for the Graph Laplacian of Fully Connected Networks". In: *Frontiers in Applied Mathematics and Statistics* 4 (2018), p. 61. issn: 2297-4687. doi: 10.3389/fams.2018.00061. url: https://www.frontiersin.org/article/10.3389/fams.2018.00061.

[3] Simon Arridge et al. "Solving inverse problems using data-driven models". In: *Acta Numerica* 28 (2019), pp. 1–174. doi: 10.1017/S0962492919000059.

[4] Hédy Attouch et al. "Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality". In: *Mathematics of operations research* 35.2 (2010), pp. 438–457.

[5] Gilles Aubert and Pierre Kornprobst. *Mathematical problems in image processing: partial differential equations and the calculus of variations*. 2nd ed. Vol. 147. Applied Mathematical Sciences. New York: Springer-Verlag, 2006. isbn: 978-0-387-32200-1. doi: 10.1007/978-0-387-44588-5.

[6] Amir Beck and Marc Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". In: *SIAM journal on imaging sciences* 2.1 (2009), pp. 183–202.

[7] Amir Beck and Marc Teboulle. "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems". In: *IEEE transactions on image processing* 18.11 (2009), pp. 2419–2434.

[8] Antonin Chambolle. "An algorithm for total variation minimization and applications". In: *Journal of Mathematical imaging and vision* 20.1 (2004), pp. 89–97.

[9] T.F. Chan and L.A. Vese. "Active contours without edges". In: *IEEE Transactions on Image Processing* 10.2 (2001), pp. 266–277. doi: 10.1109/83.902291.

[10] Veronica Corona et al. "Enhancing joint reconstruction and segmentation with non-convex Bregman iteration". In: *Inverse Problems* 35.5 (2019), p. 055001. doi: 10.1088/1361-6420/ab0b77.

[11] Thomas A Gallagher, Alexander J Nemeth, and Lotfi Hacein-Bey. "An introduction to the Fourier transform: relationship to MRI". In: *American journal of roentgenology* 190.5 (2008), pp. 1396–1405.

[12] Tom Goldstein and Stanley Osher. "The split Bregman method for L1-regularized problems". In: *SIAM journal on imaging sciences* 2.2 (2009), pp. 323–343.

[13] Gene H. Golub and Charles F. van Loan. *Matrix Computations*. 4th ed. Baltimore: Johns Hopkins University Press, 2013. isbn: 9781421407944. url: http://www.cs.cornell.edu/cv/GVL4/golubandvanloan.htm.

[14] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. url: http://www.deeplearningbook.org.

[15] Peter J. Huber. *Robust Statistics*. Wiley Series in Probability and Statistics. New York: John Wiley & Sons, 1981. isbn: 9780471418054. doi: 10.1002/0471725250.

[16] Karl Kunisch Kristian Bredies and Thomas Pock. "Total Generalized Variation". In: *SIAM Journal on Imaging Sciences* 3.3 (2010), pp. 492–526. doi: 10.1137/090769521.

[17] David Mumford and Jayant Shah. "Optimal approximations by piecewise smooth functions and associated variational problems". In: *Communications on Pure and Applied Mathematics* 42.5 (1989), pp. 577–685. doi: 10.1002/cpa.3160420503. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.3160420503. url: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160420503.

[18] Peter Ochs et al. "iPiano: Inertial proximal algorithm for nonconvex optimization". In: *SIAM Journal on Imaging Sciences* 7.2 (2014), pp. 1388–1419.

[19] David L Phillips. "A technique for the numerical solution of certain integral equations of the first kind". In: *Journal of the ACM (JACM)* 9.1 (1962), pp. 84–97.

[20] Yiling Qiao et al. "Uncertainty quantification for semi-supervised multi-class classification in image processing and ego-motion analysis of body-worn videos". In: *Electronic Imaging* 2019.11 (2019), pp. 264-1-264–7. issn: 2470-1173. doi: doi:10.2352/ISSN.2470-1173.2019.11.IPAS-264. url: https://www.ingentaconnect.com/content/ist/ei/2019/00002019/00000011/art00015.

[21] Ronny Ramlau and Wolfgang Ring. "A Mumford–Shah level-set approach for the inversion and segmentation of X-ray tomography data". In: *Journal of Computational Physics* 221.2 (2007), pp. 539–557.

[22] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. "Nonlinear total variation based noise removal algorithms". In: *Physica D: Nonlinear Phenomena* 60.1 (1992), pp. 259–268. issn: 0167-2789. doi: 10.1016/0167-2789(92)90242-F. url: https://www.sciencedirect.com/science/article/pii/016727899290242F.

[23] David Strong and Tony Chan. "Edge-preserving and scale-dependent properties of total variation regularization". In: *Inverse problems* 19.6 (2003), S165.

[24] A. N. Tikhonov. "Solution of incorrectly formulated problems and the regularization method". In: *Soviet Math. Dokl.* 4 (1963), pp. 1035–1038.

[25] Singanallur V. Venkatakrishnan, Charles A. Bouman, and Brendt Wohlberg. "Plug-and-Play priors for model based reconstruction". In: *2013 IEEE Global Conference on Signal and Information Processing*. 2013, pp. 945–948. doi: 10.1109/GlobalSIP.2013.6737048.

[26] Yilun Wang et al. "A new alternating minimization algorithm for total variation image reconstruction". In: *SIAM Journal on Imaging Sciences* 1.3 (2008), pp. 248–272.

[27] Wangmeng Zuo and Zhouchen Lin. "A Generalized Accelerated Proximal Gradient Approach for Total-Variation-Based Image Restoration". In: *IEEE transactions on image processing* 20.10 (2011), pp. 2748–2759.

7

Mean Curvature Flow on Graphs

Mathematics is the art of giving the same name to different things.

Henri Poincaré, *The Future Of Mathematics*

In chapter 3, we developed a rigorous link between graph AC flow and the graph MBO scheme. In the continuum, those two flows both have important connections to mean curvature flow (MCF). *It was therefore conjectured by Van Gennip, Guillen, Osting, and Bertozzi* [17] *that these connections would translate into the graph context, i.e. that graph AC flow and the graph MBO scheme would have some link to the graph MCF defined in that paper. In this chapter, we will first present some promising* Γ*-convergence results that support this conjecture. However, we will then demonstrate a key flaw in the* [17] *definition of graph MCF which prevents any such connection. Finally, we will present a definition that avoids this flaw and formally resembles the MBO scheme, and give directions for future research.*

Parts of this chapter have been published in *SIAM J. Math. Anal.* 52 (2020) [10].

7.1. The continuum background

In this section, we shall give a brief overview of mean curvature flow (MCF) in the continuum. We will not go into too many details of the continuum setting, as such details lie beyond the scope of this thesis. In the continuum, a closed oriented hypersurface $\Sigma_t \subset \mathbb{R}^n$ is defined to evolve under MCF when the normal velocity at a point $x \in \Sigma_t$ is the mean of the principal curvatures at x. However, this definition can break down when the evolving surface develops singularities, so a more general definition was desired.

The first such reformulation was developed by Brakke [8] using tools from geometric measure theory. A later and highly fruitful reformulation was the *level-set approach* developed by Osher and Sethian [22], Evans and Spruck [15], and Chen, Giga, and Goto [11]. Supposing that $\phi : \mathbb{R}^n \times [0,\infty) \to \mathbb{R}$ satisfies: $\phi(x,0)$ is continuous in x, $\phi(x,0) > 0$ for x "inside" Σ_0, $\phi(x,0) < 0$ for x "outside" Σ_0, $\phi(x,0) = 0$ for $x \in \Sigma_0$, and ϕ is a weak (viscosity) solution to

$$\frac{\partial \phi}{\partial t} = -|\nabla \phi| \operatorname{div}\left(\frac{\nabla \phi}{|\nabla \phi|}\right), \tag{7.1}$$

then Σ_t evolves by MCF if and only if $\Sigma_t = \{\phi(\cdot,t) = 0\}$. That is, MCF trajectories can be reformulated as level sets of viscosity solutions to (7.1). As we will briefly discuss in this chapter, this reformulation has inspired the definition of graph MCF used by Elmoataz and co-authors, e.g. in El Chakik *et al.* [13].

