Optimal Control of the Advection-Diffusion Equation for Effective Fluid Mixing

by

Christopher J. Miles

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Physics)
in the University of Michigan
2018

Doctoral Committee:

Professor Charles Doering, Chair
Professor Anthony Bloch
Assistant Professor Jesse Capecelatro
Associate Professor Robert Deegan
Professor Mark Newman

Christopher J. Miles

cmiless@umich.edu

ORCID iD: 0000-0003-3813-5050

DEDICATION

To my parents, teachers, and mentors who have provided encouragement throughout my life.

ACKNOWLEDGEMENTS

I am greatly indebted to the mentors, instructors, departmental staff, friends, and family members who have supported me throughout my Ph.D. I am especially thankful for the mentorship of my advisor Charles Doering who provided guidance throughout my studies. He has been a key element to my scientific and academic development. I would also like to gratefully acknowledge my committee members: Anthony Bloch, Jesse Capecelatro, Robert Deegan, and Mark Newman. Thank you for taking the time and effort to evaluate this work.

During my time at Michigan, I was fortunate to have many fruitful research discussions that impacted this work, especially those with Ian Tobasco, Karen Zaya, and Andre Souza. I also appreciate the helpful comments and feedback provided (some indirectly) on this work by Hongjie Dong, Luis Escauriaza, Guatum Iyer, Alexander Kiselev, Anna Mazzucato, and Christian Seis.

Last but not least, I would like to sincerely thank Irene Park for always being supportive and caring throughout my graduate school years. She was always there for me and I am forever thankful.

TABLE OF CONTENTS

LIST OF FIGURES

<u>**Figure**</u>

LIST OF APPENDICES

Appendix

ABSTRACT

Mixing is a fundamental fluid mechanism that is crucial to the engineering of industrial processes within the chemical, pharmaceutical, petrochemical, food, and many other industries. Mixing is also important to areas of science including oceanography, turbulence, and atmospheric sciences. An important question to many domains is “How does one mix efficiently?” We strive to make progress towards this question by studying a series of optimization problems on mixing.

The first study presented is on optimization of a shell model of mixing. This model is based on a system of ordinary differential equations which mimic the time evolution of the Fourier spectrum of a dye concentration governed by the advection-diffusion equation. We investigate the local-in-time and global-in-time optimization within this model and show that mixing can be limited by diffusion.

The second study investigates local-in-time optimization of the advection-diffusion partial differential equation. We demonstrate that many of the observations seen in the shell model extend to this setting such as evidence of a limitation on mixing by the inclusion of diffusion.

Lastly, we explore global-in-time optimization of the advection-diffusion equation. This last study is ongoing research at the moment: current results on this topic are presented and a comparison between local-in-time and global-in-time optimization is discussed.

CHAPTER I

Introduction

1.1 Why study fluid mixing?

Fluid mixing happens in a gust of wind and in a cup of morning coffee with cream. This phenomena is commonplace and plays an important role in many natural and engineering systems that humanity depends on. Our current lack of fundamental understanding of mixing impedes our ability to understand natural systems such as atmospheric and oceanic processes that impact our global climate. Mixing also serves as a key industrial process crucial for production within the food, chemical, pharmaceutical, and petrochemical industries. Thoughtful design of industrial mixing is essential for maximizing product yield and product quality throughout these industries. In addition, poor mixing design can come at a cost. In 1989, the cost of poor mixing was estimated to be \$1 – \$10 billion US dollars in the chemical industry alone [36]. Nearly everyone depends on these industries for basic household products, health needs, travel, and food. And in most situations, the cost in production is inevitably paid by consumers — that includes you and me.

Although mixing is highly prevalent and often utilized, its fundamental princi-

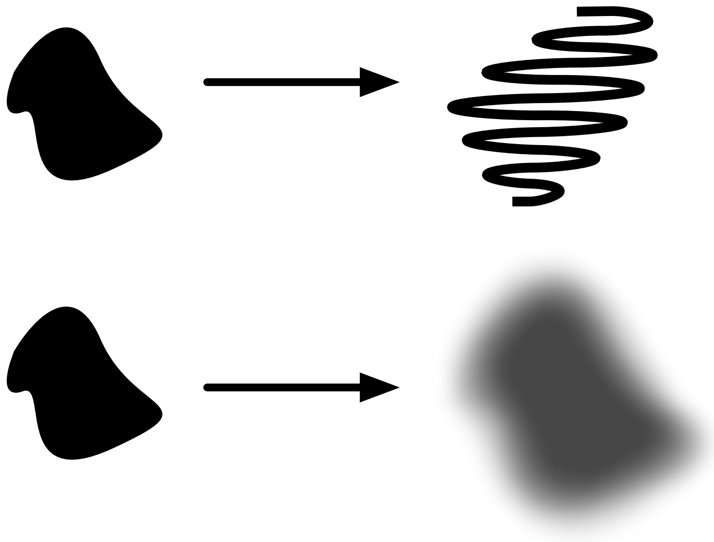

Figure 1.1: Top transformation is a reduction in the scale of segregation. Bottom transformation is a reduction in the intensity of segregation

ples are still not fully known especially concerning how the interplay of advection and diffusion processes affect mixing rates and achievable filamentation length scales. To make progress on understanding the systems that involve mixing, we must understand mixing itself. Thus, the approach taken here is to study a theoretical and mathematical framework of mixing that has been stripped down to its essential elements. The perspective taken in this work is that one must understand the idealized problems first before tackling problems with added complexity. An idealized Carnot engine provides efficiency expectations of real-world heat engines each unique with its own complexity. By analogy, we hope the idealized mixer presented here will provide theoretical principles on mixing efficiency about real-world mixers as well.

1.2 What does well-mixed mean?

In the pioneering work of P. V. Danckwerts (1952) [11], the author identifies two indicators of mixed-ness: the scale of segregation and the intensity of segregation. The *scale of segregation* is the characteristic length scale present in the concentration. For instance, the process of thinning, elongating, and folding of a blob, as seen in the top graphic of figure 1.1, reduces the scale of segregation by creating a rich maze-like pattern with thin strands of dye. The *intensity of segregation* refers to the variation of the concentration amplitude. This is naturally done by diffusion. The bottom graphic of figure 1.1 shows a reduction in the overall variation and the concentration tends towards a state with a uniform concentration.

1.3 Fluid mixing stages and mechanisms

The seminal work of C. Eckart [14] describes the three stages of mixing:

1. *The initial stage*: The mixedness of the initial state will dictate the amount of 'work' necessary to mix to a desired state. Typically, there will be large volumes of dye visible throughout the fluid. Thus the preparation of the initial concentration is a stage in its own right before the fluid mechanisms are called into action.

2. *The intermediate stage*: *Advection*, or colloquially the act of stirring, will distort, stretch, and fold the volumes of dye to produce large gradients throughout the fluid and reduce the scale of segregation.

3. *The final stage*: Lastly the gradients will disappear under the action of *diffusion*. Diffusion refers to molecular diffusion throughout this work and

not to be confused with its occasional usage of dispersal (to spread a dye thoughout space).

The intermediate and final stages are shown to happen sequentially in time. This is not completely accurate. These processes occur concurrently as we will investigate throughout this work.

In the stages above, we have introduced the two main mechanisms at play — advection and diffusion. Advection is a fluid mixing mechanism that when used appropriately can be an excellent way to reduce the scale of segregation.

1.4 A mathematical framework for studying mixing

The effect of advection and diffusion on the rate of fluid mixing depends on the particular mixing situation characterized by the unique fluid properties, specific mixing flow, and boundary geometry. In view of the vast complexity of the mixing situation, general principles of mixing underlying these various situations would be beneficial. In particular, it is valuable to determine how the mixing rate (typically the most optimal mixing rate) depends on aggregate flow intensity measures such as the stirring flows' energy and/or enstrophy. This is the objective of the research program encompassing many efforts [40, 24, 43, 28, 43, 18, 32, 49, 30, 8, 34, 33] in last decade.

With these goals in mind, a common approach taken throughout the literature is to consider the evolution of a tracer quantity θ advected by an incompressible ($\nabla \cdot \mathbf{u} = 0$) flow $\mathbf{u}$ with mild physical constraints within a periodic box D of side length L in d dimensions. All numerical simulations are done in 2 dimensions while analytical results are generally presented in arbitrary dimension. We will assume that θ has zero mean throughout this work. The tracer concentration field

θ evolves according to the advection-diffusion equation,

$$\partial_t \theta + \mathbf{u} \cdot \nabla \theta = \kappa \Delta \theta, \tag{1.1}$$

with initial data $\theta(\mathbf{x}, 0) = \theta_0(\mathbf{x})$, where κ is the molecular diffusion coefficient and $\Delta = \nabla^2$ is the Laplacian operator. The flow intensity is constrained by enstrophy

$$\|\nabla \mathbf{u}\|_{L^2} = \sqrt{\int\limits_D |\nabla \mathbf{u}|^2 \, d^d\mathbf{x}} = \Gamma L^{d/2} \tag{1.2}$$

or energy

$$\|\mathbf{u}\|_{L^2} = \sqrt{\int\limits_D |\mathbf{u}|^2 \, d^d\mathbf{x}} = U L^{d/2} \tag{1.3}$$

where Γ is the root mean square rate-of-strain and U is the root mean square speed. We will also consider the time-average versions $\frac{1}{T}\int_0^T \|\nabla \mathbf{u}\|_{L^2}^2 dt = \Gamma L^{d/2}$ or $\frac{1}{T}\int_0^T \|\mathbf{u}\|_{L^2}^2 dt = U L^{d/2}$.

The form above shows that the evolution of the tracer concentration θ is slaved to a given flow $\mathbf{u}$ which embodies, in many cases, most of the complexity of a particular mixing problem. The details of $\mathbf{u}$ alone can be complicated since in natural settings $\mathbf{u}$ is a solution of the Navier-Stokes equations. Although flow situations can be vastly different, they can still share commonalities such as incompressibility and similar amount of total energy or enstrophy. Note that enstrophy is proportional to the dissipation power for Newtonian fluids. We will only consider flows constrained by these properties for the purposes of simplicity and generality.

The negative Sobelov norms H^{-n} [32, 31, 43, 28] are measures of mixing used throughout the literature and the H^{-1} norm (and sometimes referred to as the mix-norm) will be used here. The H^{-n} norm for mean-zero scalar fields θ are given by

$$\|\theta\|_{H^{-n}} = \|\nabla^{-n}\theta\|_{L^2} = \sqrt{\int\limits_D |\nabla^{-n}\theta(\mathbf{x},t)|^2\, d^d\mathbf{x}} = \sqrt{\sum_{\mathbf{k}\neq\mathbf{0}} L^d \frac{|\hat{\theta}_{\mathbf{k}}(t)|^2}{|\mathbf{k}|^{2n}}} \tag{1.4}$$

where $\nabla^{-1} = \nabla\Delta^{-1}$, the operator Δ^{-1} acting on a function ρ returns the solution ϕ of the Poisson equation $\Delta\phi = \rho$, and $\hat{\theta}_{\mathbf{k}}(t) = \frac{1}{L^d}\int_D \theta(\mathbf{x},t)e^{-i\mathbf{k}\cdot\mathbf{x}}\, d^d\mathbf{x}$. Lower values of the H^{-1} norm correspond to a more mixed state. Note that H^{-1} norm can decrease in two ways. The first way is to decreasing the amplitudes of $|\hat{\theta}_{\mathbf{k}}|$ for $\mathbf{k} \neq 0$. This matches our first sense of mixing — *homogenization*. The second way is by transferring spectral mass from the lower wave numbers to the higher wave numbers to take advantage of the $1/|\mathbf{k}|^2$ weighting of amplitudes at different length scales. This produces a scalar field with sharp gradients and small length scales which matches our second sense of mixing — *filamentation*. Thus we can see that the H^{-1} norm embodies both senses.

The L^2 norm $\|\theta\|_{L^2}$ defined by

$$\|\theta\|_{L^2} = \sqrt{\int\limits_D |\theta(\mathbf{x},t)|^2\, d^d\mathbf{x}} = \sqrt{\sum_{\mathbf{k}} L^d |\hat{\theta}_{\mathbf{k}}(t)|^2} \tag{1.5}$$

and the H^1 norm $\|\nabla\theta\|_{L^2}$ defined by

$$\|\theta\|_{H^1} = \|\nabla\theta\|_{L^2} = \sqrt{\int\limits_D |\nabla^{-1}\theta(\mathbf{x},t)|^2\, d^d\mathbf{x}} = \sqrt{\sum_{\mathbf{k}} L^d |\mathbf{k}|^2 |\hat{\theta}_{\mathbf{k}}(t)|^2} \tag{1.6}$$

are also common measures of mixing and will be considered as well. For those interested in other measures of mixing, see [43].

1.5 Pure diffusive mixing

In the case without advection ($\mathbf{u} = \mathbf{0}$), equation (1.1) reduces to the classical heat equation [15]. The Fourier modes evolve according to $\hat{\theta}_{\mathbf{k}}(t) = \hat{\theta}_{\mathbf{k}}(0)e^{-\kappa|\mathbf{k}|^2 t}$. Thus we have explicit analytical results for the decay of the H^{-1} norm by simply substituting this result. Note that H^{-1} norm will surely decay monotonically since the amplitude of each mode does. Diffusion is unable to transfer spectral mass from the low wave number modes to the high wave number modes and thus is incapable of filamentation. Thus the pure diffusion case solely exploits homogenization. Also notice the unequal weighting attach to each mode. The Fourier modes with large wave number $|\mathbf{k}|$ decay at a much faster rate relative to the decay of those with small wave number.

1.6 Pure advective mixing

In the case without diffusion ($\kappa = 0$), pure advection of the flow is the only method of mixing, colloquially known as stirring. For a flow that is constrained by enstrophy, the mix-norm decays at most exponentially where the exponential rate is proportional to Γ [24, 40]. This was mathematically proven by two separate approaches: G. Iyer *et. al.* [24] used regularization results of partial differential equations [9] while C. Seis [40] used methods from optimal transportation theory [46]. Furthermore, enstrophy-constrained flows that realize this exponential decay rate have been constructed analytically [1]. On the other hand, energy-constrained flows can achieve even faster mixing rates. In fact they can achieve *perfect mixing in finite time* which means that the H^{-1} norm reaches zero in finite time as opposed to approaching zero in infinite time as exhibited in the case for enstrophy-constrained mixing. This can be demonstrated by a ‘checkerboard’ flow [30] where the mix-norm achieves perfect mixing in finite time via linear decay. For either flow intensity constraint, note that H^{-1} norm decreases by exclusively exploiting filamentation without homogenization. This is exactly opposite to the purely diffusive case.

Many works [10, 29, 31, 8, 18, 28, 30, 16, 3, 22, 45] have framed mixing enhancement in terms of optimization and optimal control theory. Mathew *et al.* [31] studied pure advection of a concentration field by a velocity field $\mathbf{u} = \sum_{i=1}^{N} \alpha_i \mathbf{u}_i$ where $\{\mathbf{u}_i\}$ is a finite set of divergence-free velocity fields and $\{\alpha_i\}$ is a set of time-dependent weights. The weights $\{\alpha_i\}$ were chosen to minimize the final-time $H^{-1/2}$ mix-norm subject to a fixed value of action or equivalently a fixed value of time-averaged energy. Necessary conditions for optimality were numerically solved by conjugate gradient. The authors considered two examples each using $\mathbf{u}_1$ and $\mathbf{u}_2$ as given cellular flow velocity fields. In both examples, they found that the $H^{-1/2}$ norm of the computed concentration field decayed at an exponential rate. Furthermore, the authors demonstrated that the kinetic energy must be conserved at all moments in time due to optimality conditions even though they only required that the *time-averaged* energy be fixed — analogous results hold true in this work as well. For this particular choice of velocity fields, the enstrophy turns out to also be conserved. This is consistent with other theoretical and numerical works [40, 24, 28, 1] reporting exponential decay rates under fixed enstrophy.