Another reformulation, relevant to the definition of graph MCF from Van Gennip *et al.* [17] which shall be the focus of this chapter, is the variational formulation devised by Almgren, Taylor, and Wang [2]. Let Φ be the surface area functional, i.e. the map which sends a surface to the area of that surface. Then they define the energy

$$\mathscr{E}(\Omega, \Omega', \delta t) := \Phi(\partial \Omega') + \frac{1}{\delta t} \int_{\Omega \Delta \Omega'} \operatorname{dist}(x, \partial \Omega)\, dx \tag{7.2}$$

where $\Omega, \Omega' \subseteq \mathbb{R}^n$, $\partial \Omega'$ is the boundary of Ω', and $\Omega \Delta \Omega'$ is the symmetric difference of Ω and Ω'. Then if we define approximate discrete-time flows by

$$\Omega_{k+1}^{\delta t} \in \operatorname*{argmin}_{\Omega} \mathscr{E}\left(\Omega_k^{\delta t}, \Omega, \delta t\right), \qquad \Omega^{\delta t}(t) := \Omega_{\lfloor t/\delta t \rfloor}^{\delta t}$$

they define a *flat* Φ *curvature flow* as one that arises as a limit of approximate flows as $\delta t \to 0$ along some subsequence. These coincide with MCF when MCF does not exhibit singularities. So we can think of MCF as a kind of gradient flow of the surface area of Σ_t, as it is a limit of a kind of generalised minimising movement. A note that will be important for later is that $\Phi(\partial\Omega)$ is given by the *total variation* of χ_Ω, so MCF is a kind of gradient flow of total variation. A similar minimising movements scheme was developed by Luckhaus and Sturzenhecker [20] for their weak formulation of MCF.

Finally, it is well-studied that, in the continuum, AC flow, the MBO scheme, and MCF are interrelated in important ways. The MBO scheme was developed in [6] as a means of approximating motion according to MCF, and that paper gave

a formal analysis showing that diffusion of a set locally corresponded to motion with curvature dependent velocity, suggesting a convergence as the MBO time-step (corresponding to τ in the graph case) goes to zero. This formal analysis was then supported by rigorous convergence proofs by Evans [16] and Barles and Georgelin [3], making use of the level-set formulation of MCF. Recently, Swartz and Kwan Yip [24] presented an elementary proof of the convergence via the weak formulation of MCF in Luckhaus and Sturzenhecker [20]. The connections between Ginzburg–Landau dynamics and mean curvature flow have been extensively studied, dating back to a formal analysis by Allen and Cahn [1]. The basic convergence result, see e.g. [9, 14, 23], is that as $\varepsilon \to 0$ the solution to the AC flow tends to a phase-separation with the interface evolving by MCF. Thus a method of approximating MCF is as a singular limit of "phase fields" evolving under AC flow.[1] These results suggest the conjecture that graph MCF should also be linked to the graph MBO scheme and AC flow.

7.2. Γ-convergence results

A positive answer to the question of linking the graph MBO scheme, AC flow, and MCF has been suggested by Γ-convergence[2] results linking the associated energies of graph AC flow [19] and the MBO scheme [18] to graph *total variation*

$$\mathrm{TV}(u) := \frac{1}{2}\sum_{i,j\in V} \omega_{ij}|u_i - u_j|$$

which, based on the continuum analogy, should be a Lyapunov functional for any reasonable definition of graph MCF. We here extend these results to GL_ε and $\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}$ with the double-obstacle potential, and to the SDIE Lyapunov functionals H and H_0 (with and without fidelity forcing, respectively).

First, let us define Γ-convergence.

Definition 7.2.1 (Γ-convergence, see [7, Definitions 1.5 and 1.45])**.** *Let X be a metric space. A sequence $f_n : X \to \mathbb{R} \cup \{\pm\infty\}$* Γ-converges *to a function $f : X \to \mathbb{R} \cup \{\pm\infty\}$ if for all $x \in X$:*

(i) (Lim-inf inequality) For all sequences x_n in X such that $x_n \to x$

$$f(x) \le \operatorname{liminf}_n f_n(x_n),$$

(ii) (Existence of a recovery sequence for the f_n and f) There exists a sequence $\bar{x}_n$ in X such that $\bar{x}_n \to x$ such that

$$f(x) = \lim_n f_n(\bar{x}_n).$$

Note that there are numerous equivalent conditions for (ii). Finally, for a continuous parameter α defining a family of functions $f_\alpha : X \to \mathbb{R} \cup \{\pm\infty\}$, we define f to be the Γ-limit of f_α as $\alpha \to 0$ if for all sequences α_n with $\alpha_n \to 0$, f_{α_n} Γ-converges to f.

[1] See [4, 9, 14] for details on this method.

[2] For details on Γ-convergence, see e.g. Braides [7] and Dal Maso [21].

This notion of convergence is important because of the following theorem.

Theorem 7.2.2 (See [7, Theorem 1.21]). *Let X be a metric space, and let $f_n : X \to \mathbb{R} \cup \{\pm\infty\}$ be a sequence of functions with Γ-limit f, such that there exists a non-empty compact set $K \subseteq X$ such that for all n, $\inf_X f_n = \inf_K f_n$. Then there exists a minimiser of f in X and $\min_X f = \lim_n \inf_X f_n$. Furthermore, if x_n is a sequence contained within a compact subset of X, such that $\lim_n f_n(x_n) = \lim_n \inf_X f_n$, then every accumulation point of x_n is a minimiser of f. In particular, if x_n is a minimiser of f_n for all n and $x_n \to x$, then x is a minimiser of f.*

This makes Γ-convergence ideally suited as a tool for investigating anything related to minimisation problems. The AC flow monotonically decreases the Ginzburg–Landau functional (and likewise for the SDIE schemes and their respective Lyapunov functionals) and we will seek to define MCF so that it monotonically decreases TV. Therefore, if we have Γ-convergences of the former functionals to TV that is very promising, and will suggest that these flows are indeed linked.

We also note another useful property, which will come in handy below.

Proposition 7.2.3 (Cf. [7, Remark 1.7]). *Let X be a metric space, $f_n : X \to \mathbb{R} \cup \{\pm\infty\}$ be a sequence of functions with Γ-limit f, and $g_n : X \to \mathbb{R} \cup \{\pm\infty\}$ be a sequence of functions uniformly converging on X to $g : X \to \mathbb{R} \cup \{\pm\infty\}$, a continuous function. Then $f_n + g_n$ Γ-converges to $f + g$. In particular, by taking $g_n \equiv g$, $f_n + g$ Γ-converges to $f + g$.*

Proof. Fix $x \in X$. We prove each condition in turn. To show the lim-inf inequality, note first that for all sequences x_n, $\liminf_n f_n(x_n) + g_n(x_n) \geq \liminf_n f_n(x_n) + \liminf_n g_n(x_n)$. Thus since f_n Γ-converges to f, it suffices to show that for $x_n \to x$, $\liminf_n g_n(x_n) \geq g(x)$. But by the uniform convergence of g_n to g on X, for all $\varepsilon > 0$ there exists N such that for all $n > N$ and $x' \in X$, $g_n(x') > g(x') - \varepsilon$. Therefore

$$\liminf_n g_n(x_n) \geq \liminf_n g(x_n) - \varepsilon = g(x) - \varepsilon$$

with the latter equality following from the continuity of g. The lim-inf inequality follows.