Cortelezzi *et al.* [8] also considered controlling two given flows to enhance mixing. But rather than considering a superposition of two flows, the authors considered switching entirely between one flow and the other. In particular, the authors considered controls that picked one of two sine flows $\mathbf{u}_1 = \sin(2\pi y)\hat{x}$ and $\mathbf{u}_2 = \sin(2\pi x)\hat{y}$ at uniformly spaced switching times. The authors divided the optimization task into multiple optimization sub-problems performed over short time horizons covering the entire time interval. They found, in the presence and absence of diffusion, that the mixing efficiency, as measured by the $H^{-1/2}$ mix-norm, of the short-horizon optimization schemes was substantially better than the periodic control that alternates between the two flows at each switching time. They also concluded that mixing can be greatly enhanced when optimizing over very short time horizons.

Lin *et al.* [28] explored short-time considerations even further. They found an analytic expression for the instantaneous optimal choice of velocity field given the current concentration field under fixed energy and enstrophy constraints. This was done by minimizing the time derivative of the H^{-1} norm at each instant. Using the resulting expression, they numerically integrated the advection equation forward in time while determining the optimal velocity field at each time step. For an enstrophy-constrained flow, they numerically demonstrated exponential decay of the $H^{-1/2}$ and H^{-1} norms consistent with [40, 24, 1, 31]. Lunasin *et al.* [30] also performed a similar analysis as Lin *et al.* for flows with fixed palenstrophy ($\|\Delta \mathbf{u}\|_{L^2}$). This form of optimization is referred to as local-in-time optimization and will be discussed further shortly.

1.7 The interplay of advection and diffusion

Finally, the case with diffusion and advection is the least explored in this framework and the focus of this thesis. It is known that the evolution of the H^{-1} and L^2 norms decrease monotonically under the checkerboard flow introduced by [30] while the H^1 increases until it reaches a peak and then decreases [18]. This peak corresponds to a time when the length scales developed are small enough for diffusion to effectively act on steep gradients. In contrast to the 'pure' cases mentioned earlier, it is important to note that the H^{-1} can now decrease by the two avenues of homogenization and filamentation simultaneously.

At this point, we can already see a glimpse of a conflict between diffusion and advection for the ultimate goal of optimal mixing. Pure advection succeeds at filamentation by transferring spectral mass from the low wave number modes to the high wave number modes in a continuous fashion. However in the presence of diffusion, a once optimal pure advection flow exceptional at filamentation will be met with potential conflict since homogenization by diffusion can stifle its progress in transferring spectral mass to high wave number modes. Given that diffusion is ubiquitous, we must come to terms with this conflict to produce efficient mixing.

1.8 The question and goals

In this work, the interplay of advection and diffusion is explored to determine its impact on the rate of mixing. As we have mentioned in the last section, there appears to be a conflict between advection and diffusion that arises when both are acting to reduce homogenization through reduction in scale and intensity of segregation simultaneously. The main question underlying this entire thesis work is:

> *What is the optimal mixing rate achievable under the enstrophy and energy constrained flows when both advection and diffusion are active?*

We hope to make progress in answering this by posing the question as an optimization problem. We will consider the *local-in-time optimization* problem:

$$\min_{\mathbf{u}} \frac{d}{dt} \|\nabla^{-1}\theta(\cdot, t)\|_{L^2}^2 \tag{1.7}$$

where the flow intensity is constrained by enstrophy

$$\|\nabla \mathbf{u}\|_{L^2}^2 = \Gamma^2 L^d \tag{1.8}$$

or energy

$$\|\mathbf{u}\|_{L^2}^2 = U^2 L^d. \tag{1.9}$$

We also consider the *global-in-time optimization* problem:

$$\min_{\mathbf{u}} \|\nabla^{-1}\theta(\cdot, T)\|_{L^2}^2 \tag{1.10}$$

where the flow intensity is constrained by *time-averaged* enstrophy

$$\frac{1}{T}\int_0^T \|\nabla \mathbf{u}\|_{L^2}^2 = \Gamma^2 L^d \tag{1.11}$$

or time-averaged energy

$$\frac{1}{T}\int_0^T \|\mathbf{u}\|_{L^2}^2 = U^2 L^d. \tag{1.12}$$

For all formulations, the flow is always required to be divergence-free ($\nabla\cdot\mathbf{u} = 0$) and θ solves the advection-diffusion equation with initial data $\theta(\mathbf{x}, 0) = \theta_0(\mathbf{x})$.

1.9 Organization of dissertation

The rest of this dissertation is organized as follows: Chapter II describes the local- and global-in-time optimization within the context of a shell model, a model representing the spectral dynamics of the advection-diffusion equation. Here we find the first indication that diffusion in some cases can penalize mixing performance. This work was published in the Journal of Nonlinear Science in 2017 [33].

Chapter III studies local-in-time optimization in the context of the advection-diffusion partial differential equation. Here we investigate further the impact of diffusion. We find that diffusion can in some cases negatively impact the long-term mixing rate for local-in-time optimal flows. This work has been accepted for publication in Nonlinearity [34]. Chapter IV presents on-going work on global-in-time optimization of the advection-diffusion equation. An analytical result is presented showing that it is optimal to expend the stirring budget uniformly in time for the pure advection case. This result appears in an Appendix section of the 2017 Journal of Nonlinear Science article [33].

CHAPTER II

Shell model [1]

2.1 Introduction

Shell models are systems of ordinary differential equations that mimic the mathematical structure of the spectral representation of a partial differential equations [13]. They were introduced in the context of the Navier-Stokes equations to study turbulent cascade dynamics [25, 21, 48] while avoiding mathematical difficulties inherent in the full nonlinear partial differential equation. We provide a similar treatment of the advection and advection-diffusion equations in the context of transient mixing and use optimization techniques to study shell-model stirring strategies to optimally mix a tracer concentration.

The shell model is designed to show qualitative features of advection, notably conservation of the tracer density variance, i.e., the L^2 — or more precisely ℓ^2 — norm of the tracer concentration. Mixing is quantified by a negative Sobolev norm, the H^{-1} norm, for the tracer concentration, with a natural extension to the shell model, that can measure tracer dispersion even in the absence of diffusion. The shell model also displays quantitative correspondence with results for maximal

[1] The content of this chapter has been published in the Journal of Nonlinear Science (2017) [33].

mixing in the partial differential equation formulation including perfect (complete) mixing in finite time for sufficiently weak constraints on the stirring flow field and exponential decay of the mix-norm for other protocols. We extend the optimal stirring analysis to make predictions for the influence of diffusion on mix-norm decay.

This chapter is organized as follows. In Section 2.2 we introduce the shell model. Local- and global-in-time optimization schemes are described in, respectively, Sections 2.3 and 2.4. The two shell model stirring strategies are compared without diffusion in Section 2.5 and with diffusion in Section 2.6. The concluding section 2.7 contains a discussion of the results. Appendix A contains exact analytical results for a three-shell truncation and derivations of lower bounds on the mix-norm.

2.2 A shell model

Shell models are coarse-grain versions of the spectral representation of a partial differential equation. In particular the Fourier transform of the advection-diffusion equation (1.1) becomes the infinite set of coupled ordinary differential equations

$$\partial_t \hat{\theta}(\mathbf{k},t) + i \sum_{j=1,2,3} \sum_{\mathbf{k}' \in K} \hat{u}_j(\mathbf{k}-\mathbf{k}',t)\, k'_j\, \hat{\theta}(\mathbf{k}',t) + \kappa\, |\mathbf{k}|^2 \hat{\theta}(\mathbf{k},t) = 0.$$

We course-grain this relation by ‘binning’ the Fourier variables $\hat{\theta}(\mathbf{k},t)$ with wavenumbers $2^{n-1}k_0 < |\mathbf{k}| < 2^n k_0$ ($k_0 = \frac{1}{2L}$) into a single variable $\theta_n(t)$ for $n = 1, 2, \ldots, \infty$. This binning process divides $\mathbf{k}$-space into concentric shells and hence the name—shell model. We similarly course-grain the Fourier amplitudes of the flow field $\hat{u}_i(\mathbf{k},t)$ into the variables $u_n(t)$ and look for the simplest shell model that retains mode-coupling between neighboring shells. Thus, we choose the following form:

$$\frac{d}{dt}\theta_n = k_{n-1}u_{n-1}\theta_{n-1} - k_n u_n \theta_{n+1} - \kappa\, k_n^2 \theta_n, \quad n = 1, 2, \dots, \tag{2.1}$$

where $k_n = k_0 2^n$ and $\theta_0 \equiv 0 \equiv u_0$. κ has units of L^2/T; k_n has units of $1/L$; u_n has units of L/T; and θ_n is unitless. We may also consider N-shell truncated models with $n = 1, 2, \dots, N$ where $\theta_{n \geq N+1} = 0 = u_{n \geq N}$. See references [25, 47, 13] for alternative shell models of advection-diffusion.

This construction is not intended to be mathematically rigorous. Rather, it is meant to mimic the natural cascade of the spectrum of the tracer, progressively visiting each shell as stirring stimulates smaller length scales. Note that this model preserves the relation $\frac{d}{dt}\|\theta\|_{L^2}^2 = -2\kappa\|\nabla\theta\|_{L^2}^2$ (found by multiplying the advection-diffusion equation by θ and integrating over D), but now in an l^2 sense:

$$\frac{d}{dt}\sum_{n=1}\theta_n^2 = -2\kappa\sum_{n=1}k_n^2\theta_n^2. \tag{2.2}$$

It follows that the l^2 norm is conserved for the non-diffusive case.

Lastly, we define the h^α shell-model Sobolev norm as

$$\|\psi(t)\|_{h^\alpha}^2 \equiv \sum_{m=1} k_m^{2\alpha}\psi_m^2(t). \tag{2.3}$$

for any vector $\psi = (\psi_1, \psi_1, \dots)$. The shell-model h^{-1} mix-norm is defined as $\|\theta(t)\|_{h^{-1}}$. We denote the intensity constraints of the shell-model flow in terms of $\|u(t)\|_{h^\alpha}^2$. When $\alpha = 0$ this is the ℓ^2 analog of the energy and $\alpha = 1$ returns an expression for enstrophy. The norm operator $\|\cdot\|_{h^\alpha}$ has units of $L^{-\alpha}$.

2.3 Instantaneous optimization

We begin by asking, “What admissible control will produce the best instantaneous mixing rate?” The analysis shown here parallels the work done by Lin *et al* [28] in the partial differential equation setting. We formulate this question as the following: find the u that realizes

$$\min_u \frac{d}{dt}\|\theta(t)\|_{h^{-1}}^2 \tag{2.4}$$

at each time t subject to the constraint

$$\|u(t)\|_{h^\alpha} = W^{(\alpha)} \tag{2.5}$$

where $W^{(0)} = U$ (energy) and $W^{(1)} = 1/\tau$ (enstrophy). The root-mean-square rate-of-strain Γ is given by $\Gamma = 1/\tau$.

Differentiating the mix-norm and using (2.1), we find

$$\frac{d}{dt}\|\theta(t)\|_{h^{-1}}^2 = 2\sum_{n=1}\left(k_{n+1}^{-2} - k_n^{-2}\right)\theta_n\theta_{n+1}k_nu_n - \kappa\,\theta_n^2 \tag{2.6}$$

and the optimization problem is solved by the method of Lagrange multipliers. The solution is

$$u_n^{(\alpha)}(t) = -\frac{W^{(\alpha)}\gamma_n^{(\alpha)}(t)}{k_n^\alpha\|\gamma^{(\alpha)}(t)\|_{l^2}} \tag{2.7}$$

where $\gamma_n^{(\alpha)}(t) \equiv (k_{n+1}^{-2} - k_n^{-2})k_n^{1-\alpha}\,\theta_n(t)\,\theta_{n+1}(t)$ — at least, when $\|\gamma^{(\alpha)}(t)\|_{l^2} \neq 0$.

An alternative stategy is needed when $\|\gamma^{(\alpha)}(t)\|_{l^2} = 0$. An analgous situation arises in the partial differential equation setting [28] and the second derivative $\frac{d^2}{dt^2}\|\theta(t)\|_{h^{-1}}^2$ is minimized instead at these instances. We will do the same here.

We write the second derivative as

$$\frac{d^2}{dt^2}\|\theta(t)\|_{h^{-1}}^2 = u^T B u + c^T u + d \tag{2.8}$$

where

$$B_{nm} \equiv \begin{cases} 2(k_{n+1}^{-2} - k_n^{-2})k_n^2(\theta_n^2 - \theta_{n+1}^2) & m = n \\ -2(k_{n+1}^{-2} - k_n^{-2})k_n k_{n+1}\theta_n\theta_{n+2} & m = n+1 \\ 2(k_{n+1}^{-2} - k_n^{-2})k_n k_{n-1}\theta_{n-1}\theta_{n+1} & m = n-1 \\ 0 & \text{otherwise} \end{cases},$$

$$c_n = -2\kappa\,(k_{n+1}^{-2} - k_n^{-2})(k_{n+1}^2 + k_n^2)k_n\theta_n\theta_{n+1}, \qquad \text{and} \qquad d = \sum_{n=1} 4\kappa^2\, k_n^2\theta_n^2.$$

We want to find the minimum of (2.8) subject to the intensity constraint (2.5). By the method of Lagrange multipliers, the optimizer satisfies the following system of equations:

$$\left(B^T + B - \lambda K^{(\alpha)}\right)\, u = -c \tag{2.9a}$$

$$\|u\|_{h^\alpha} = W^{(\alpha)} \tag{2.9b}$$

where λ is a lagrange multiplier and $K^{(\alpha)} = \text{diag}(k_1^{2\alpha}, \ldots, k_{N-1}^{2\alpha})$. The general local-in-time strategy is to minimize the $(n+1)$th time derivative of $\|\theta(t)\|_{h^{-1}}$ if the control u does not affect the 1st through nth derivatives. We will soon return to the local-in-time strategy when applying it in sections 2.5 and 2.6.

2.4 Global-in-time optimization

Let's explore the global-in-time strategy which optimizes mixing at the end time rather than instantaneously. In this case we wish to solve the following global-in-time optimization problem: at some final time $T > 0$ find

$$\min_{u} \|\theta(T)\|^2_{h^{-1}} \tag{2.10}$$

subject to the time averaged intensity constraint $\frac{1}{T}\int_0^T \|u(t)\|^2_{h^\alpha}\, dt = [W^{(\alpha)}]^2$. Toward this end we introduce the augmented Lagrangian

$$\begin{aligned}\mathcal{L}\{\theta, u, \phi, \mu\} = \frac{1}{2}\sum_{n=1} \frac{\theta_n^2(T)}{k_n^2} + \int\limits_0^T \Bigg\{ \sum_{n=1} \phi_n \left(k_{n-1}u_{n-1}\theta_{n-1} - k_n u_n \theta_{n+1} - \kappa k_n^2 \theta_n - \dot{\theta}_n\right) \\ + \frac{\mu}{2}\left(\sum_{n=1} k_n^{2\alpha} u_n^2 - [W^{(\alpha)}]^2\right) \Bigg\}\, dt\end{aligned}$$

where for truncated shell models the first two sums above run up to $n = N$ while the third terminates at $N-1$. At extrema the first variations vanish with respect to the variables θ, ϕ, u and μ:

$$\begin{aligned}
\frac{\delta \mathcal{L}}{\delta \theta_n(T)} = 0 \Rightarrow &\qquad \frac{\theta_n(T)}{k_n^2} - \phi_n(T) = 0 && \text{(2.11a)}\\
\frac{\delta \mathcal{L}}{\delta \theta_n} = 0 \Rightarrow &\qquad \dot{\phi}_n - k_{n-1}u_{n-1}\phi_{n-1} + k_n u_n \phi_{n+1} - \kappa\, k_n^2 \phi_n = 0 && \text{(2.11b)}\\
\frac{\delta \mathcal{L}}{\delta \phi_n} = 0 \Rightarrow &\qquad \dot{\theta}_n - k_{n-1}u_{n-1}\theta_{n-1} + k_n u_n \theta_{n+1} + \kappa\, k_n^2 \theta_n = 0 && \text{(2.11c)}\\
\frac{\delta \mathcal{L}}{\delta u_n} = 0 \Rightarrow &\qquad k_n \phi_{n+1}\theta_n - k_n \phi_n \theta_{n+1} + \mu k_n^{2\alpha} u_n = 0 && \text{(2.11d)}\\
\frac{\delta \mathcal{L}}{\delta \mu} = 0 \Rightarrow &\qquad \frac{1}{T}\int\limits_0^T \|u(t)\|^2_{h^\alpha}\, dt - [W^{(\alpha)}]^2 = 0 && \text{(2.11e)}
\end{aligned}$$

Thus, (2.11) holds true for all extrema of the augmented Lagrangian and therefore gives necessary conditions for a global optimizer.