Next, since f_n Γ-converges to f we have a recovery sequence for the f_n and f, i.e. a sequence $\bar{x}_n \to x$ such that $f_n(\bar{x}_n) \to f(x)$. To show that $\bar{x}_n$ is a recovery sequence for the $f_n + g_n$ and $f + g$, it therefore suffices to show that $g_n(\bar{x}_n) \to g(x)$. This follows since

$$|g_n(\bar{x}_n) - g(x)| \leq |g_n(\bar{x}_n) - g(\bar{x}_n)| + |g(\bar{x}_n) - g(x)| \to 0$$

with the former term converging to 0 because of the uniform convergence of g_n to g, and the latter term because of the continuity of g. □

Next, let us define the following function on $\mathcal{V}_{[0,1]}$:

$$\mathrm{TV}_0(u) := \begin{cases} \frac{1}{2}\,\mathrm{TV}(u), & u \in \mathcal{V}_{[0,1]} \cap \mathcal{V}_{\{0,1\}}, \\ \infty, & u \in \mathcal{V}_{[0,1]} \setminus \mathcal{V}_{\{0,1\}}. \end{cases}$$

Then we have the following Γ-convergences.

Theorem 7.2.4 (Cf. [19, Theorem 3.1]). *The Ginzburg–Landau functional* GL_ε *with double-obstacle potential defined in* (3.4) *has* Γ*-limit in* $\mathcal{V}_{[0,1]}$:

$$\Gamma - \lim_{\varepsilon \downarrow 0} \mathrm{GL}_\varepsilon = \mathrm{TV}_0 .$$

Proof. The proof is more or less identical to its counterpart in [19]. Let $u_\varepsilon \to u$ for $u_\varepsilon, u \in \mathcal{V}_{[0,1]}$. Suppose $u_i \in (0,1)$ for some $i \in V$, then eventually $(u_\varepsilon)_i \in (0,1)$ and $\mathrm{GL}_\varepsilon(u_\varepsilon) \geq \frac{1}{2\varepsilon} d_i^r (u_\varepsilon)_i (1-(u_\varepsilon)_i) \to \infty$, so $\mathrm{TV}_0(u) \leq \liminf_{\varepsilon\to 0} \mathrm{GL}_\varepsilon(u_\varepsilon)$. Now if $u \in \mathcal{V}_{\{0,1\}}$ then $\mathrm{TV}_0(u) = \frac{1}{2}||\nabla u||_{\mathcal{V}}^2 = \lim_{\varepsilon\to 0} \frac{1}{2}||\nabla u_\varepsilon||_{\mathcal{V}}^2 \leq \liminf_{\varepsilon\to 0} \mathrm{GL}_\varepsilon(u_\varepsilon)$.

Now let $u \in \mathcal{V}_{[0,1]}$ and choose the recovery sequence $\bar{u}_\alpha \equiv u$. If $u_i \in (0,1)$ for some $i \in V$, then $\mathrm{GL}_\varepsilon(u) \geq \frac{1}{2\varepsilon} d_i^r u_i (1-u_i) \to \infty$ so $\mathrm{TV}_0(u) = \lim_{\varepsilon\to 0} \mathrm{GL}_\varepsilon(u)$. If $u \in \mathcal{V}_{\{0,1\}}$ then $\mathrm{GL}_\varepsilon(u) = \frac{1}{2}||\nabla u||_{\mathcal{V}}^2 = \mathrm{TV}_0(u)$ so again $\mathrm{TV}_0(u) = \lim_{\varepsilon\to 0} \mathrm{GL}_\varepsilon(u)$. □

Corollary 7.2.5. *Let* $g_{\mu,\tilde{f}}(u) := \frac{1}{2}\langle u - \tilde{f}, M(u-\tilde{f})\rangle_{\mathcal{V}}$ *where* $M := \mathrm{diag}(\mu)$. *Then the Ginzburg–Landau functional with fidelity forcing* $\mathrm{GL}_{\varepsilon,\mu,\tilde{f}}$ *with double-obstacle potential defined in* (3.9) *has* Γ*-limit in* $\mathcal{V}_{[0,1]}$:

$$\Gamma - \lim_{\varepsilon \downarrow 0} \mathrm{GL}_{\varepsilon,\mu,\tilde{f}} = \mathrm{TV}_0 + g_{\mu,\tilde{f}}.$$

Proof. Follows immediately from Proposition 7.2.3, since $g_{\mu,\tilde{f}}$ is continuous and $\mathrm{GL}_{\varepsilon,\mu,\tilde{f}} = \mathrm{GL}_\varepsilon + g_{\mu,\tilde{f}}$. □

Corollary 7.2.6. *The Lyapunov functional* H_0 *for the SDIE scheme without fidelity forcing defined in* (4.40a) *has* Γ*-convergence in* $\mathcal{V}_{[0,1]}$:

$$\Gamma - \lim_{\varepsilon \downarrow 0,\, 0<\tau\leq\varepsilon} \frac{1}{2\tau} H_0 = \mathrm{TV}_0,$$

and the Lyapunov functional for the SDIE scheme with fidelity forcing H *defined in* (4.40b) *has* Γ*-convergence in* $\mathcal{V}_{[0,1]}$:

$$\Gamma - \lim_{\varepsilon \downarrow 0,\, 0<\tau\leq\varepsilon} \frac{1}{2\tau} H = \mathrm{TV}_0 + g_{\mu,\tilde{f}} - \frac{1}{2}\langle \tilde{f}, M\tilde{f}\rangle_{\mathcal{V}},$$

recalling the notation from Corollary 7.2.5.

Proof. Fix sequences τ_n and ε_n such that for all n, $0 < \tau_n \leq \varepsilon_n$, and $\varepsilon_n \downarrow 0$ (and hence $\tau_n \downarrow 0$).

Recall from Proposition 4.5.11 the notation $H_\tau(u) := \frac{1}{2\tau} H(u)$ and $H_{0,\tau}(u) := \frac{1}{2\tau} H_0(u)$. Furthermore, recall that for $u \in \mathcal{V}_{[0,1]}$

$$H_\tau(u) = \mathrm{GL}_{\varepsilon,\mu,\tilde{f}}(u) - \frac{1}{2}\langle \tilde{f}, M\tilde{f}\rangle_{\mathcal{V}} + \frac{1}{2}\tau\langle u, Q_\tau(Au - 2f)\rangle_{\mathcal{V}}$$
$$H_{0,\tau}(u) = \mathrm{GL}_\varepsilon(u) + \frac{1}{2}\tau\langle u, Q_\tau' u\rangle_{\mathcal{V}}$$

where $Q_\tau := \tau^{-2}(F_\tau(A) - \tau I)$ (for F_τ defined by Definition 3.2.5), $Q'_\tau := \tau^{-2}(I - \tau\Delta - e^{-\tau\Delta})$, and $f := \mu \odot \tilde{f}$ where $\odot$ is the Hadamard product. By Proposition 7.2.3, Theorem 7.2.4, and Corollary 7.2.5, to prove the result it will suffice to show that

$$H_{0,\tau_n} - \mathrm{GL}_{\varepsilon_n} \to 0, \text{ and}$$
$$H_{\tau_n} - \mathrm{GL}_{\varepsilon_n,\mu,\tilde{f}} \to -\frac{1}{2}\langle \tilde{f}, M\tilde{f}\rangle_{\mathcal{V}},$$

both uniformly in n. By the above expressions for H_τ and $H_{0,\tau}$, it therefore suffices to prove that $\frac{1}{2}\tau\langle u, Q'_\tau u\rangle_{\mathcal{V}}$ and $\frac{1}{2}\tau\langle u, Q_\tau(Au - 2f)\rangle_{\mathcal{V}}$ both tend to zero uniformly as $\tau \to 0$. This was proved in the proof of Proposition 4.5.11. □

Note 38. *Taking $\tau = \varepsilon$ and considering $J(u) := \langle 1 - u, e^{-\tau\Delta}u\rangle_{\mathcal{V}}$, the Lyapunov functional for the MBO scheme (see* [17, Proposition 4.6] *and Lemma 4.4.1), we have that $H_0 = J$ and so in $\mathcal{V}_{[0,1]}$:*

$$\Gamma - \lim_{\tau\downarrow 0} \frac{1}{\tau} J|_{\mathcal{V}_{[0,1]}} = 2\,\mathrm{TV}_0\,.$$

This is a special case of the result of [18, Theorem 5.10].

7.3. The Van Gennip *et al.* [17] definition

In [17], motivated by the variational formulation of MCF in [2], graph MCF was defined as the minimisation scheme

$$S_{n+1} \in \operatorname*{argmin}_{S\subseteq V} \mathrm{TV}(\chi_S) + \frac{1}{\delta t}\langle \chi_S - \chi_{S_n}, (\chi_S - \chi_{S_n})d^{\Sigma_n}\rangle_{\mathcal{V}} \tag{7.3}$$

where $\Sigma_n := \{i \in V \mid \exists j \in V \text{ s.t. } \omega_{ij} > 0 \text{ and } (\chi_{S_n})_i \neq (\chi_{S_n})_j\}$ is the *graph boundary* of S_n, and $d_i^{\Sigma_n}$ is the graph distance from i to Σ_n, i.e. the length of the shortest path from i to a vertex in Σ_n (see [17, Definition 2.3] for details).

In this section, we will describe two issues with connecting this scheme to the MBO scheme, one which is very serious and another which is in a sense harmless but suggests a way forward.

7.3.1. The key issue

Let $G = (V, E, \omega)$ be a graph as defined in chapter 2, and for $\alpha > 0$ let $G^\alpha = (V, V^2, \omega^\alpha)$ be the complete graph defined by $\omega^\alpha_{ij} := \omega_{ij}$ if $ij \in E$ and $\omega^\alpha_{ij} := \alpha$ otherwise. This construction is illustrated in Fig. 7.1.

Then $\Delta^\alpha = \Delta + \mathcal{O}(\alpha)$ relative to the limit $\alpha \downarrow 0$, so for α sufficiently small the AC flow and MBO scheme will be essentially the same on G and G^α. More precisely we have the following result.

Theorem 7.3.1. *Let $u, u_\alpha, v, v_\alpha \in \mathcal{V}_{[0,1],t\in[0,\infty)}$ be defined by $u(0) = u_\alpha(0) =: u_0$,*

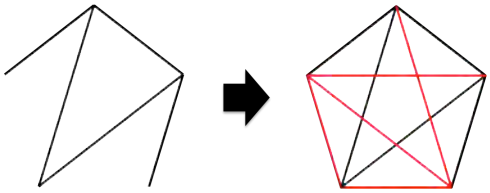


Figure 7.1: Modification of G (left) into G^α (right); red edges have weight α.

$v(0) = v_\alpha(0) =: v_0$, *and*

$$\frac{du}{dt}(t) = -\Delta u(t) - \frac{1}{\varepsilon} f(u), \qquad \frac{dv}{dt}(t) = -\Delta v(t),$$
$$\frac{du_\alpha}{dt}(t) = -\Delta^\alpha u_\alpha(t) - \frac{1}{\varepsilon} f(u_\alpha), \qquad \frac{dv_\alpha}{dt}(t) = -\Delta^\alpha v_\alpha(t),$$

for $f \in C^{0,1}([0,\infty); \mathcal{V})$ *(e.g.,* f *arising from the gradient of a* C^2 *double-well potential). Then there exists* $C \geq 0$*, independent of* α*, such that* $||u(t) - u_\alpha(t)||_\mathcal{V} \leq \mathcal{O}(\alpha) t e^{Ct}$ *and* $||v(t) - v_\alpha(t)||_\mathcal{V} \leq \mathcal{O}(\alpha) t e^{Ct}$*.*

Proof. Let $E := \Delta^\alpha - \Delta = \mathcal{O}(\alpha)$. Then

$$\begin{aligned} ||u(t) - u_\alpha(t)||_\mathcal{V} &= \left|\left| \int_0^t \Delta u(s) - \Delta^\alpha u_\alpha(s) + \frac{1}{\varepsilon} \left(f(u(s)) - f(u_\alpha(s)) \right) ds \right|\right|_\mathcal{V} \\ &\leq \int_0^t \left(||\Delta|| + \frac{1}{\varepsilon} ||f||_{C^{0,1}} \right) ||u(s) - u_\alpha(s)||_\mathcal{V} \, ds + t||E||\,||\mathbf{1}||_\mathcal{V} \end{aligned}$$

and so by Grönwall's integral inequality [5]

$$||u(t) - u_\alpha(t)||_\mathcal{V} \leq t||E||\,||\mathbf{1}||_\mathcal{V} e^{t\left(||\Delta|| + \frac{1}{\varepsilon}||f||_{C^{0,1}}\right)}.$$

The argument for v, v_α follows by setting $f = 0$. □

Note 39. *For the MBO scheme it is possible for this* $\mathcal{O}(\alpha)$ *difference in the diffused state to make an* $\mathcal{O}(1)$ *difference to the MBO update, but it is unlikely as it could only occur if the diffused state were very close to the threshold value. The SDIE scheme relaxes MBO's discontinuous thresholding to a Lipschitz one (see Theorem 4.2.1), avoiding this issue. Note also that the above AC flow doesn't include the double-obstacle AC flow that was the main subject of chapter 3, due to the* $C^{0,1}$ *condition on* f *not being satisfied. To derive a similar result for that flow, observe that by Theorem 3.4.8 the difference between double-obstacle AC flows* u *and* u_α *is continuous in* α *except for the contribution of a term of the form*

$$\int_0^t \beta(s) - \beta_\alpha(s) \, ds$$

for $\beta(s) \in \mathcal{B}(u(s))$ *and* $\beta_\alpha(s) \in \mathcal{B}(u_\alpha(s))$*. By Theorem 3.4.4, these terms vary discontinuously in* u *and* u_α*, however if* $u_i(s), (u_\alpha(s))_i$ *are both in* $(0,1)$ *then* $\beta_i(s) =$

$(\beta_\alpha(s))_i$, *and if both equal 0 or both equal 1, then* $(\beta_\alpha(s))_i - \beta_i(s) = \varepsilon(\Delta(u_\alpha(s) - u(s)) + Eu_\alpha(s))_i$. *Hence the discontinuity will only produce an* $\mathcal{O}(1)$ *difference under specific and limited circumstances.*

Now consider the impact of moving from G to G^α on (7.3). If $S_n \notin \{\emptyset, V\}$, then by definition the graph boundary $\Sigma_n^\alpha = V$ and so $d^{\Sigma_n^\alpha} = \mathbf{0}$. Therefore on G^α the objective function of (7.3) collapses to $\mathrm{TV}(\chi_S)$, so we only get solutions $S_{n+1}^\alpha \in \{\emptyset, V\}$ which in general will be very different from the behaviour on G.

In summary, moving from G to G^α we observe an $\mathcal{O}(\alpha)$ change in the building blocks of the MBO scheme and AC flow, though this can occasionally induce an $\mathcal{O}(1)$ change via discontinuous elements in those flows. By contrast, the change in MCF is massive, immediately excluding all non-trivial segmentations, no matter how small α is.

7.3.2. A difference between MCF and the MBO scheme for large time steps

There is a less critical but nonetheless illuminating difference between MCF as defined by (7.3) and the MBO scheme. Let us consider the behaviour as δt in (7.3) and τ, the MBO time step, are both taken very large.

From [17, Lemma 2.6(c)], for $\bar{u} \neq \frac{1}{2}$ and τ sufficiently large (depending on $\bar{u}$), we have $||e^{-\tau\Delta}u - \bar{u}\mathbf{1}||_\infty < |\bar{u} - \frac{1}{2}|$. It follows that the MBO update of u for τ sufficiently large is $\mathbf{1}$ if $\bar{u} > \frac{1}{2}$ and $\mathbf{0}$ if $\bar{u} < \frac{1}{2}$.