For the non-diffusive case, an explicit calculation of the time derivative of

$\|u(t)\|_{h^\alpha}$ reveals that $\|u(t)\|_{h^\alpha}$ is conserved for an optimal trajectory by making use of (2.11) as done in Appendix A.3. This is interesting, since we only demanded that the *time-average* of stirring strength be fixed and equal to $W^{(\alpha)}$. An analogous statement holds in the context of the partial differential equation as well (see Chapter IV) which is an extension of the results first demonstrated by Mathew *et al* [31]. The theory developed here will be applied to various cases in the next two sections.

2.5 Mixing without diffusion

We will first consider the local-in-time strategy starting from the most unmixed state. Then we study the three-shell truncated model which demonstrates the difference between local-in-time and global-in-time strategies. Lastly before introducing diffusion, we will show that the key features of global-in-time optimization shown in three-shell truncated model carry over naturally to models with a larger number of shells.

2.5.1 Local-in-time strategy for infinite system

Consider the enstrophy-constrained case and start the infinite system with the most unmixed possible state, $\theta(0) = (1, 0, 0 \dots)^T$. The local-in-time strategy uses each component of the control vector u sequentially and in a piecewise fashion over time. We segment time into intervals, $[t_n, t_{n+1}]$, of equal duration where $t_n = \frac{\tau(n-1)\pi}{2}$ is the time when the state vector is entirely in the nth shell ($\theta_n = 1$ and $\theta_{m\neq n} = 0$). More precisely for $t \in [t_n, t_{n+1}]$, the optimal control is $u_n = \frac{1}{\tau k_n}$ and $u_{m\neq n} = 0$ while the state vector is given by $\theta_n(t) = \cos((t - t_n)/\tau)$ and $\theta_{n+1}(t) = \sin((t - t_n)/\tau)$ and all other components of θ are identically zero. The local-in-time strategy is shown graphically in figure 2.1. We find that the mix-norm evaluated at times t_n falls off exponentially; Given that $\|\theta(t_n)\|^2_{h^{-1}} = \frac{1}{2k_n^2} = \frac{1}{2\, 2^{2n-2}}$ and using the relation $t_n = \frac{\tau(n-1)\pi}{2}$, we find that

$$\|\theta(t_n)\|_{h^{-1}} = \|\theta(0)\|_{h^{-1}} \exp(-\log(2)t_n/\pi\tau). \tag{2.12}$$

We highlight that this exponential decay agrees qualitatively with known results on the mixing rate with the enstrophy constraint [24, 40, 28, 1, 49]. In fact it can be shown definitively that the mix-norm decays no faster than exponentially. More precisely, (see Appendix A.1 for derivation)

$$\|\theta(t)\|_{h^{-1}} \geq \|\theta(0)\|_{h^{-1}} \exp\left(-\frac{3t}{2\tau}\right) \tag{2.13}$$

for *every* stirring strategy. The local-in-time strategy is illustrated in figure 2.1 and compared to this bound (2.13).

For the energy-constrained case, we again segment time into intervals $[t_n, t_{n+1}]$ which are geometrically decreasing in duration where the times t_n are defined by $t_{n+1} = t_n + \Delta t_n$, $t_1 = 0$, and $\Delta t_n = \frac{\pi}{2Uk_n}$. During each interval $[t_n, t_{n+1}]$, the control is given by $u_n = U$ and $u_{m\neq n} = 0$ while the state vector is given by $\theta_n(t) = \cos(k_n U(t-t_n))$ and $\theta_{n+1}(t) = \sin(k_n U(t-t_n))$. Therefore the solution is similar to the enstrophy-constrained case except now the intervals are shrinking at a geometric rate. Thus the mix-norm goes to zero ($\lim_{n\to\infty} \|\theta(t_n)\|_{h^{-1}} = \lim_{n\to\infty} \frac{1}{k_n} = 0$) in finite time since

$$(t_\infty \equiv) \lim_{n\to\infty} t_n = \sum_{n=1}^{\infty} \Delta\tau_n = \frac{\pi}{2k_0 U}. \tag{2.14}$$

Note that if the entire concentration starts in the mth shell ($\theta(0) = \mathbf{e}_m$), then

t_∞ becomes the partial sum $t_\infty = \sum_{n=m}^{\infty} \Delta\tau_n$. Once more we get qualitative agreement with known results from fluid mixing. Lunasin *et al* [30] showed that perfect mixing in finite time for a simple binary distribution with fixed energy is indeed possible.

We also obtain a lower bound for the energy-constrained case: (see derivation in Appendix A.2)

$$\|\theta(t)\|_{h^{-1}} \geq \|\theta(0)\|_{h^{-1}}(1 - t/t_c) \tag{2.15}$$

where $t_c = \frac{2}{3U}\frac{\|\theta(0)\|_{h^{-1}}}{\|\theta(0)\|_{l^2}}$. Figure 2.1 shows the local-in-time strategy for this case compared to the bound (2.15) shown above.

In either constraint, the state vector moves from plane to plane. The state vector first rotates in the θ_1-θ_2 plane from the θ_1 axis to the θ_2 axis and then rotates in the θ_2-θ_3 plane from the θ_2 axis to the θ_3 axis and so forth. Note that for this particular initial condition, the analysis holds for N-shell truncated models and the above strategy holds for times $t < t_N$. When $t = t_N$, the state vector has reached the final shell and it is no longer possible to decrease the mix-norm any further. Note that the local-in-time strategy behaves somewhat like a discrete analog to the self-similar strategies [1, 49] found in the continuous partial differential equation problem since the same transformation is applied sequentially at piecewise time intervals at smaller and smaller scales.

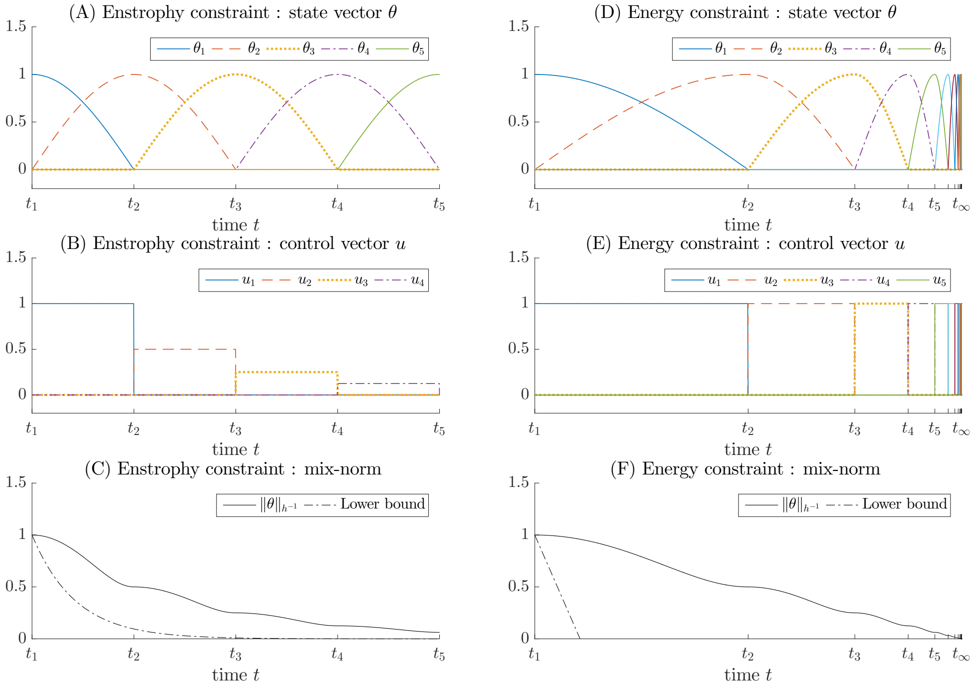

Figure 2.1: Local-in-time strategy without diffusion starting initially from the most unmixed state. The entrophy-constrained case ($\frac{1}{\tau} = 1$) is shown on the left subplots where (A) shows the state, (B) shows the control, and (C) shows the mix-norm. The energy-constrained case ($U = 1$) is shown on the right subplots where (D) shows the state, (E) shows the control, and (F) shows the mix-norm.

2.5.2 Global-in-time strategy for 3-shell truncated model with enstrophy constraint

The diffusionless 3-shell truncated model, given by

$$\begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & -k_1 u_1 & 0 \\ k_1 u_1 & 0 & -k_2 u_2 \\ 0 & k_2 u_2 & 0 \end{pmatrix}}_{\equiv A(t)} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix}, \tag{2.16}$$

is the simplest reduced model that retains many interesting features of the full infinite system. From the last section, we know that the local-in-time strategy is to move along planes one by one. This holds for the 3-shell truncated model as well.

Now, let us determine how you can improve upon the local-in-time strategy by considering the global-in-time strategy. We would like to minimize $\|\theta(T)\|_{h^{-1}}^2$ subject to the enstrophy constraint, $\frac{1}{T}\int_0^T \|u(t)\|_{h^1}^2 dt = \frac{1}{\tau^2}$. It is shown in Appendix A.4 that the solution to (2.11) for $N = 3$ has the following form:

$$k_1 u_1 = \frac{1}{\tau}\cos(\omega t) \qquad \text{and} \qquad k_2 u_2 = \frac{1}{\tau}\sin(\omega t) \tag{2.17}$$

where ω is a real number left to be determined. To help determine the solution to (2.16) given this optimal control (2.17), we decompose A as $A = k_2u_2S_x + k_1u_1S_z = \mathbf{B}\cdot\mathbf{S}$ where $\mathbf{B} = [k_2u_2, 0, k_1u_1]$ and $\mathbf{S} = [S_x, S_y, S_z]$ whose elements are a common choice of basis for $\mathbf{so}(3)$ given by

$$S_x = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \quad S_y = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \quad S_z = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Notice that these elements satisfy the following commutation relations: $[S_x, S_y] = S_z$, $[S_y, S_z] = S_x$, and $[S_z, S_x] = S_y$. Given the above reformulation, we arrive at

$$\frac{d}{dt}\theta = \mathbf{B}\cdot\mathbf{S}\,\theta \tag{2.18}$$

which is similar to the Schrödinger equation for a magnetic field-spin interaction. In this view the optimal solution behaves like a rotating magnetic field as seen in nuclear magnetic resonance. As a result this system 'maps' to a two-state spin system coupled with a driven oscillatory magnetic field [17, 41, 39]. We adapt well-known techniques [38] from this area to arrive at the solution (see Appendix A.5)

$$\theta(t) = \exp(\omega t S_y) \exp\left(-\omega t S_y + \frac{t}{\tau} S_z\right) \theta(0) \tag{2.19}$$

or rewritten as [2]

$$\theta(\omega, \tau, t) = \begin{pmatrix} \cos(\omega t)\cos(\nu t) + \frac{\omega}{\nu}\sin(\omega t)\sin(\nu t) \\ \frac{1}{\nu\tau}\sin(\nu t) \\ -\sin(\omega t)\cos(\nu t) + \frac{\omega}{\nu}\cos(\omega t)\sin(\nu t) \end{pmatrix} \tag{2.20}$$

where $\nu = \sqrt{\omega^2 + \frac{1}{\tau^2}}$. Given the end condition $\phi_n(T) = \theta_n(T)/k_n^2$ and the optimality condition (2.11d), we arrive at the system of nonlinear equations:

$$F_1(\omega, \mu; \tau, T) \equiv \mu \frac{T}{\tau}\cos(\omega T) - \left(\frac{1}{k_1^2} - \frac{1}{k_2^2}\right) \theta_1(\omega, \tau, T)\, \theta_2(\omega, \tau, T) = 0 \tag{2.21a}$$

$$F_2(\omega, \mu; \tau, T) \equiv \mu \frac{T}{\tau}\sin(\omega T) - \left(\frac{1}{k_2^2} - \frac{1}{k_3^2}\right) \theta_2(\omega, \tau, T)\, \theta_3(\omega, \tau, T) = 0. \tag{2.21b}$$

Using parameters $1/\tau = 2$ and $T = 1$, we numerically computed $\omega \approx 1.249$ from (2.21). Thus, (2.17) and (2.20) are known functions of time. With these parameters, the local-in-time and global-in-time trajectories evolve on a sphere with a radius defined by $\|\theta(0)\|_{l^2}$ in θ-state space. And, the global-in-time strategy 'takes a shortcut' past the θ_2 axis relative to the local-in-time strategy as shown in figure 2.2. This short-cutting feature generalizes to truncated shell models with larger N as shown in the next section.

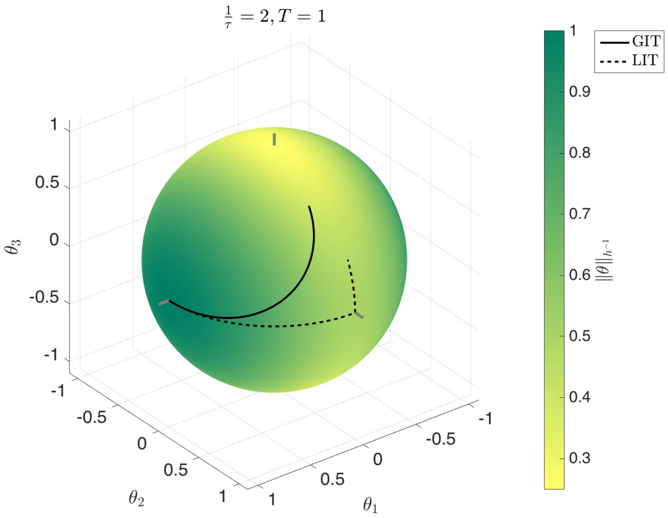

Figure 2.2: Global-in-time and local-in-time trajectories for 3-shell model with $\frac{1}{\tau} = 2$ and $T = 1$ confined to a sphere with a radius given by the conserved quantity $\|\theta\|_{l^2}$. The color indicates the degree of mixing quantified by the mix-norm $\|\theta\|_{h^{-1}}$.

When $\frac{1}{\tau} = \frac{1}{\tau^*} \equiv \frac{\sqrt{3}\pi}{2T}$, the optimal control is given by

$$k_1 u_1 = \frac{1}{\tau^*}\cos(\omega^* t) \quad k_2 u_2 = \frac{1}{\tau^*}\sin(\omega^* t). \tag{2.22}$$

where $\omega^* = \frac{\pi}{2T}$. (2.22) satisfies the budget constraint. This form is again the same as (2.17) and therefore the state vector solution is given by (2.20) with $\frac{1}{\tau} = \frac{1}{\tau^*}$ and $\omega = \omega^*$. By evaluating (2.20) at $t = T$, we find that $\theta(T) = (0, 0, 1)^T$ which is the most mixed state. Therefore the proposed control (2.22) is a *global* optimum. The parameter regime with $\frac{1}{\tau} > \frac{1}{\tau^*}$ is not of interest since this corresponds to having excess budget. To handle this situation, introduce an inequality rather than equality in our budget constraint. If this change is made, (2.22) would be the optimal solution for all values $\frac{1}{\tau} > \frac{1}{\tau^*}$.