Now consider (7.3) for δt very large. One can check that the distance term is bounded above by $\mathrm{diam}(G)\langle \mathbf{1}, \mathbf{1}\rangle_\mathcal{V}$ where $\mathrm{diam}(G)$ is the diameter of G (i.e. the maximal graph distance between vertices of G), and $\mathrm{TV}(\chi_S) \geq \min_{ij\in E} \omega_{ij}$ if $S \notin \{\emptyset, V\}$. Hence for $\delta t > \mathrm{diam}(G)||\mathbf{1}||_\mathcal{V}^2 \left(\min_{ij\in E} \omega_{ij}\right)^{-1}$, the only possible minimisers of (7.3) are $S \in \{\emptyset, V\}$. Since in either case $\mathrm{TV}(\chi_S) = 0$, $S = \emptyset$ is a minimiser if and only if $S = \emptyset$ yields a lower value for the latter term in (7.3) than $S = V$, i.e. if and only if

$$\langle \chi_{S_n}, \chi_{S_n} d^{\Sigma_n} \rangle_\mathcal{V} \leq \langle \mathbf{1} - \chi_{S_n}, (\mathbf{1} - \chi_{S_n}) d^{\Sigma_n} \rangle_\mathcal{V}.$$

In summary, for sufficiently large time steps we observe the same basic behaviour between MCF and the MBO scheme, that is, the updates are constantly 0 or constantly 1, but the condition governing which is chosen is different. Although our chief concern is really for small time steps, as that is when the continuum theory suggests that the MBO scheme and MCF should be linked, it will be promising if they also agree in the very large time step case.

7.4. An improved definition

Given the issues with defining graph MCF by (7.3), we seek a new definition. First, we observe that for $u \in \mathcal{V}_{\{0,1\}}$

$$\langle u, \Delta u \rangle_\mathcal{V} = \frac{1}{2} \sum_{i,j\in\mathcal{V}} \omega_{ij}(u_i - u_j)^2 = \frac{1}{2} \sum_{i,j\in\mathcal{V}} \omega_{ij}|u_i - u_j| = \mathrm{TV}(u). \tag{7.4}$$

As shown in [17, p. 52], (7.4) has a useful consequence: the Lyapunov functional J_0 (see Lemma 4.4.1, note that this is denoted "J" in [17]) for the graph MBO scheme has the property that

$$J_0(\chi_S) := \langle \mathbf{1} - \chi_S, e^{-\tau\Delta}\chi_S\rangle_{\mathcal{V}} = \tau\,\mathrm{TV}(\chi_S) + R_S(\tau)$$

where $R_S(\tau) := \sum_{k\geq 2}(-1)^k \frac{\tau^k}{k!}\langle \chi_{S^c}, \Delta^k \chi_S\rangle_{\mathcal{V}}$. Therefore, the MBO update of $S_n \subseteq V$ given by Definition 3.2.1 obeys

$$\begin{aligned} S_{n+1} \in \operatorname*{argmin}_{S\subseteq V} \langle \mathbf{1} - 2e^{-\tau\Delta}\chi_{S_n}, \chi_S\rangle_{\mathcal{V}} \\ &= J(\chi_S) + \left\langle \chi_S - \chi_{S_n}, e^{-\tau\Delta}(\chi_S - \chi_{S_n})\right\rangle_{\mathcal{V}} - \left\langle \chi_{S_n}, e^{-\tau\Delta}\chi_{S_n}\right\rangle_{\mathcal{V}} \\ &\simeq \mathrm{TV}(\chi_S) + \frac{R_S(\tau) + ||e^{-\frac{1}{2}\tau\Delta}(\chi_S - \chi_{S_n})||_{\mathcal{V}}^2}{\tau}. \end{aligned}$$

Since $R_S(\tau)/\tau = \mathcal{O}(\tau)$, this suggests that the following definition of graph MCF:

$$S_{n+1} \in \operatorname*{argmin}_{S\subseteq V} \mathrm{TV}(\chi_S) + \frac{||e^{-\frac{1}{2}\tau\Delta}(\chi_S - \chi_{S_n})||_{\mathcal{V}}^2}{\tau} \tag{7.5}$$

might strongly resemble the MBO scheme for small τ. The MCF-like behaviour of this scheme is captured by the following proposition.

Proposition 7.4.1. *For S_{n+1} solving (7.5), $\mathrm{TV}(\chi_{S_{n+1}}) \leq \mathrm{TV}(\chi_{S_n})$, with equality if and only if $S_{n+1} = S_n$.*

Proof. By (7.5),

$$\mathrm{TV}(\chi_{S_{n+1}}) \leq \mathrm{TV}(\chi_{S_{n+1}}) + \frac{||e^{-\frac{1}{2}\tau\Delta}(\chi_{S_{n+1}} - \chi_{S_n})||_{\mathcal{V}}^2}{\tau} \leq \mathrm{TV}(\chi_{S_n}).$$

Finally, $e^{-\frac{1}{2}\tau\Delta}$ is invertible, so $e^{-\frac{1}{2}\tau\Delta}(\chi_{S_{n+1}} - \chi_{S_n}) = \mathbf{0}$ if and only if $\chi_{S_{n+1}} = \chi_{S_n}$. □

We note immediately that (7.5) avoids the key issue from the previous section, since the objective function is only $\mathcal{O}(\alpha)$ different on G^α. Furthermore, note that since $\chi_S - \chi_{S_n} \in \mathcal{V}_{[-1,1]}$, it follows that $e^{-\frac{1}{2}\tau\Delta}(\chi_S - \chi_{S_n}) \in \mathcal{V}_{[-1,1]}$ since $e^{-\frac{1}{2}\tau\Delta}$ is a non-negative matrix (see the proof of Theorem 3.2.6, and consider the $A = \Delta$ special case). Therefore the latter term in (7.5) is bounded above by $||\mathbf{1}||_{\mathcal{V}}^2/\tau$, and so for $\tau > ||\mathbf{1}||_{\mathcal{V}}^2 \left(\min_{ij\in E}\omega_{ij}\right)^{-1}$ the only possible minimisers are $S \in \{\emptyset, V\}$. Then in both cases $\mathrm{TV}(\chi_S) = 0$, and so $S = \emptyset$ is a minimiser if and only if

$$||e^{-\frac{1}{2}\tau\Delta}\chi_{S_n}||_{\mathcal{V}}^2 \leq ||e^{-\frac{1}{2}\tau\Delta}(\mathbf{1} - \chi_{S_n})||_{\mathcal{V}}^2 = ||\mathbf{1} - e^{-\frac{1}{2}\tau\Delta}\chi_{S_n}||_{\mathcal{V}}^2.$$

By expanding both sides and simplifying, this is equivalent to (recalling the notation $\mathcal{M}(u) := \langle u, \mathbf{1}\rangle_{\mathcal{V}}$ from Definition 3.2.9)

$$\mathcal{M}(e^{-\frac{1}{2}\tau\Delta}\chi_{S_n}) = \mathcal{M}(\chi_{S_n}) \leq \frac{1}{2}\mathcal{M}(\mathbf{1})$$

which is the same condition as for the large τ MBO update.

We can rewrite (7.5) in a way that makes the connection to the MBO scheme more explicit. By (7.4) for $u_n := \chi_{S_n}$ and $u := \chi_S$, (7.5) is equivalent to

$$\operatorname*{argmin}_{u \in \mathcal{V}_{\{0,1\}}} \langle u, \tau\Delta u\rangle_{\mathcal{V}} + \langle u - u_n, e^{-\tau\Delta}(u - u_n)\rangle_{\mathcal{V}} \simeq -\tau^2 \langle u, Q'_\tau u\rangle_{\mathcal{V}} + ||u - e^{-\tau\Delta}u_n||^2_{\mathcal{V}}$$

where $Q'_\tau := \tau^{-2}\left(I - \tau\Delta - e^{-\tau\Delta}\right)$. Thus if we suppress the $\mathcal{O}(\tau^2)$ terms (7.5) has solution equal to

$$\operatorname*{argmin}_{u \in \mathcal{V}_{\{0,1\}}} ||u - e^{-\tau\Delta}u_n||^2_{\mathcal{V}}$$

which is the MBO thresholding of $e^{-\tau\Delta}u_n$. Thus up to the influence of $\mathcal{O}(\tau^2)$ terms, (7.5) corresponds to an MBO update.

However this result, although very promising, may not be as significant as it might appear. As was demonstrated in [17, Theorem 4.2] (see also section 4.3.2), the MBO scheme freezes if τ is taken too small. We now prove a similar pinning result for (7.5).

Theorem 7.4.2. *Suppose that τ satisfies*

$$\tau e^{\tau||\Delta||} < \frac{\min_{i \in V} d_i^r}{\mathrm{TV}_{max}} \tag{7.6}$$

where $\mathrm{TV}_{max} := \max_{S \subseteq V} \mathrm{TV}(\chi_S)$. Then for all $S' \subseteq V$, if $S_n = S'$ and S_{n+1} solves (7.5), then $S_{n+1} = S'$.

Note 40. *The condition (7.6) is equivalent to*

$$\tau < ||\Delta||^{-1} W_L\left(\frac{||\Delta|| \min_{i \in V} d_i^r}{\mathrm{TV}_{max}}\right)$$

where W_L is the Lambert W-function [12], which for $x \in [0, \infty)$ can be defined as the unique function on $[0, \infty)$ satisfying $W_L(x)e^{W_L(x)} = x$.

Proof. Note first that, since $e^{-\frac{1}{2}\tau\Delta}$ is self-adjoint, the $||e^{-\frac{1}{2}\tau\Delta}(\chi_S - \chi_{S_n})||^2_{\mathcal{V}}$ term in (7.5) is equal to $\langle \chi_S - \chi_{S_n}, e^{-\tau\Delta}(\chi_S - \chi_{S_n})\rangle_{\mathcal{V}}$. Next, note that $e^{-\tau\Delta}$ has smallest eigenvalue $e^{-\tau||\Delta||}$ since Δ is positive semi-definite. From these two observations, it follows that for all $S, S_n \subseteq V$,

$$||e^{-\frac{1}{2}\tau\Delta}(\chi_S - \chi_{S_n})||^2_{\mathcal{V}} \geq e^{-\tau||\Delta||}||\chi_S - \chi_{S_n}||^2_{\mathcal{V}}.$$

Let S' be an arbitrary subset of V, $S_n = S'$, and let $\mathcal{G}$ denote the objective function of (7.5). Then by the above inequality, since TV is non-negative it follows that if $S \neq S'$ then

$$\mathcal{G}(S) \geq \tau^{-1}e^{-\tau||\Delta||}||\chi_S - \chi_{S'}||^2_{\mathcal{V}} \geq \tau^{-1}e^{-\tau||\Delta||} \min_{i \in V} d_i^r.$$

Furthermore,

$$\mathcal{G}(S') = \mathrm{TV}(\chi_{S'}) \leq \mathrm{TV}_{max}\,.$$

Therefore, for $S = S'$ to be the unique minimiser of (7.5), it suffices that

$$\mathrm{TV}_{max} \leq \tau^{-1} e^{-\tau||\Delta||} \min_{i \in V} d_i^r$$

which rearranges to give (7.6). □

Hence, for τ below a certain threshold there is a trivial equivalence between (7.5) and the MBO scheme. The regime of interest therefore is for τ above these thresholds, and so we cannot treat $\mathcal{O}(\tau^2)$ terms as being negligible.

7.5. Future work

In the above, we have given a formal argument for a connection between (7.5) and the graph MBO scheme. A major direction for future work will be making this connection rigorous (in particular, establishing this connection for τ above the threshold at which (7.5) and the MBO scheme pin, where the $\mathcal{O}(\tau^2)$ terms are not negligible), and explicitly quantifying the error between these two flows.

A more broad direction will be investigating whether (7.5) "deserves" to be called an MCF. One explicit way to answer this will be to compare (7.5) to the definition of graph MCF considered by El Chakik, Elmoataz, and Desquesnes in [13]. We will briefly describe this other definition, which is motivated by the well-known level-set formulation of continuum MCF (see e.g. Osher and Sethian [22]). The authors of [13] define the graph mean curvature[3] by

$$K_i(u) := \sum_{j \in V} \frac{\omega_{ij}}{d_i} \operatorname{sgn}(\nabla u)_{ij},$$

and define (respectively) upwind and downwind gradient norms of $u \in \mathcal{V}$ by[4]

$$\left|\left|(\nabla^+ u)_i\right|\right|_p^p := \sum_{j \in V} \omega_{ij} \left|(\nabla u)_{ij}^+\right|^p, \qquad \left|\left|(\nabla^+ u)_i\right|\right|_\infty := \max_{j \in V} \omega_{ij} \left|(\nabla u)_{ij}^+\right|,$$

$$\left|\left|(\nabla^- u)_i\right|\right|_p^p := \sum_{j \in V} \omega_{ij} \left|(\nabla u)_{ij}^-\right|^p, \qquad \left|\left|(\nabla^- u)_i\right|\right|_\infty := \max_{j \in V} \omega_{ij} \left|(\nabla u)_{ij}^-\right|,$$

where superscript $+$ and $-$ denote respectively the positive and negative part operators (i.e. $x^+ := \max\{x, 0\}$ and $x^- := -\min\{x, 0\}$), and they define graph MCF as the ODE flow:

$$\frac{du_i}{dt} = K_i^+(u(t)) \left|\left|(\nabla^+ u(t))_i\right|\right|_p - K_i^-(u(t)) \left|\left|(\nabla^- u(t))_i\right|\right|_p \tag{7.7}$$

[3] For $u = \chi_S$ we note a similarity between this definition and [17, Definition 3.2].

[4] We adapt their notation to align with that of this thesis. Recall from chapter 2 that $(\nabla u)_{ij} := u_j - u_i$ for i and j neighbours and $(\nabla u)_{ij} := 0$ otherwise.

for $p \in [1, \infty]$. We note that, for $r = 1$, it is easy to see that $-K(u)$ is the gradient of TV at u, and that therefore (7.7) monotonically decreases TV along trajectories.[5] A topic for future research will be to investigate whether solutions of (7.5) can be related to trajectories of this ODE.

[5] We leave the details of this demonstration as an exercise for the reader.

Bibliography

[1] Samuel M Allen and John W Cahn. "A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening". In: *Acta metallurgica* 27.6 (1979), pp. 1085–1095.

[2] Fred Almgren, Jean E Taylor, and Lihe Wang. "Curvature-driven flows: a variational approach". In: *SIAM Journal on Control and Optimization* 31.2 (1993), pp. 387–438.

[3] Guy Barles and Christine Georgelin. "A simple proof of convergence for an approximation scheme for computing motions by mean curvature". In: *SIAM Journal on Numerical Analysis* 32.2 (1995), pp. 484–500.

[4] Guy Barles, H Mete Soner, and Panagiotis E Souganidis. "Front propagation and phase field theory". In: *SIAM Journal on Control and Optimization* 31.2 (1993), pp. 439–469.

[5] Richard Bellman. "The stability of solutions of linear differential equations". In: *Duke Mathematical Journal* 10.4 (1943), pp. 643–647. doi: 10.1215/S0012-7094-43-01059-2.

[6] J. Bence, B. Merriman, and S. Osher. "Diffusion generated motion by mean curvature". In: *CAM Report, 92-18, Department of Mathematics, University of California, Los Angeles* (1992).

[7] Andrea Braides. *Gamma-convergence for Beginners*. Oxford Scholarship Online. Oxford: Oxford University Press, 2007. doi: 10.1093/acprof:oso/9780198507840.001.0001.

[8] K Brakke. "The motion of a surface by its mean curvature". In: *Mathematical Notes, Princeton University Press, Princeton, NJ* (1978).

[9] Lia Bronsard and Robert V. Kohn. "Motion by mean curvature as the singular limit of Ginzburg-Landau dynamics". English (US). In: *Journal of Differential Equations* 90.2 (Apr. 1991), pp. 211–237. doi: 10.1016/0022-0396(91)90147-2.

[10] Jeremy Budd and Yves van Gennip. "Graph Merriman–Bence–Osher as a SemiDiscrete Implicit Euler Scheme for Graph Allen–Cahn Flow". In: *SIAM Journal on Mathematical Analysis* 52 (Jan. 2020), pp. 4101–4139. doi: 10.1137/19M1277394.