2.5.3 Global-in-time strategy for N-shell truncated models

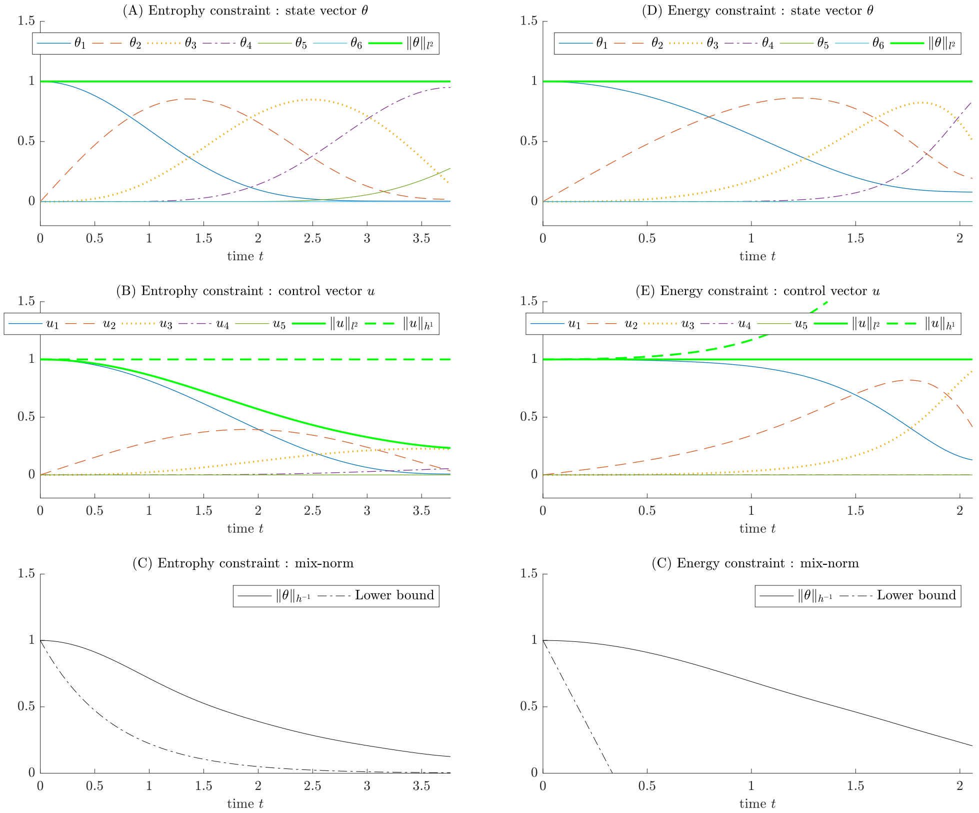

Figure 2.3: Global-in-time strategy applied to the 6-shell truncated model starting initially from the most unmixed state. The entrophy-constrained case ($\frac{1}{\tau} = 1, T = 3.77$) is shown on the left subplots where (A) shows the state, (B) shows the control, and (C) shows the mix-norm. The energy-constrained case ($U = 1, T = 2.06$) is shown on the right subplots where (D) shows the state, (E) shows the control, and (F) shows the mix-norm.

It is uncertain if an analytic solution can be found for arbitrary shell truncation number N and constraint type (enstrophy or energy). However, it is possible to solve the general case numerically by using a gradient-based method (classical gradient descent adapted to converge to a saddle point rather than a minimum).

The algorithm begins with an initial guess for $u(t) = u^{(0)}(t)$ and $\mu = \mu^{(0)}$. Given u, we numerically integrate (2.11c) forward in time to determine $\theta(t)$. We then use the terminal condition $\phi_n(T) = \frac{\theta_n(T)}{k_n^2}$ to provide an 'initial' condition for ϕ when evolving (2.11b) backwards in time. We relax the optimality condition (2.11d) and budget constraint (2.11e). Our update rules for $u^{(k)}$ and $\mu^{(k)}$ are given by

$$u_n^{(k+1)} = u_n^{(k)} - \nu_u \frac{\delta \mathcal{L}}{\delta u_n} \tag{2.23}$$

$$\mu^{(k+1)} = \mu^{(k)} + \nu_\mu \frac{\delta \mathcal{L}}{\delta \mu_n} \tag{2.24}$$

where

$$\frac{\delta \mathcal{L}}{\delta u_n} = k_n \phi_{n+1} \theta_n - k_n \phi_n \theta_{n+1} + \mu k_n^{2\alpha} u_n \tag{2.25}$$

$$\frac{\delta \mathcal{L}}{\delta \mu} = \frac{1}{T} \int\limits_0^T \|u\|_{h^\alpha} dt - [W^{(\alpha)}]^2. \tag{2.26}$$

Our convergence criteria is given by

$$\left\| \frac{\delta \mathcal{L}}{\delta u_n} \right\|_{l^\infty} < \delta \text{ and } \left| \frac{\delta \mathcal{L}}{\delta \mu} \right| < \delta$$

where both inequalities above must be true to deem convergence with tolerance δ.

Using this method, we computed the optimal solution for the 6-shell truncated model. We choose $\frac{1}{\tau} = 1$ for the enstrophy-constrained case and $U = 1$ for the energy-constrained case. Both cases are shown in figure 2.3 – this should be compared with the local-in-time strategy for both constraint types. The rate of mixing with the global-in-time strategy shows improvement over local-in-time strategy in both the enstrophy and energy constrained cases. As first seen in 3-shell truncated model, we again see the ‘short-cutting’ feature where the state vector never visits a θ_n axis as seen in the local-in-time case. Although we only show the case for $N = 6$ as an example, this feature was observed for larger values of N. Figure 2.3 also shows how the mix-norm over time compares to the lower bounds derived in Appendices A.1 and A.2.

2.6 Mixing with diffusion

In this section, we will see how diffusion affects the dynamics. One key characteristic is that the quantity $\|\theta\|_{l^2}$ is no longer conserved as shown clearly by the relation (2.2) with positive κ.

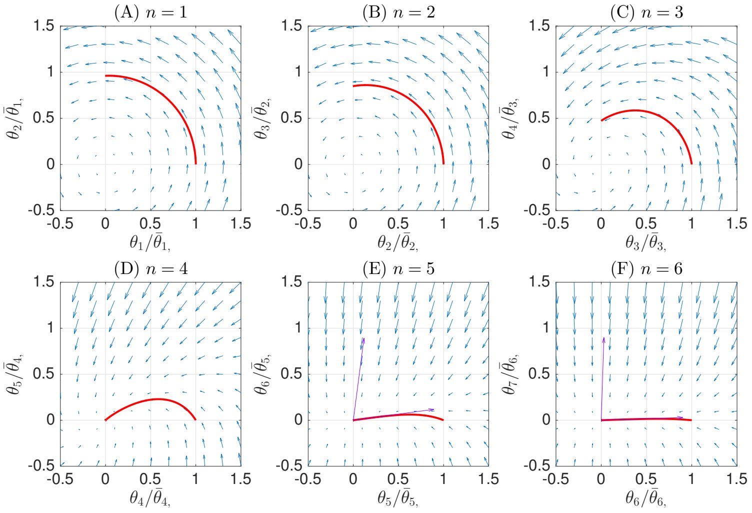

Figure 2.4: Local-in-time optimal strategy with diffusion ($\kappa = 0.01$) and fixed enstrophy ($1/\tau = 1$). The state trajectory is indicated in red and the normalized eigenvectors are purple. The fixed point at the origin is a stable spiral for low values of n and becomes a stable node when n is greater than 4.

We again consider the same initial condition and local-in-time optimization

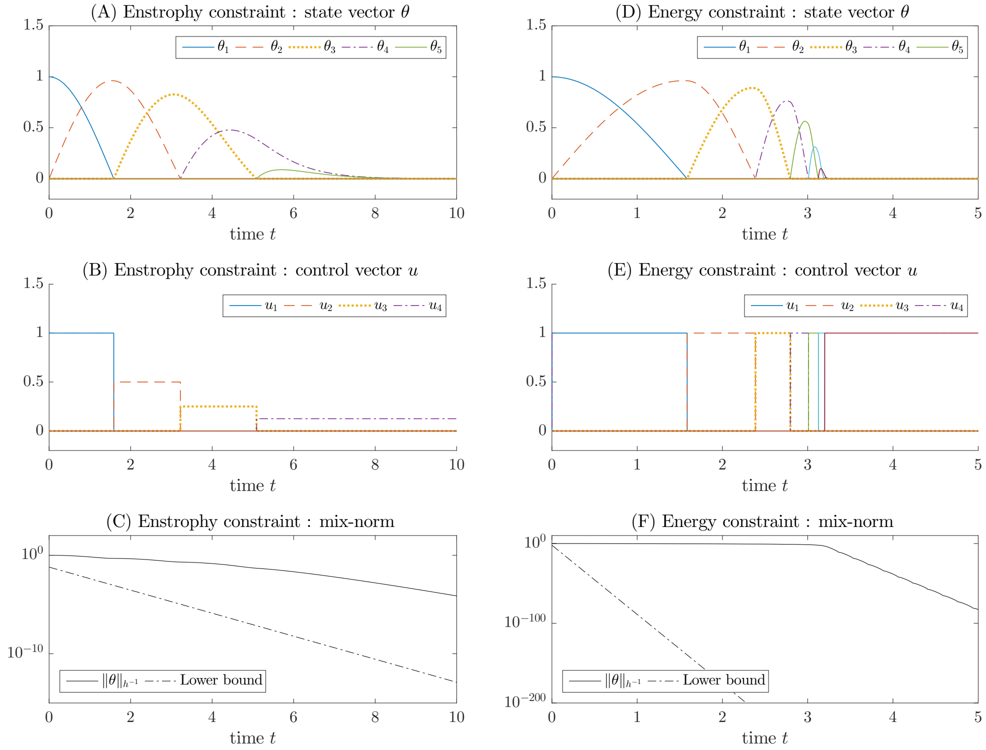

Figure 2.5: Local-in-time strategy with diffusion ($\kappa = 0.01$) starting initially from the most unmixed state. The entrophy-constrained case ($\frac{1}{\tau} = 1$) is shown on the left subplots where (A) shows the state, (B) shows the control, and (C) shows the mix-norm. The energy-constrained case ($U = 1$) is shown on the right subplots where (D) shows the state, (E) shows the control, and (F) shows the mix-norm. Lower bounds from applying *Theorem 2* in Appendix A.6 are also shown in subplots (C) and (F).

problem seen in 2.5.1. Recall that the initial state is the most umixed state, $\theta(0) = (1, 0, 0, \dots)^T$. We deal with the generalized constraint (2.5) parameterized by α. By employing results from the section 2.3, the optimal strategy is to initially use $u_1 = W^{(\alpha)}/k_1^\alpha$ with the other components of u set to zero. Thus, the motion will initially be in the θ_1-θ_2 plane with the following reduced state equation

$$\frac{d}{dt}\begin{pmatrix}\theta_1\\ \theta_2\end{pmatrix}=\begin{pmatrix}-\kappa k_1^2 & -k_1u_1\\ k_1u_1 & -\kappa k_2^2\end{pmatrix}\begin{pmatrix}\theta_1\\ \theta_2\end{pmatrix}.$$

Once the state encounters the axis $(0,1,0,0,\dots)^T$, the optimal strategy will switch to $u_2 = W^{(\alpha)}/k_2^\alpha$ until the state encounters the next axis $(0,0,1,0,0\dots)^T$. This trend will continue as long as the state keeps visiting each axis. In general, the motion is governed piecewise in time by the following state equation

$$\frac{d}{dt}\begin{pmatrix}\theta_n\\ \theta_{n+1}\end{pmatrix}=\begin{pmatrix}-\kappa k_n^2 & -k_nu_n\\ k_nu_n & -\kappa k_{n+1}^2\end{pmatrix}\begin{pmatrix}\theta_n\\ \theta_{n+1}\end{pmatrix} \tag{2.27}$$

after visiting the nth axis where $u_n = W^{(\alpha)}/k_n^\alpha$. The eigenvalue problem can be solved to produce the eigenvalues,

$$\lambda_\pm = -\frac{1}{2}\kappa(k_{n+1}^2 + k_n^2) \pm \frac{1}{2}\beta_n^{(\alpha)} \tag{2.28}$$

where $\beta_n^{(\alpha)} = \sqrt{\kappa^2(k_{n+1}^2 - k_n^2)^2 - 4k_n^{2(1-\alpha)}[W^{(\alpha)}]^2}$, and eigenvectors, in the θ_n-θ_{n+1} plane,

$$\Theta_\pm = \begin{pmatrix}\frac{1}{2}\kappa(k_{n+1}^2 - k_n^2) \pm \frac{1}{2}\beta_n^{(\alpha)}\\ k_n^{1-\alpha}W^{(\alpha)}\end{pmatrix}. \tag{2.29}$$

Define $\Theta_n(t) = (\theta_n(t), \theta_{n+1}(t))^T$. With the initial condition $\Theta_n(0) = (\bar{\theta}_n, 0)$ where $\bar{\theta}_n$ is the initial value of θ_n at the nth time interval, we obtain the solution

$$\Theta_n(t) = \frac{\bar{\theta}_n}{\beta_n^{(\alpha)}}\left[e^{\lambda_+ t}\Theta_+ - e^{\lambda_- t}\Theta_-\right]. \tag{2.30}$$

The fixed point at the origin is a stable spiral for low values of n. However, when

$$n > n_c = \frac{1}{1+\alpha} \log_2 \left(\frac{2W^{(\alpha)}}{3\kappa k_0^{1+\alpha}} \right), \tag{2.31}$$

the origin transitions to a stable node. At this point, the trajectory *cannot* move to the next plane since it must intercept the θ_{n+1} axis to do so. Figure 2.4 shows how the phase portrait changes with shell number n for the enstrophy-constrained case. Figure 2.5 shows the local-in-time strategy for both the fixed enstrophy and energy cases. In other words, it is no longer optimal to keep progressing to the next shell. This result demonstrates that even the slightest degree of diffusion prohibits the local-in-time strategy from performing perfect mixing in finite time which was possible with the perfectly non-diffusive case ($\kappa = 0$).

After seeing how diffusion can drastically change the system behavior under the local-in-time scheme, it is natural to ask "Is perfect mixing in finite time impossible?" For the enstrophy-constrained case, we apply *Theorem 2* in Appendix A.6 to the initial condition $\theta(0) = (1, 0, 0, \dots)^T$ with $\kappa = 0.01$ and $\tau = 1$. We find the lower bound on the mix-norm,

$$\|\nabla^{-1}\theta(t)\|_{L^2} \geq A \exp(r_+ t) \tag{2.32}$$

where $A = 0.0625$ and $r_+ \approx -2.69249$. For the energy-constrained case with identical parameters and initial condition, we have the lower bound,

$$\|\nabla^{-1}\theta(t)\|_{L^2} \geq A \exp(r_+ t) \tag{2.33}$$

where $A = 0.007$ and $r_+ \approx -199.805$. This shows that perfect mixing in finite time is indeed impossible with diffusion.

For global-in-time optimization, we use the same numerical scheme detailed in

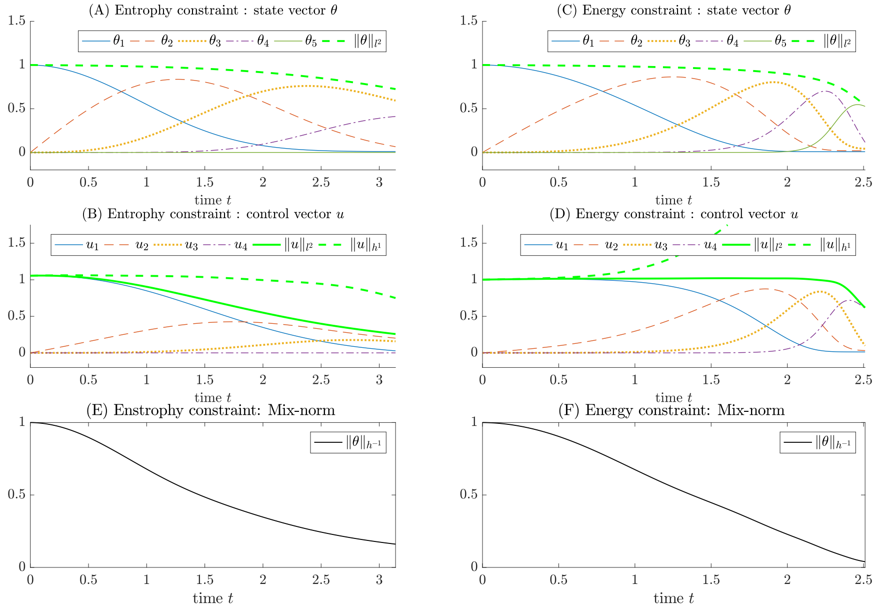

Figure 2.6: Global-in-time strategy applied to truncated shell model ($N = 5$) with diffusion ($\kappa = 0.01$) starting initially from the most unmixed state. The entrophy-constrained case ($\frac{1}{\tau} = 1, T = \pi$) is shown on the left subplots where (A) shows the state, (B) shows the control, and (C) shows the mix-norm. The energy-constrained case ($U = 1, T = 0.8\pi$) is shown on the right subplots where (D) shows the state, (E) shows the control, and (F) shows the mix-norm.

section 2.5.3 here with $\kappa \neq 0$ in equation (2.11). Figure 2.6 shows the numerical solution for a truncated shell model with $N = 5$ and $\kappa = 0.01$. The global-in-time strategy appears similar to that with non-diffusive situation in the sense that we see the feature of ‘short-cutting’ relative to local-in-time strategy. We also notice the expected overall decay of the l^2 norm of θ. We see that the optimal control no longer conserves energy or enstrophy with diffusion. Specifically, we see that it is optimal to use more of the budget earlier on rather than later for both the energy and enstrophy constrained cases.

2.7 Discussion

We first focused on mixing without diffusion. The enstrophy-constrained local-in-time strategy exhibited exponential decay while energy-constrained local-in-time strategy showed linear decay — and hence perfect mixing in finite time. We obtained an analytic solution to the global-in-time optimization problem of the 3-shell truncated model by using methods from nuclear magnetic resonance. The global-in-time strategy applied to the 3-shell truncated model showed an improvement on the mixing rate relative to that of the local-in-time strategy by using a short-cutting method (illustrated in figure 2.2). This short-cutting feature generalized to models with higher truncation number N. We were surprised to find that it is optimal to use the (energy or enstrophy) budget uniformly in time (rather than consuming more budget earlier than later or vice versa). This is consistent with the work of Mathew *et al.* [31] that demonstrated this feature in the partial differential equation context.

Mixing with diffusion was explored and demonstrated interesting effects. Perfect mixing in finite time for the local-in-time strategy, while constraining either energy or enstrophy, becomes *impossible* (recall that it was at least possible for the energy-constrained case without diffusion). The local-in-time dynamics were restricted to θ_n-θ_{n+1} planes piecewise in time similar to that of the local-in-time strategy seen without diffusion. As the state vector progressed to θ_n-θ_{n+1} planes of larger n, the diffusive terms progressively dominate over the advective terms in our shell model. Thus, a plane is eventually reached where it is no longer advantageous to progress to the next plane. This suggests that we have succeeded at mixing to a length scale where diffusion can then take over. For energy-constrained flow, this length scale is

$$l_u = \frac{3}{2}\kappa/U. \tag{2.34}$$

For the enstrophy-constrained case, this length scale is

$$l_\tau = \sqrt{\frac{3}{2}\kappa\tau} \tag{2.35}$$

which is naturally interpreted as the Batchelor length scale in turbulent mixing theory [12, 23, 42, 4, 7, 35]. In fact, (2.35) has the same scaling in molecular diffusivity κ and rate-of-strain $1/\tau$ as the turbulent theory type. Local-in-time optimization may suggest that we should mix the tracer concentration until we arrive at these small critical length scales.

But, how should one use a flow intensity budget over time during this approach to small scales? For this, we turn to global-in-time optimization. Recall that global-in-time optimization without diffusion revealed that it is optimal to use your budget uniformly in time. We find however that this is no longer the case with diffusion. It is optimal to use more of the (energy or enstrophy) budget earlier than later. Therefore, this suggests that budget use is more effective at larger scales away from the Batchelor length scale.

These observations prompt the following questions: "Will diffusion always dominate advection eventually?" and "If so, does this mean that perfect mixing in finite time is impossible?" We showed in Appendix A.6 that perfect mixing in finite time is indeed impossible for the enstrophy and energy local-in-time constraints by producing exponential lower bounds on the mix-norm in both cases.

In conclusion the developed shell model preserves many known features of the partial differential equation problem. For instance, the shell model performed

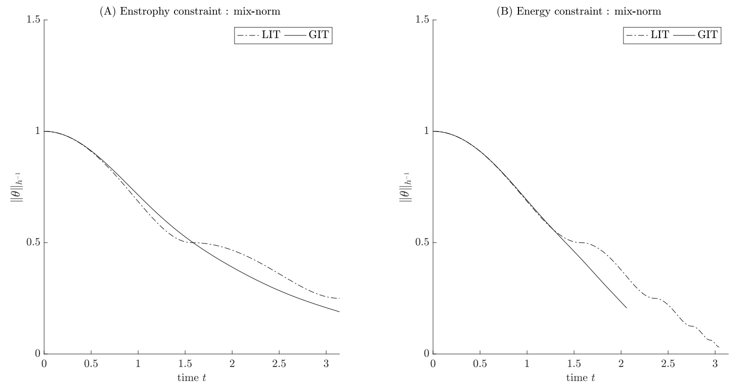

Figure 2.7: The mix-norm over time for the local-in-time (same data from figure 2.1) and global-in-time (same data from figure 2.3) strategies applied to the 6-shell truncated shell model without diffusion. The initial condition is $\theta(0) = \mathbf{e}_1$. The left plot (A) shows the enstrophy-constrained case with $\tau = 1$ and the right plot (B) shows the energy-constrained case with $U = 1$.

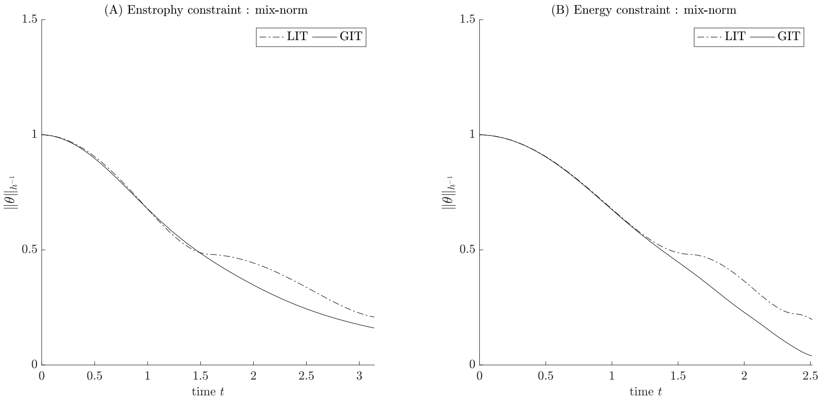

Figure 2.8: The mix-norm over time for the local-in-time (same data from figure 2.5) and global-in-time (same data from figure 2.6) strategies applied to the 5-shell truncated shell model with diffusion. The initial condition is $\theta(0) = \mathbf{e}_1$. The left plot (A) shows the enstrophy-constrained case with $\tau = 1$ and the right plot (B) shows the energy-constrained case with $U = 1$.

perfect mixing in finite time without diffusion and with an energy constraint (Lunasin *et al.* [30]), and showed exponential mix-norm decay without diffusion

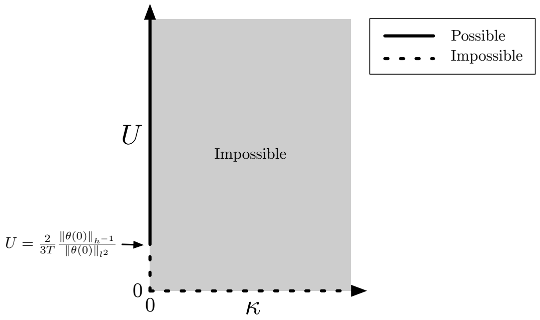
Possible
Impossible
U
Impossible
$U = \frac{2}{3T}\frac{\|\theta(0)\|_{h^{-1}}}{\|\theta(0)\|_{l^2}}$
0
0
κ

Figure 2.9: The possibility of perfect mixing in finite time with energy constraint $\|u\|_{L^2} = U$ and diffusion constant κ.

and with an enstrophy constraint (Seis [40] and Iyer *et al.* [24]). For the case with diffusion in the partial differential equation setting, it remains to be shown that exponential bounds on the mix-norm exist (as it was shown for the shell model) with L^2 norm constraints. However for the L^∞ extensions of these constraints, strictly positive lower bounds on the mix-norm can be derived by extending the analysis of Poon [37]. This rules out the possibility of perfect mixing in finite time for this situation (and will be discussed further in the next chapter).

Like any reduced model, there are limitations. Some of the bound estimates obtained rely on series inequalities where their integral analogs do not hold (i.e. for series, we have $\sum_n a_n b_n \leq \sum_n a_n \sum_m b_m$ for $a_n, b_n > 0$; while for integrals, the analogous expression $\int f(x)g(x)dx$ is not less than $\int f(x)dx \int g(x)dx$ for $f, g > 0$ in general). Dimensional effects such as incompressibility or integrating volume factors originating in Fourier transforms are neglected.

Nevertheless, we were able to obtain insights into mixing and arrive at the following answers, in the context of the shell model, to the questions 1 and 2 presented in the introduction:

1. *Global-in-time performed slightly better than local-in-time with and without*

diffusion (See figures 4.5 and 2.8).

2. *Without diffusion it is optimal to use a stirring budget uniformly in time; with diffusion it is optimal to expend more of the stirring budget earlier than later.*

In fact, it is optimal to use a stirring budget uniformly in time also in partial differential equation context without diffusion (see Chapter IV). We surmise that the other conclusions hold true in this setting as well.

Furthermore, we found that perfect mixing in finite time is impossible for the enstrophy constraint $\|u(t)\|_{h^1} = \frac{1}{\tau}$ for all values of rate-of-strain $\frac{1}{\tau} \geq 0$ and $\kappa \geq 0$. As for the energy-constraint $\|u(t)\|_{l^2} = U$, perfect mixing in finite time is impossible for most of the U-κ parameter space (see figure 2.9) except for the singular case of $\kappa = 0$ and $U \geq \frac{2}{3T} \frac{\|\nabla^{-1}\theta(0)\|_{L^2}}{\|\theta(0)\|_{L^2}}$ that was realized by the local-in-time strategy. Therefore perfect mixing in finite time is a phenomena confined to pure advection ($\kappa = 0$).

CHAPTER III

Local-in-time optimization [1]

3.1 Introduction

In this chapter, we make progress towards answering “What is the most optimal mixing rate in the presence of diffusion for an enstrophy or energy constrained flow?” This question was also asked in the context of the shell model. We would like to determine if the predictions of the shell model hold in the partial differential equation setting.

We approach the posed question by considering the general setup, introduced in the introduction chapter, of the evolution of passive scalar in a periodic box. We consider the local-in-time optimization problem introduced by Z. Lin *et al.* [28] in context of pure advection. We now study this optimization problem with the inclusion of diffusion. Local-in-time optimization seeks to find the optimal flow that achieves the best instantaneous mixing rate. We will see that the best choice leads to a $\mathbf{u}$ that depends on θ. This feedback causes the dynamics of θ governed by (1.1) to be nonlinear.

We will demonstrate that homogenization via diffusion and filamentation via

[1] The content of this chapter is included within a journal article accepted for publication in Nonlinearity [34].

advection can sometimes be in conflict and collectively produce a negative impact on mixing. We show numerical evidence that filamentation length scale appears to be limited by the Batchelor scale as seen in the shell model. Even when actively trying to choose the most optimal flow to enhance filamentation. Thus, this may suggest that the Batchelor scale does not only limit turbulent flows but also all incompressible flows under the flow constraints considered here. Although these quantities have been known in the context of turbulence theory, the impact of these limitations on mixing rates has not been fully studied to our knowledge.

The chapter is organized as follows. We introduce the necessary theory regarding local-in-time optimization, a shell model, and L^∞ flow constraints in section 3.2. Section 3.3 details the methodology and results of numerically implementing local-in-time flow optimization. Lastly, we finish with a discussion and conclusion in sections 3.4 and 3.5 respectively.

3.2 Theory

3.2.1 Local-in-time flow optimization

We will consider the evolution of a tracer quantity θ governed by equation 1.1 under an the incompressible flow $\mathbf{u}$. Recall the flow is constrained by enstrophy $\|\nabla \mathbf{u}\|_{L^2} = \Gamma L^{d/2}$ or energy $\|\mathbf{u}\|_{L^2} = U L^{d/2}$ where Γ is the root mean square rate-of-strain and U is the root mean square speed.

For the enstrophy-bounded flow problem, we choose the same length scale L, the velocity scale $L\Gamma$, and the time scale $1/\Gamma$. For the energy-bounded flow problem, we non-dimensionalize the system by choosing L as the length scale, U as the velocity scale, and L/U as the time scale. Both scalings produce the following form of the advection-diffusion equation,

$$\partial_t \theta + \mathbf{u} \cdot \nabla \theta = \frac{1}{Pe} \Delta \theta, \tag{3.1}$$

where $Pe = \frac{\Gamma L^2}{\kappa}$ for the enstrophy-constrained case and $Pe = \frac{UL}{\kappa}$ for the energy-constrained case. The non-dimensional flow constraints become $\|\nabla \mathbf{u}\|_{L^2} = 1$ or $\|\mathbf{u}\|_{L^2} = 1$.

We consider the local-in-time optimization strategy first introduced by Lin *et al.* [28] in the case without diffusion. We find that this strategy generalizes to the case with diffusion. The local-in-time optimal velocity fields maximize the instantaneous mixing rate by minimizing $\frac{d}{dt}\|\nabla^{-1}\theta\|_{L^2}^2$. We highlight that local-in-time optimization is not the same as global-in-time or finite-time optimization where the objective is to minimize $\|\theta(\,\cdot\,, T)\|_{H^{-1}}$ at the final time T. These objectives generally produce different results. In the context of the shell model, however, these strategies yielded similar decay rates. The differences between these two objectives under the evolution of (3.1) will be the focus of the next chapter and future study.

The optimal velocity fields are given instantaneously for the enstrophy case by (in non-dimensional form)

$$\mathbf{u} = \frac{-\Delta^{-1}\mathbb{P}(\theta \nabla \Delta^{-1}\theta)}{\langle |\nabla^{-1}\mathbb{P}(\theta \nabla \Delta^{-1}\theta)|^2 \rangle^{1/2}} \tag{3.2}$$

and for the energy case by

$$\mathbf{u} = \frac{\mathbb{P}(\theta \nabla \Delta^{-1}\theta)}{\langle |\mathbb{P}(\theta \nabla \Delta^{-1}\theta)|^2 \rangle^{1/2}} \tag{3.3}$$

where $\mathbb{P}$ is the Leray divergence-free projector given by $\mathbb{P}(\mathbf{v}) = \mathbf{v} - \nabla\Delta^{-1}(\nabla \cdot \mathbf{v})$

and $\langle\cdot\rangle$ is the spatial average. These flows will be studied numerically later and is the main focus of this chapter.