[11] Yun Gang Chen, Yoshikazu Giga, and Shun'ichi Goto. "Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations". In: *Journal of differential geometry* 33.3 (1991), pp. 749–786.

[12] R. M. Corless et al. "On the LambertW function". In: *Adv. Comput. Math.* 5 (1996), pp. 329–359. doi: 10.1007/BF02124750.

[13] Abdallah El Chakik, Abderrahim Elmoataz, and Xavier Desquesnes. "Mean curvature Flows on Graphs for Image and Manifold Restoration and Enhancement". In: *Signal Processing* 105 (Dec. 2014), pp. 449–463. doi: 10.1016/j.sigpro.2014.04.029.

[14] Lawrence C Evans, H Mete Soner, and Panagiotis E Souganidis. "Phase transitions and generalized motion by mean curvature". In: *Communications on Pure and Applied Mathematics* 45.9 (1992), pp. 1097–1123.

[15] Lawrence C Evans and Joel Spruck. "Motion of level sets by mean curvature. I". In: *Journal of Differential Geometry* 33.3 (1991), pp. 635–681.

[16] Lawrence C. Evans. "Convergence of an Algorithm for Mean Curvature Motion". In: *Indiana University Mathematics Journal* 42.2 (1993), pp. 533–557. issn: 00222518, 19435258. url: http://www.jstor.org/stable/24897106.

[17] Y. van Gennip et al. "Mean Curvature, Threshold Dynamics, and Phase Field Theory on Finite Graphs". In: *Milan Journal of Mathematics* 82 (2014), pp. 3–65.

[18] Yves van Gennip. "An MBO Scheme for Minimizing the Graph Ohta–Kawasaki Functional". In: *Journal of Nonlinear Science* 30.2 (Oct. 2020), pp. 2325–2373. doi: 10.1007/s00332-018-9468-8.

[19] Yves van Gennip and Andrea L. Bertozzi. *Gamma-convergence of graph Ginzburg-Landau functionals*. 2018. arXiv: 1204.5220 [math.AP].

[20] Stephan Luckhaus and Thomas Sturzenhecker. "Implicit time discretization for the mean curvature flow equation". In: *Calculus of variations and partial differential equations* 3.2 (1995), pp. 253–271.

[21] Gianni Dal Maso. *An Introduction to Γ-Convergence*. Vol. 8. Progress in Nonlinear Differential Equations and Their Applications. Basel: Birkhäuser, 1993. doi: 10.1007/978-1-4612-0327-8.

[22] Stanley Osher and James A Sethian. "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations". In: *Journal of computational physics* 79.1 (1988), pp. 12–49.

[23] Halil Mete Soner. "Ginzburg-Landau equation and motion by mean curvature, I: convergence". In: *J. Geom. Anal* 7.3 (1997), pp. 437–475.

[24] Drew Swartz and Nung Kwan Yip. "Convergence of diffusion generated motion to motion by mean curvature". In: *Communications in Partial Differential Equations* 42.10 (2017), pp. 1598–1643.

8

Conclusion

Roads go ever ever on,
Over rock and under tree,
By caves where never sun has shone,
By streams that never find the sea;
Over snow by winter sown,
And through the merry flowers of June,
Over grass and over stone,
And under mountains in the moon.
Roads go ever ever on,
Under cloud and under star.
Yet feet that wandering have gone
Turn at last to home afar.
Eyes that fire and sword have seen,
And horror in the halls of stone
Look at last on meadows green,
And trees and hills they long have known.

J. R. R. Tolkien, *The Hobbit*

Graph-based learning is an exciting approach to learning problems. Graphs provide a natural model for representing interconnected data, and as we have seen this perspective translates into concrete algorithms which are both efficient and accurate. Furthermore, the graph context is a fertile mathematical landscape, with many avenues for deep theoretical exploration. In particular, there are many fascinating questions that arise in the translation of continuum ideas and methods into this new framework.

In this thesis, we have made important strides forward on both the theoretical and applied sides of the "PDEs on graphs" strand of graph-based learning. On the theory side, we have rigorously proved that graph AC flow (with the double-obstacle potential) and the graph MBO scheme are linked by our SDIE scheme, and showed that this link is robust to the inclusion of both mass conservation and fidelity forcing constraints. We have furthermore proved a wealth of desirable theoretical properties of these flows/schemes, and it is our belief that this theoretical framework can be extended in the future to incorporate a wider family of flows, such as multi-class AC flow and multi-class MBO schemes. We have also made some small headway towards answering the theoretical question of how mean curvature flow on graphs should be defined, and whether it too can be linked to the graph MBO scheme and AC flow. Finding a key flaw in the Van Gennip *et al.* [5] definition of this flow, we have suggested an improved definition, and demonstrated a formal similarity between this newly defined flow and the MBO scheme. This suggests promising research directions of making this formal argument more rigorous, and of connecting this definition of graph mean curvature flow with other definitions in the literature, such as El Chakik *et al.* [4].

On the applications side, we first explored the application of graph PDE methods to image segmentation. Our primary question was whether our SDIE scheme might provide a better algorithm than the earlier MBO-based or AC-based segmentation algorithms of [1, 6]. Whilst for the example we considered the answer to that question turned out to unfortunately be no, along the way we investigated a number of refinements to those earlier algorithms, with the upshot that our algorithm produced a significantly more accurate segmentation than the segmentations of that example in previous work. Following this, we then turned to the more complicated task of reconstruction-segmentation. Traditionally this task would be performed sequentially, as a reconstruction followed by a segmentation, but a more powerful technique is to perform both reconstruction and segmentation jointly. We have developed a novel framework for performing this within the graph context, allowing for the use of our more sophisticated graph-based segmentation methods over the more straightforward Chan–Vese methods used in e.g. Corona *et al.* [3]. This work is still ongoing, but when it is mature we believe that it will have important applications in medical imaging, where many tasks (e.g. radiotherapy) involve solving a reconstruction-segmentation problem.

For our concluding remarks for the future, let us return to that wonderful quote of Dyson's, with which this thesis began. For most of this thesis, whether in presenting current work or considering work for the future, we have been occupying the perspective of a frog. Let us now consider the perspective of a bird. Graph-based

learning is, as we described in the introduction, a field of many strands, one strand of which has been the focus of this work. A topic for future work will be weaving these strands together. For example, how are the graph PDE-based classification methods we have considered connected to the Laplace learning methods of e.g. [2, 8]? And zooming out even further, graph-based learning is itself but one strand of the ever-growing tapestry of machine learning. This area is a meeting point of a great many threads. We have already seen the highly profitable exchange of ideas between the graph and continuous settings, and this exchange will only become richer as the graph PDEs and the graph continuum limits strands are further woven together. Furthermore, in this work we have only briefly touched on how to build the graph itself: the question of how best to do this for a given application is a topic which has lately begun to incorporate ideas from deep learning (see e.g. de Vriendt *et al.* [7]). In summary, there is a great deal left for one to explore in the field of graph-based learning, whether one is a bird or a frog.

Bibliography

[1] Andrea Bertozzi and Arjuna Flenner. "Diffuse Interface Models on Graphs for Classification of High Dimensional Data". In: *Multiscale Modeling Simulation* 10 (July 2012), pp. 1090–1118. doi: 10.1137/11083109X.

[2] J. Calder et al. "Poisson Learning: Graph Based Semi-Supervised Learning At Very Low Label Rates". English. In: *Proceedings of the International Conference on Machine Learning*. 2020, pp. 8588–8598.

[3] Veronica Corona et al. "Enhancing joint reconstruction and segmentation with non-convex Bregman iteration". In: *Inverse Problems* 35.5 (2019), p. 055001. doi: 10.1088/1361-6420/ab0b77.

[4] Abdallah El Chakik, Abderrahim Elmoataz, and Xavier Desquesnes. "Mean curvature Flows on Graphs for Image and Manifold Restoration and Enhancement". In: *Signal Processing* 105 (Dec. 2014), pp. 449–463. doi: 10.1016/j.sigpro.2014.04.029.