We introduce the following measures as useful observables of mixing over time. we use the H^{-1} norm to define the (exponential) rate of mixing as

$$r(t) = -\frac{\frac{d}{dt}\|\nabla^{-1}\theta\|_{L^2}}{\|\nabla^{-1}\theta\|_{L^2}}. \tag{3.4}$$

We define the following ratio as a measure of the characteristic filamentation length scale:

$$\lambda(t) \equiv 2\pi\frac{\|\nabla^{-1}\theta(\,\cdot\,,t)\|_{L^2}}{\|\theta(\,\cdot\,,t)\|_{L^2}}. \tag{3.5}$$

Note that if the tracer concentration field is composed of only one Fourier mode with wave number $\mathbf{k}$ (i.e. $\theta(\mathbf{x},t) = Re[Ae^{-i\mathbf{k}\cdot\mathbf{x}}]$ where A is a complex constant), then $\lambda(t)$ returns the wavelength of the wave number $\mathbf{k}$. In general, λ is the weighted root mean square wavelength with weights given by $|\theta_{\mathbf{k}}|^2/\|\theta\|_{L^2}^2$.

3.2.2 Shell model predictions of local-in-time optimization

The shell model is a model that mimics the spectral dynamics present in the advection-diffusion equation. The model consists of a system of ordinary differential equations with nearest-neighbour coupling between 'shells' in wave number space. [33] performed local-in-time mixing optimization in this model. The shell-model analysis predicts a limiting length scale given by the Batchelor scale, $\Lambda_\Gamma = \sqrt{\frac{\kappa}{\Gamma}}$ and its generalization $\Lambda_U = \frac{U}{\kappa}$. The non-dimensional versions are given by $\lambda_\Gamma = \frac{1}{\sqrt{Pe}}$ and $\lambda_U = \frac{1}{Pe}$. From here forward, we will refer to the Batchelor scale to mean either λ_Γ or its generalization λ_U. The predicted long-term rates (after reaching the Batchelor scale) are given by $R_\Gamma = \kappa/\lambda_\Gamma^2$ and $R_U = \kappa/\lambda_U^2$. The non-dimensional versions are given by $r_\Gamma = 1$ and $r_U = Pe$.

3.2.3 Bounds for L^∞ constrained flows

We now consider a subset of L^2 constrained flows — those belonging to L^∞. In this restricted setting the rate-of-strain and speed are bounded point-wise uniformly in space and time rather than demanding that they merely be L^2 integrable as before. Similar analysis of what follows has been attempted in the context of L^2 constrained flows, but without success, and appears to be challenging. Thus, we focus on these restricted L^∞ constrained subsets of flows where we have been successful at determining bounds on λ and measures of mixing.

3.2.3.1 Results for $\|\nabla \mathbf{u}\|_{L^\infty} = 1$

From (3.1), we find

$$\frac{1}{(2\pi)^2}\frac{d\lambda^2}{dt} = \frac{2}{Pe}\left[\frac{\|\nabla\theta\|_{L^2}^2\|\nabla^{-1}\theta\|_{L^2}^2}{\|\theta\|_{L^2}^4} - 1\right] + 2\frac{\int_D \nabla^{-1}\theta\cdot\nabla\mathbf{u}\cdot\nabla^{-1}\theta\, d^d\mathbf{x}}{\|\theta\|_{L^2}^2}$$

and by Hölder's inequality, we deduce

$$\frac{1}{(2\pi)^2}\frac{d\lambda^2}{dt} \geq \frac{2}{Pe}\left[\frac{\|\nabla\theta\|_{L^2}^2\|\nabla^{-1}\theta\|_{L^2}^2}{\|\theta\|_{L^2}^4} - 1\right] - \frac{2}{(2\pi)^2}\lambda^2.$$

This establishes a lower bound on λ at each instant: by apply Grönwall's inequality and the fact that the bracketed term is greater than or equal to zero, it follows that

$$\lambda(t) \geq \lambda(0)e^{-t}. \tag{3.6}$$

Furthermore,

$$\begin{aligned}\frac{d}{dt}\left(\frac{\|\nabla\theta\|_{L^2}^2}{\|\theta\|_{L^2}^2}\right) &= \frac{\|\theta\|_{L^2}^2\frac{d}{dt}\|\nabla\theta\|_{L^2}^2-\|\nabla\theta\|_{L^2}^2\frac{d}{dt}\|\theta\|_{L^2}^2}{\|\theta\|_{L^2}^4}\\ &= \frac{-2\int_D\nabla\theta\cdot\nabla\mathbf{u}\cdot\nabla\theta\,d^d\mathbf{x}-\frac{2}{Pe}\|\Delta\theta\|_{L^2}^2}{\|\theta\|_{L^2}^2}+\frac{2}{Pe}\frac{\|\nabla\theta\|_{L^2}^4}{\|\theta\|_{L^2}^4}\\ &= -\frac{2}{Pe}\left(\frac{\|\Delta\theta\|_{L^2}^2}{\|\theta\|_{L^2}^2}-\frac{\|\nabla\theta\|_{L^2}^4}{\|\theta\|_{L^2}^4}\right)-2\frac{\int_D\nabla\theta\cdot\nabla\mathbf{u}\cdot\nabla\theta\,d^d\mathbf{x}}{\|\theta\|_{L^2}^2}\\ &\leq 2\frac{\|\nabla\theta\|_{L^2}^2}{\|\theta\|_{L^2}^2}\end{aligned}$$

and using $\frac{d}{dt}\|\theta\|_{L^2}^2=-\frac{2}{Pe}\|\nabla\theta\|_{L^2}^2$, it follows that

$$\|\theta\|_{L^2}\geq\|\theta_0\|_{L^2}\exp\left[-\frac{1}{2Pe}\frac{\|\nabla\theta_0\|_{L^2}^2}{\|\theta_0\|_{L^2}^2}\left(e^{2t}-1\right)\right]. \tag{3.7}$$

Using this with (3.6), we deduce the double exponential lower bound

$$\|\nabla^{-1}\theta\|_{L^2}\geq\|\nabla^{-1}\theta_0\|_{L^2}\exp\left[-t-\frac{1}{2Pe}\frac{\|\nabla\theta_0\|_{L^2}^2}{\|\theta_0\|_{L^2}^2}\left(e^{2t}-1\right)\right]. \tag{3.8}$$

Therefore, perfect mixing in finite time is impossible for bounded rate-of-strain flows.

3.2.3.2 Results for $\|\mathbf{u}\|_{L^\infty}=1$

Here we follow and refine an analysis of Poon [37] to show that the presence of diffusion also rules out perfect mixing in finite time for bounded velocity flows as well. First note that

$$\begin{aligned}\|\nabla\theta\|_{L^2}^2 &= -2\int_D\theta\Delta\theta\,d^d\mathbf{x}\\ &= Pe\int_D\theta\left(\partial_t\theta-\frac{1}{Pe}\Delta\theta\right)d^d\mathbf{x}-Pe\int_D\theta\left(\partial_t\theta+\frac{1}{Pe}\Delta\theta\right)d^d\mathbf{x},\end{aligned}$$

$$\begin{aligned}
\frac{d}{dt}\|\theta\|_{L^2}^2 &= 2\int\limits_D \theta\partial_t\theta\, d^d\mathbf{x} \\
&= \int\limits_D \theta\left(\partial_t\theta - \frac{1}{Pe}\Delta\theta\right)\, d^d\mathbf{x} + \int\limits_D \theta\left(\partial_t\theta + \frac{1}{Pe}\Delta\theta\right)\, d^d\mathbf{x},
\end{aligned}$$

and

$$\begin{aligned}
\frac{d}{dt}\|\nabla\theta\|_{L^2}^2 &= -2\int\limits_D \partial_t\theta\Delta\theta\, d^d\mathbf{x} \\
&= Pe\int\limits_D \left(\partial_t\theta - \frac{1}{Pe}\Delta\theta\right)^2\, d^d\mathbf{x} - Pe\int\limits_D \left(\partial_t\theta + \frac{1}{Pe}\Delta\theta\right)^2\, d^d\mathbf{x}.
\end{aligned}$$

Then simplify and compute:

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\|\nabla\theta\|_{L^2}^2}{\|\theta\|_{L^2}^2}\right) &= \frac{1}{\|\theta\|_{L^2}^4}\left[\|\theta\|_{L^2}^2\frac{d}{dt}\|\nabla\theta\|_{L^2}^2 - \frac{d}{dt}\|\theta\|_{L^2}^2\|\nabla\theta\|_{L^2}^2\right] \\
&= \frac{1}{\|\theta\|_{L^2}^2}\Bigg[Pe\int\limits_D \left(\partial_t\theta - \frac{1}{Pe}\Delta\theta\right)^2\, d^d\mathbf{x} \\
&\qquad\qquad -Pe\int\limits_D \left(\partial_t\theta + \frac{1}{Pe}\Delta\theta\right)^2\, d^d\mathbf{x}\Bigg] \\
&- \frac{1}{\|\theta\|_{L^2}^4}\Bigg[Pe\left(\int\limits_D \theta\left(\partial_t\theta - \frac{1}{Pe}\Delta\theta\right)\, d^d\mathbf{x}\right)^2 \\
&\qquad\qquad -Pe\left(\int\limits_D \theta\left(\partial_t\theta + \frac{1}{Pe}\Delta\theta\right)\, d^d\mathbf{x}\right)^2\Bigg].
\end{aligned}$$

Using Hölder's inequality and (3.1), this simplifies to the observation originally noted by Poon [37]:

$$\frac{d}{dt}\left(\frac{\|\nabla\theta\|_{L^2}^2}{\|\theta\|_{L^2}^2}\right) \le \frac{Pe}{\|\theta\|_{L^2}^2}\left[\int\limits_D (\mathbf{u}\cdot\nabla\theta)^2\, d^d\mathbf{x}\right]. \tag{3.9}$$

Now applying Hölder's inequality again we have

$$\frac{d}{dt}\left(\frac{\|\nabla\theta\|_{L^2}^2}{\|\theta\|_{L^2}^2}\right) \leq Pe\frac{\|\nabla\theta\|_{L^2}^2}{\|\theta\|_{L^2}^2} \tag{3.10}$$

and thus

$$\frac{\|\nabla\theta\|_{L^2}}{\|\theta\|_{L^2}} \leq \frac{\|\nabla\theta_0\|_{L^2}}{\|\theta_0\|_{L^2}}\exp\left(\frac{Pe}{2}t\right).$$

The inequality $\|\nabla\theta\|_{L^2}\|\nabla^{-1}\theta\|_{L^2} \geq \|\theta\|_{L^2}^2$ then ensures that

$$\lambda(t) \geq 2\pi\frac{\|\theta_0\|_{L^2}}{\|\nabla\theta_0\|_{L^2}}\exp\left(-\frac{Pe}{2}t\right). \tag{3.11}$$

Using (3.10) together with $\frac{d}{dt}\|\theta\|_{L^2}^2 = -\frac{2}{Pe}\|\nabla\theta\|_{L^2}^2$ we observe that

$$\|\theta\|_{L^2} \geq \|\theta_0\|_{L^2}\exp\left[-\frac{1}{Pe^2}\frac{\|\nabla\theta_0\|_{L^2}^2}{\|\theta_0\|_{L^2}^2}\left(e^{Pe\,t}-1\right)\right]$$

and this combined with (3.11) implies another (distinct) double exponential

$$\|\nabla^{-1}\theta\|_{L^2} \geq \frac{\|\theta_0\|_{L^2}^2}{\|\nabla\theta_0\|_{L^2}}\exp\left[-\frac{Pe}{2}\,t-\frac{1}{Pe^2}\frac{\|\nabla\theta_0\|_{L^2}^2}{\|\theta_0\|_{L^2}^2}\left(e^{Pe\,t}-1\right)\right]. \tag{3.12}$$

3.3 Numerical experiment: local-in-time optimization

3.3.1 Methodology

We solve (3.1) with either flow (3.2) or (3.3) by using a Fourier basis to represent the discretized spatial domain with a 4th order Runge-Kutta time-stepping method. We slightly perturb the concentration field $\theta_0(\mathbf{x}) = \sin(2\pi x/L)$ by evolving the field according to (3) with a steady sin flow given by $\mathbf{u}(\mathbf{x}) = \sin(2\pi y/L)\hat{x}$ for a time duration of 0.01. The concentration field, resulting from this short time integration, is then used as an initial condition for the local-in-time optimization scheme. This perturbation is necessary since the denominator is zero in both expressions (4) and (5) for pure Fourier modes such as θ_0 [2]. The number of Fourier modes is chosen large enough to resolve the spatial resolution and give by the following criteria $N = \min\left[2^{ceil(log_2(4(M-1)+6))}, 512\right]$ where $M = L/(0.25\lambda_B)$ and λ_B is the appropriate Batchelor scale; The choice of $N = 2^{ceil(log_2(4(M-1)+6))}$ is suggested as a rule-of-thumb by Ref. [6]. The cap of 512 is suitable for the range of Pe values considered — the Batchelor wavenumber $k_B = 2\pi/\lambda_B$ is well within the range of Fourier modes present even after considering 2/3 dealiasing. The time step is chosen by the CFL condition $dt = 0.25 \min\left[L/(UN), L^2/(\kappa N^2)\right]$ for the enstrophy-constrained case and $dt = 0.25 \min\left[1/(\Gamma N), L^2/(\kappa N^2)\right]$ for the energy-constrained case. All simulation code was created in the programming language Python with package modules, pyfftw and numpy. The code is provided in Appendix B.

3.3.2 Results

We now investigate the mixing performance under the local-in-time optimal flows. Figure 3.1 and 3.2 show how the different mixing measures (H^{-1}, L^2, and H^1 norms) vary in time for different values of Pe for the enstrophy and energy constrained cases respectively. Notice how the long-term mixing rate appears to be exponential for all three mixing measures. This exponential rate is consistent with shell model predictions, yet weaker than the double-exponential decay rate derived by the L^∞ constrained flow analysis.

Figure 3.3 shows the evolution of a scalar field under the optimal flow for the enstrophy constraint. The top film strip corresponds to $Pe = \infty$ while the bottom is $Pe = 256$. The time evolution is initially similar but soon diverges over

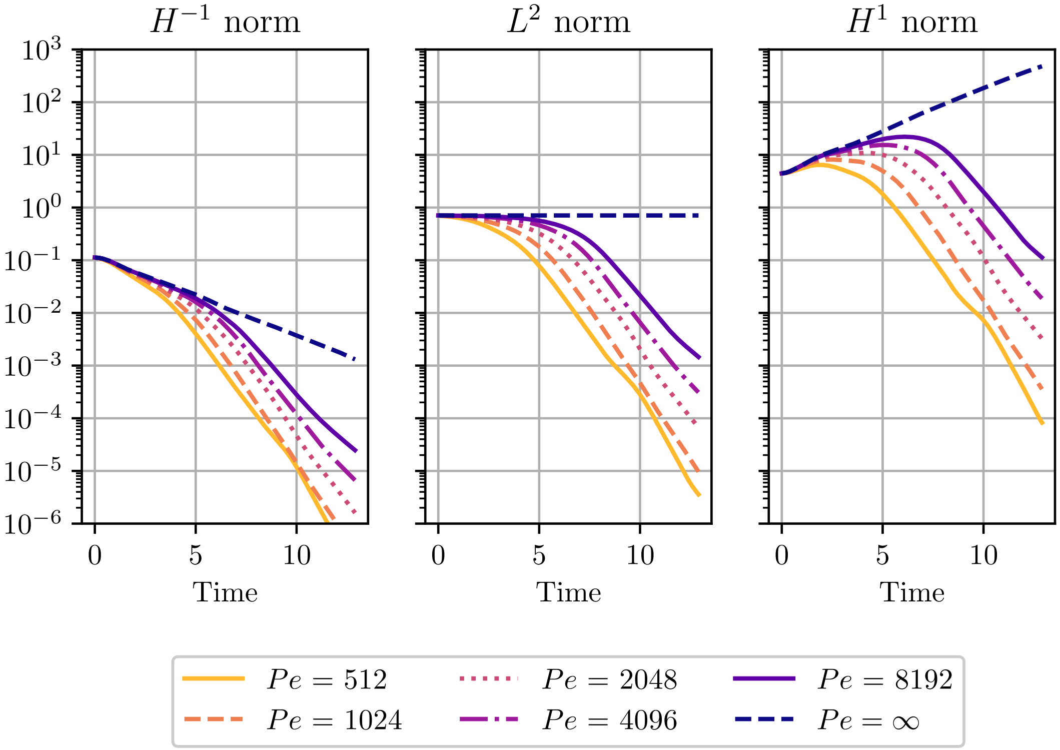

Figure 3.1: H^{-1}, L^2, and H^1 norms of the concentration field under the optimal enstrophy-constrained flow.

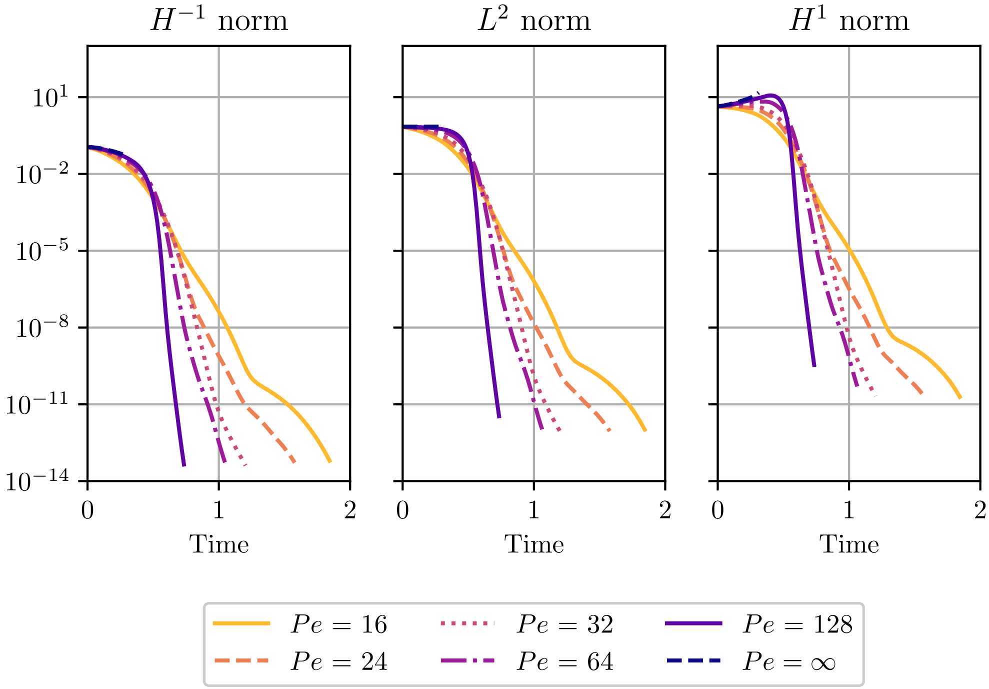

Figure 3.2: H^{-1}, L^2, and H^1 norms of the concentration field under the optimal energy-constrained flow.

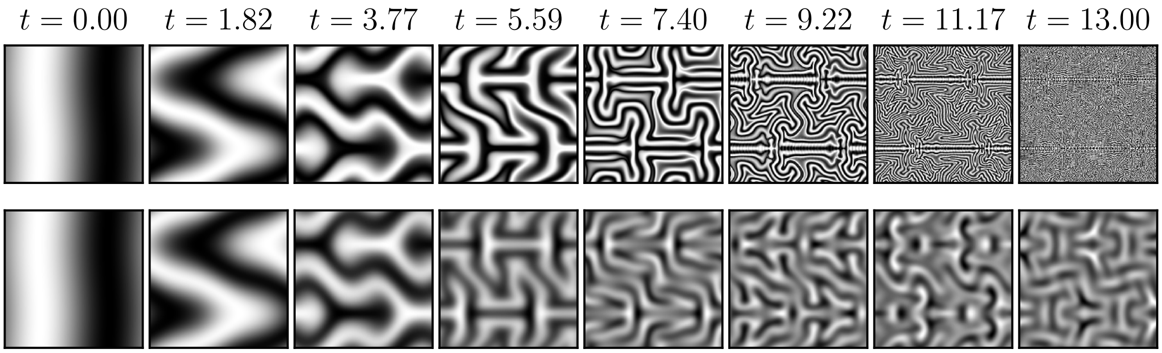

Figure 3.3: Local-in-time optimization with enstrophy constraint. Top filmstrip is for $Pe = \infty$ and the bottom filmstrip is $Pe = 2048$. Note that the grey-scale for the $Pe = \infty$ is constant in time while it is adjusted to show the tracer concentration structure in the finite Pe case.

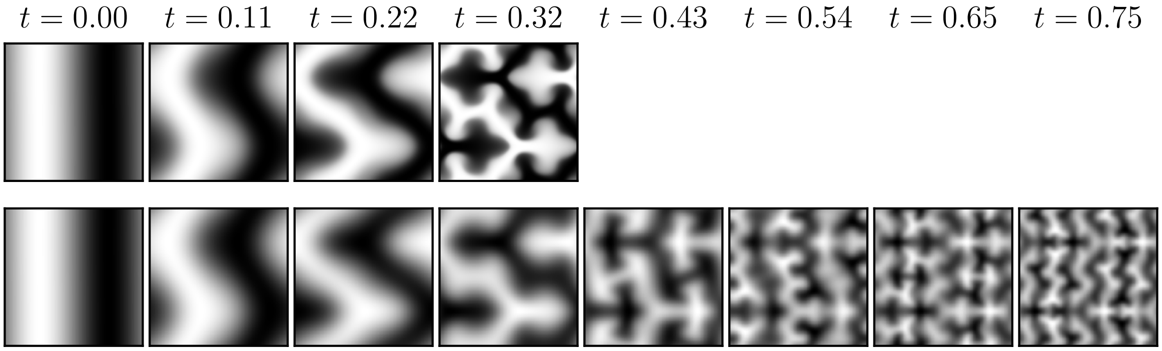

Figure 3.4: Local-in-time optimization with energy constraint. Top filmstrip is for $Pe = \infty$ and the bottom filmstrip is $Pe = 32$. Note that the grey-scale for the $Pe = \infty$ is constant in time while it is adjusted to show the tracer concentration structure in the finite Pe case. The numerical computation is truncated at time $t = 0.34$ due to length scales rapidly decreasing past the grid size resolution immediately after $t = 0.34$. Fixed energy constrained flows that produce infinitesimally small lengths in finite time have been constructed [30]. We suspect that the same phenomena may be occurring here.

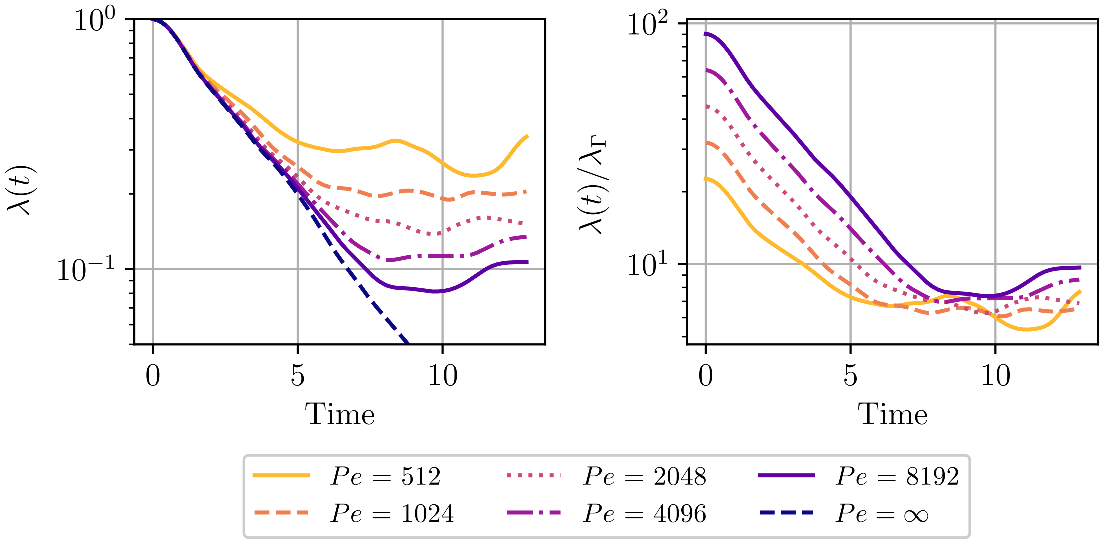

Figure 3.5: The left subplot shows the filament length λ over time subject to the optimal enstrophy-constrained flow. The right subplot is the same data except scaled: $\lambda(t)/\lambda_\Gamma = \lambda(t)\sqrt{Pe}$.

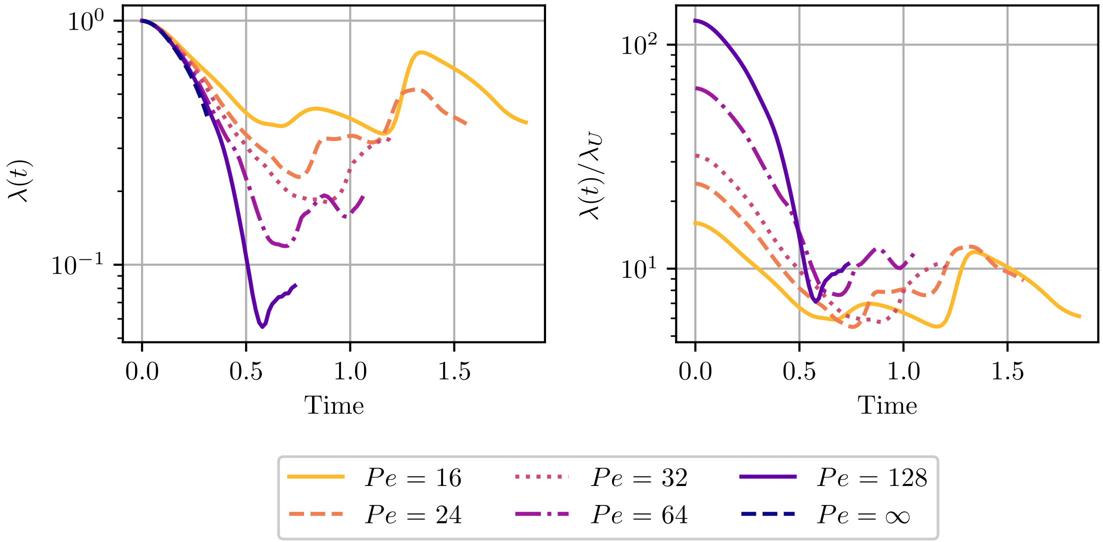

Figure 3.6: The left subplot shows the filament length λ over time subject to the optimal energy-constrained flow. The right subplot is the same data except scaled: $\lambda(t)/\lambda_U = \lambda(t)Pe$.

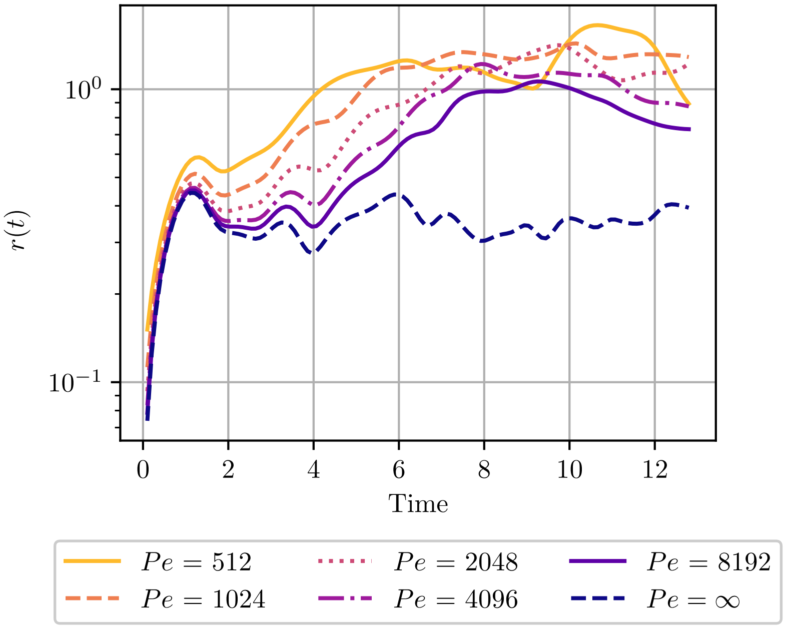

Figure 3.7: Mixing rate $r(t)$ over time when subject to the optimal enstrophy-constrained flow.

time. Figure 3.4 shows the evolution for the energy constraint. The top film strip corresponds to $Pe = \infty$ while the bottom is $Pe = 32$. Notice that, unlike the $Pe = \infty$ cases, the flows with finite Pe are incapable of creating length scales arbitrarily small for either the energy or enstrophy cases. The left subplot of Figures 3.5 and 3.6 shows this phenomena more quantitatively by showing λ over time eventually reaching a plateau. The shell-model prediction of this limiting length scale is the Batchelor scale given by $\lambda_\Gamma = 1/\sqrt{Pe}$ for the enstrophy case and $\lambda_U = 1/Pe$ for the energy case. The right plots of Figures 3.5 and 3.6 shows scaled versions of λ given by λ/λ_Γ and λ/λ_U respectively. Notice how they plateau around an $O(1)$ constant. Thus this result is consistent with the shell-model predictions.