[5] Y. van Gennip et al. "Mean Curvature, Threshold Dynamics, and Phase Field Theory on Finite Graphs". In: *Milan Journal of Mathematics* 82 (2014), pp. 3–65.

[6] Ekaterina Merkurjev, Tijana Kostić, and Andrea Bertozzi. "An MBO Scheme on Graphs for Classification and Image Processing". In: *SIAM Journal on Imaging Sciences* 6 (Oct. 2013), pp. 1903–1910. doi: 10.1137/120886935.

[7] Marianne de Vriendt, Philip Sellars, and Angelica I Aviles-Rivero. "The GraphNet Zoo: An All-in-One Graph Based Deep Semi-supervised Framework for Medical Image Classification". In: *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Graphs in Biomedical Image Analysis*. Springer, 2020, pp. 187–197.

[8] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions". In: *IN ICML*. 2003, pp. 912–919.

Acknowledgements

First of all, I would like to thank Johan Dubbeldam and especially Yves van Gennip for promoting this thesis. To Yves, I would like to thank you for first introducing me to this fascinating topic even before we'd met, for your excellent supervision, and for providing me the opportunity to do my PhD in this beautiful city. You were able to somehow both provide me with constant support, and yet also provide me with the space to explore my own path. To Johan, though our interactions have been more sparse, you have been most helpful in navigating the details of this last stage at TU Delft.

I thank Jonas Latz and Simone Parisotto for their contributions to chapters 5 and 6 of this thesis, especially to the code without which the numerical examples would not have be possible.

I thank Carola Schönlieb, for her support during my PhD applications, her ongoing support of my academic endeavours, and for introducing me to the fascinating technique of joint reconstruction-segmentation.

I thank Andrea Bertozzi, for co-pioneering the subsubfield which is the topic of this thesis, for various helpful contributions over the last few years, and for a very stimulating secondment at UCLA.

I thank Arieh Iserles, for some very helpful suggestions regarding the work in chapter 5 for arranging for me to be a student helper at the B(A)MC 2015, where I first encountered Yves' work on PDEs on graphs, for his help storing my belongings in between Cambridge terms, and for support that I was too small to now remember.

I thank Imre Leader, for his continuous support throughout my time in Cambridge, and especially for his support during Part III exam term, without which I might not have gotten the grades needed for my PhD. I also thank him for inviting me to the Rice Exchange Dinner, which was the best food I've ever tasted.

I thank the University of Cambridge for my undergraduate/masters education and the University of Nottingham for hosting the first year and a half of my PhD.

I thank the Cambridge Tolkien Society, for keeping me sane not only during my time in Cambridge, but also—through the magic of virtual platforms—during the COVID-19 pandemic. You are all some of my closest friends, and the end of our virtual meetings will be the one product of the pandemic which I shall miss.

Finally, I give the warmest thanks to my family, especially the family dog Monty (and to a lesser extent, the family dog Josh—he knows what he did), for all their support in ways far too long to list.

It is traditional for an acknowledgements section to end with the final acknowledgement. However, given the uniqueness of these times I feel it is necessary to hand out a special *anti-acknowledgement*, which I award to SARS-CoV-2, for (as of the time of writing) 4.5 million reasons.

Curriculum Vitæ

Jeremy Michael Budd

Basic Information

27-11-1994 Born in Bristol, United Kingdom.

Education

- Technische Universiteit Delft (2019-present)
- University of Nottingham (2017-2019)
- Trinity College, University of Cambridge (2013-2017)

- **PhD** Started at the University of Nottingham and continued at the Technische Universiteit Delft, under the supervision of Dr. Yves van Gennip and promotion of Dr. Johan Dubbeldam.
- **Degree** MMath degree from the University of Cambridge, graduated with Honours with Merit.

Experience

Presentations

June 2021: Gave an invited talk at the 8ECM in Portorož, Slovenia.

April 2021: Gave a talk for the Stochastics research seminar at the Technische Universität Bergakademie Freiberg.

April 2021: Gave a contributed talk at the B(A)MC2021 at the University of Glasgow.

January 2021: Gave a talk for the Oberseminar Analysis at the Universität Bonn.

August 2020: Was scheduled to give an invited talk at the IFIP TC7 Conference at the Escuela Politecnica Nacional, Quito, however this was postponed to August 2021 due to COVID-19.

May 2020: Gave a talk for the Numerical Analysis seminar at the University of Bath.

February 2020: Gave a talk for the Applied Mathematics seminar at Utrecht University.

January 2020: Gave a talk for the Image Analysis seminar at the University of Cambridge.

August 2019: Gave an invited talk at the ICIAM2019 in Valencia.

June 2019: Gave a talk for the PDE and Applications seminar at the Technische Universiteit Delft.

May 2019: Presented a poster at the International Workshop on Nonlocal Models, PDEs and Applications at the University of Caen Basse-Normandie.

April 2019: Gave a contributed talk at the BAMC2019 at the University of Bath.

April 2018: Presented a poster at the BAMC2018 at the University of St Andrews.

Industrial workshops

January 2020: Participated in the Studiegroep Wiskunde met de Industrie (SWI) at Fontys University, Tilburg, working with a team of mathematicians on a problem of designing a better ratings system for the dutch tennis organisation KNLTB.

April 2019: Participated in the European Study Group with Industry (ESGI) at the University of Cambridge, working with a team of mathematicians on a problem of predicting the impact of small-scale power generation on the UK National Grid.

January 2019: Participated in the SWI at Wageningen University, working with a team of mathematicians on a problem of optimal traffic flow on a network of intersections.

August 2017: Participated in the Industrial Problem Solving Workshop (IPSW) at the Université de Montréal, working with a team of mathematicians on a problem of retinal image registration.

August 2016: Participated in the Mathematical Modelling in Industry Workshop (MMIW) at the PIMS institute at the University of British Columbia, working with a team of mathematicians on a problem of optimising the safe seating capacity in a concert hall.

August 2014: Participated in the IPSW at the Fields Institute, Toronto, working with a team of mathematicians on a problem regarding a novel method for mass spectrometry.

August 2013: Participated in the ESGI at the University of Southern Denmark, working with a team of mathematicians on a problem of forecasting/timetabling for R & D for a pharmaceutical company.

Teaching

November 2020–January 2021: Assisted with the virtual lectures for an Analysis course and an ODEs course.

2017–2020: Performed teaching assistance for a variety of university maths courses, both for mathematics and non-mathematics students. My duties included running problems classes, assisting in marking, and assisting in online vivas.

September 2015: Provided STEP tutoring for an Oxbridge applicant. They achieved a 1, the highest grade that is required of a student.

Other

March 2021: Organised a minisymposium on "Theory and Applications of Graph-Based Learning" for the SIAM CSE21.

December 2020: Was invited to review a paper for the *Journal of Nonlinear Science*.

November 2020: Became a reviewer for *Mathematical Reviews*/MathSciNet.

July–September 2018: Did a two month research secondment at UCLA, assisting in their REU. I worked with Dr. Andrea Bertozzi on a DARPA funded project on the detection of certain structures within large graphs, for which I developed and presented on a promising new technique.

April 2015: Was a student volunteer at the B(A)MC2015 at the University of Cambridge. I assisted in the registration process and attended multiple symposia and plenary lectures as technical support.

List of Publications

3. J. Budd and Y. van Gennip, *Mass-conserving diffusion-based dynamics on graphs*, European Journal of Applied Mathematics (2021), pp. 1–49. DOI: 10.1017/S0956792521000061.

2. J. Budd, Y. van Gennip, and J. Latz, *Classification and image processing with a semi-discrete scheme for fidelity forced Allen–Cahn on graphs*, GAMM Mitteilungen **44** (2021), no. 1, pp. 1–43. DOI: 10.1002/gamm.202100004.

1. 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), pp. 4101–4139. DOI: 10.1137/19M1277394.