The mixing rates for the enstrophy case are shown in Figure 3.7. The rate

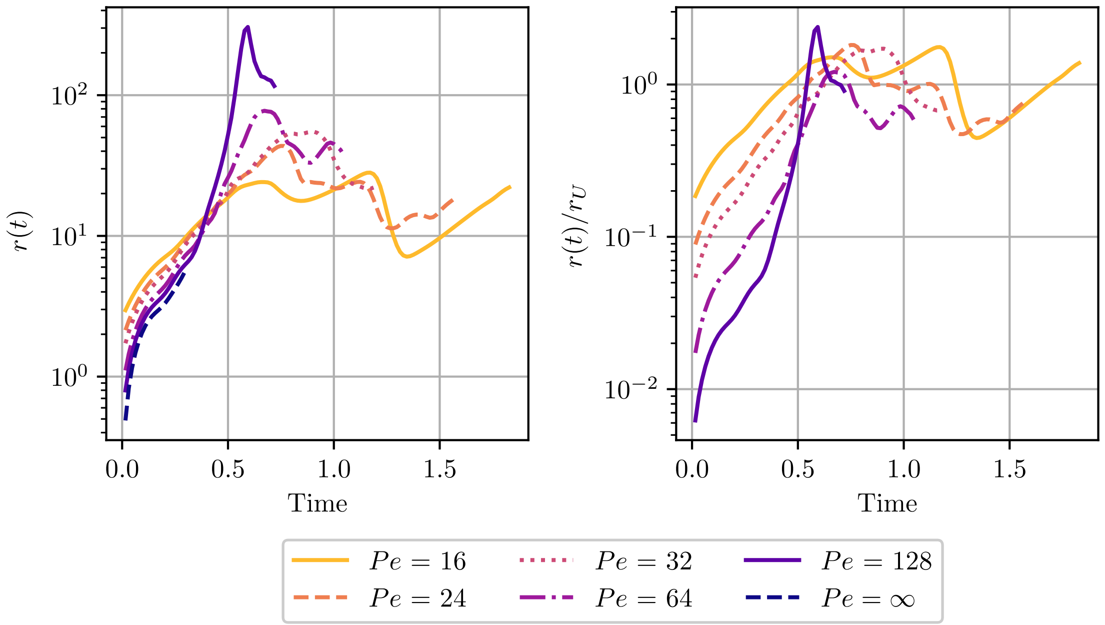

Figure 3.8: The left subplot shows the mixing rate $r(t)$ over time when subject to the optimal energy-constrained flow. The right subplot is the same data except scaled: $r(t)/r_U = r(t)/Pe$.

during the transient phase is Γ which is consistent with rates expected from $Pe = \infty$ mixing studies. For all Pe considered, there is an increase in the rate of mixing after transient behaviour has finished to a long-term rate. Perhaps surprisingly, this long-term mixing rate appears to be *independent* of Pe for fixed enstrophy. This suggests that the optimal long-term rate of mixing is only dependent on the rate-of-strain Γ and not influenced by the strength of diffusion.

It should be noted that the onset of the long-term rate is affected by the value of Pe. When there is strong diffusion (small Pe), the Batchelor scale is reached quickly. From the work of G. Iyer *et al.* [24] and C. Seis [40], we know that λ decreases at most exponentially for $Pe = \infty$. If we assume that the local-in-time optimal flows nearly saturate this bound in the transient phase, we model λ as $\lambda(t) = \lambda(0)\exp(-\alpha t)$ during this time. We expect the critical transition time t_c that marks the end of this transient period to satisfy $\lambda(t_c) = \lambda_\Gamma$. This time is theorized to be $t_c = \frac{1}{\alpha}\ln(\lambda(0)/\lambda_\Gamma) = \frac{1}{\alpha}\ln(\sqrt{Pe})$ for $Pe > 1$ (If $Pe \leq 1$, then there is no transient phase). Hence, a smaller value of Pe will result in an earlier onset of the long-term rate of mixing. Therefore, it is advantageous to have strong diffusion (small Pe) so that there is an earlier onset of the long-term mixing rate (although independent of Pe) which is an improvement over the mixing rate of the purely non-diffusive situation ($Pe = \infty$).

For the energy case, the long-term mixing rate decreases with decreasing Pe (see the left subplot of figure 3.8). *Thus, strong diffusion results in a weak long-term mixing rate.* The right subplot of Figure 3.8 is $r/r_U = r/Pe$. We see oscillations of r/r_U around a value that is $O(1)$ which indicates that our numerical results are consistent with our predictions from the shell model. Thus, the long-term mixing rate is proportional to Pe in contrast to the long-term mixing rate of enstrophy which carries no dependence on Pe.

For the energy case, the onset of the long run-mixing behaviour can be determined by the following model. From the work of E. Lunasin *et al.* [30] on the fixed energy case, $\lambda(t)$ can decrease linearly in time to produce perfect mixing in finite time. We model the transient phase as $\lambda(t) = \lambda(0)(1 - \beta t)$. Therefore, we theorize that the critical transition time is $t_c = \frac{1}{\beta}(1 - \lambda_U/\lambda(0)) = \frac{1}{\beta}(1 - 1/Pe)$ with $Pe > 1$ (If $Pe \leq 1$, there is no transient phase) for the energy case. Thus, it is true that one can still achieve an earlier onset of the long-term mixing behaviour by choosing a smaller Pe. However, an earlier onset time is accompanied by a slower long-term mixing rate. As for choosing a large Pe, the onset time is bounded above by $\frac{1}{\beta}$ and results in a faster long-term mixing rate. Thus, it is advantageous to have weak diffusion (large Pe) for mixing in the fixed energy case. This benefit is well illustrated by H^{-1} norm in figure 3.2. Notice that the mixing rate is initially slow for $Pe = 512$ but then out competes the mixing rate of smaller values of Pe.

3.4 Discussion

The local-in-time optimization results suggest that there is a limiting length scale for passive tracer mixing whenever L^2 flows (either $\|\mathbf{u}\|_{L^2}$ or $\|\nabla\mathbf{u}\|_{L^2}$) are instantaneously optimized to decrease the H^{-1} norm. The bounds derived under both L^∞ constrained flow assumptions did not result in proving this observation, but they did definitively rule out the possibility of perfect mixing in finite time for these L^∞ flow constraints.

We suspect that the bounds obtained for L^∞ flows are not sharp and could be improved further. The L^∞ flow analysis produced a double-exponential lower bound on the H^{-1} norm rather than exponential as possibly expected given the numerical results for local-in-time optimal L^2 flows. The double-exponential bounds arise from the use of exponential upper bounds on the quantity $\frac{\|\nabla\theta\|_{L^2}}{\|\theta\|_{L^2}}$ in time for both L^∞ flow constraints considered. We surmise that in fact $\frac{\|\nabla\theta\|_{L^2}}{\|\theta\|_{L^2}} < C$ (where C is a constant) for all time t as suggested by the numerical results. If this is true generally for the L^∞ flows, then our previous analysis would demonstrate that the H^{-1} norm is bounded below by a single exponential instead of a double exponential.

Note that the pure diffusive case discussed in the introduction can always be employed as a mixing strategy by simply not having a flow field at all ($\mathbf{u} = \mathbf{0}$) provided that the flow intensity constraints are generalized to inequalities such as $\|\mathbf{u}\| \leq UL^{d/2}$ and $\|\nabla\mathbf{u}\| \leq \Gamma L^{d/2}$. This is a valuable strategy if one is content with mixing at a long-term rate of $\kappa k_{\min}^2$ where $k_{\min} = \min\{|\mathbf{k}| : |\hat{\theta}_{\mathbf{k}}(0)| > 0\}$. This may be advised in fact if $k_{\min} > 2\pi/\lambda_B$. This may well be the most optimal strategy. Invoking a flow may cause the lower wave number modes to become 'populated' and therefore may limit the mixing rate. It is important to keep this simple strategy in mind when trying to rigorously prove bounds on the H^{-1} norm. This strategy has an important implication — there does *not* exist a lower bound on the H^{-1} norm of the form $\|\nabla^{-1}\theta\|_{L^2} \geq Ae^{-rt}$ where r is independent of the initial data.

In the next chapter and future work, we will consider the optimal control problem with finite-time optimization to minimize the H^{-1} norm at the end time rather than instantaneously attempting to minimize its decay rate. This might lead to flows that can produce even smaller length scales. In Miles and Doering [33], finite-time optimization was explored in the context of the shell model where it was found that global-in-time and local-in-time optimization appeared to give similar mixing rates. For the shell model, however, the analysis was consistent with computation. In the partial differential equation case, the gap between analysis and computation remains to be closed.

3.5 Conclusion

Our numerical study of local-in-time optimization suggests that there is a limiting length scale, a generalized Batchelor length scale, which in turn determines a long-term mixing "Batchelor rate". In dimensional form, this Batchelor rate was found to be proportional to Γ for the fixed enstrophy case and U^2/κ for the fixed energy case. These rates are consistent with those found in the context of the shell model. Although the Batchelor scale has been a theorized lower bound on the length scales present on turbulent flows, it has not been proven rigorously. We hope this numerical study provides insight and promotes investigation into mathematically proving what conditions are necessary on the flow for a length scale limitation. This is especially important since it plays a crucial role in the achievable mixing rates. Furthermore, we provided numerical evidence that (1), for fixed enstrophy optimal flows, strong diffusion can benefit from an early onset of a long-term mixing rate (where the rate itself however is independent of diffusion strength) while (2), for energy fixed optimal flows, strong diffusion weakens the long-term mixing rate.

CHAPTER IV

Global-in-time optimization

4.1 Introduction

In this chapter, we explore global-in-time or finite-horizon optimization. The objective is to maximize mixing at a prescribed final time as oppose to instantaneously as before. This optimization problem employs the techniques from calculus of variations and optimal control [26, 20, 19, 44, 5, 27]. We will focus primarily on the enstrophy-constrained case but will explicitly mention when analogous results carry over to the energy-constrained problem.

This chapter is organized as follows. We introduce the theory and setup of the optimization problem in Section 4.2. This section introduces the associated Euler-Lagrange equations and total variation. Section 4.3 describes two numerical methods for solving the Euler-Lagrange equations. Lastly, analytical and numerical results are presented in Section 4.4 for the pure advective case ($\kappa = 0$).

This project is currently ongoing and the work done so far is presented.

4.2 Theory

4.2.1 The optimal control problem

Here we describe the global-in-time optimization problem for enstrophy-constrained flows. Let $D = [0, L]^d$ be our domain where L is the side length and d is the total number of spatial dimensions. All functions defined on D have periodic boundary conditions. We are interested in the following optimization problem:

$$\min_{\mathbf{u}} \|\theta(\,\cdot\,, T)\|^2_{H^{-1}} \tag{4.1}$$

subject to the constraints

$$\partial_t\theta + \mathbf{u}\cdot\nabla\theta = \kappa\Delta\theta \tag{4.2}$$

with

$$\nabla\cdot\mathbf{u} = 0 \tag{4.3}$$

and a *time-averaged* enstrophy constraint

$$\frac{1}{T}\int\limits_0^T\int d^dxdt|\nabla\mathbf{u}|^2 = \Gamma^2L^d. \tag{4.4}$$

In addition, we are provided with initial data

$$\theta(\mathbf{x}, 0) = \theta_0(\mathbf{x}). \tag{4.5}$$

4.2.2 First and total variation for enstrophy-constraint

Calculus of variations provides the appropriate framework for investigating the conditions placed on optimizers of functionals. Here we present the first and total variation results for the enstrophy-constrained case with diffusion. Prior to presenting these results, it is useful to make the following definitions:

- **Definition** The pair of functions $\{\theta^*(x,t), \mathbf{u}^*(x,t)\}$ on D is said to be **admissible** if it satisfies the following constraints

$$\partial_t \theta + \mathbf{u} \cdot \nabla \theta = \kappa \Delta \theta$$

$$\nabla \cdot \mathbf{u} = 0$$

$$\int_0^T \int_D d^d x dt |\nabla \mathbf{u}|^2 = \Gamma^2 L^d T$$

  and the initial data $\theta(\mathbf{x}, 0) = \theta_0(\mathbf{x})$.

- **Definition** An admissible pair $\{\theta^*(\mathbf{x},t), \mathbf{u}^*(\mathbf{x},t)\}$ on D is said to be an **optimal solution**, **minimizer**, or **minima** of the cost functional C if the **total variation**

$$\Delta C = C\{\theta, u\} - C\{\theta^*, u^*\}$$

  is non-negative for all admissible pairs $\{\theta, \mathbf{u}\}$.

- **Definition** A pair $\{\theta(\mathbf{x},t), \mathbf{u}(\mathbf{x},t)\}$ on D is said to be an **extrema** for the cost functional C if the pair is admissible and the **first variation**

$$\delta C = \lim_{\epsilon \to 0} \frac{C\{\theta + \epsilon\tilde{\theta}, \mathbf{u} + \epsilon\tilde{\mathbf{u}}\} - C\{\theta, \mathbf{u}\}}{\epsilon}$$

  vanishes.

It remains to be shown that a minimizer exists within the constrained set of velocity fields satisfying incompressibility and in the H^1 Sobolev space (required by the enstrophy constraint). Is this a sufficient restriction to ensure that a minimizer exists? Proving the existence of a minimizer typically requires demonstrating weakly lower semicontinuity of the cost functional for a sufficiently restricted set of velocity fields. Furthermore, it remains to be shown that a minimizer is an extrema and thus must satisfy the Euler-Lagrange equations which require sufficient regularity of the cost functional. In the context of traditional calculus, this is analogous to ensuring that a minimizer of a function coincides with a point where the derivative is zero. Nevertheless, it is still worthwhile to solve the Euler-Lagrange equations for candidate solutions. To determine the Euler-Lagrange equations, we introduce the associated augmented Lagrangian

$$\mathcal{L} = \frac{1}{2}\int\limits_D d^d x |\nabla\Delta^{-1}\theta(\mathbf{x},T)|^2 + \int\limits_D d^d x \phi_0(\theta(\mathbf{x},0) - \theta_0(\mathbf{x}))$$
$$+ \int\limits_D d^d x \int dt \left\{\phi(\partial_t\theta + \mathbf{u}\cdot\nabla\theta - \kappa\Delta\theta) + \frac{\mu}{2}(|\nabla\times\mathbf{u}|^2 - \Gamma^2) + p(\nabla\cdot\mathbf{u})\right\} \quad (4.6)$$

where ϕ_0, ϕ, μ, and p are Lagrange multipliers introduced to enforce the system constraints. Assuming that a minimizer is an extrema, we find that a minimizer must satisfy the Euler-Lagrange equations:

$$\frac{\delta\mathcal{L}}{\delta\theta(T)} = 0 \quad \Rightarrow \quad \Delta^{-1}\theta(\mathbf{x},T) - \phi(\mathbf{x},T) = 0 \qquad (4.7\text{a})$$

$$\frac{\delta\mathcal{L}}{\delta\theta} = 0 \quad \Rightarrow \quad \partial_t\phi + \mathbf{u}\cdot\nabla\phi + \kappa\Delta\phi = 0 \qquad (4.7\text{b})$$

$$\frac{\delta\mathcal{L}}{\delta u} = 0 \quad \Rightarrow \quad \phi\nabla\theta - \nabla p - \mu\Delta\mathbf{u} = 0. \qquad (4.7\text{c})$$

$$\frac{\delta\mathcal{L}}{\delta\phi} = 0 \quad \Rightarrow \quad \partial_t\theta + \mathbf{u}\cdot\nabla\theta - \kappa\Delta\theta = 0 \qquad (4.7\text{d})$$

$$\frac{\delta \mathcal{L}}{\delta p} = 0 \quad \Rightarrow \quad \nabla \cdot \mathbf{u} = 0 \tag{4.7e}$$

$$\frac{\delta \mathcal{L}}{\delta \mu} = 0 \quad \Rightarrow \quad \int\limits_0^T \int d^d x dt |\nabla \times \mathbf{u}|^2 - \Gamma^2 L^d T = 0 \tag{4.7f}$$

$$\frac{\delta \mathcal{L}}{\delta \phi_0} = 0 \quad \Rightarrow \quad \theta(\mathbf{x}, 0) - \theta_0(\mathbf{x}) = 0. \tag{4.7g}$$

Note that equation 4.7b has the same analytic and numerical challenges as the backwards heat equation given the sign of diffusion term. As a consequence, this equation will be solved backwards in time in our numerical schemes presented in a later section.

We highlight that 4.7 provides necessary, but not sufficient, conditions for an optimal solution. To prove optimality, it suffices to show the total variation at an extrema (see Appendix C.1 for calculation details)

$$\Delta C \;\; = \;\; \iint \left\{ \nabla \phi \cdot \delta \mathbf{u} \delta \theta + \mu |\nabla \delta \mathbf{u}|^2 \right\} d\mathbf{x} dt + \int\limits_D |\nabla^{-1} \delta \theta_T|^2 d\mathbf{x} \geq 0 \tag{4.8}$$

for all perturbations $\delta\theta = \tilde{\theta} - \theta$ and $\delta \mathbf{u} = \tilde{\mathbf{u}} - \mathbf{u}$ about the candidate solution $\{\theta, \mathbf{u}\}$ where $\{\tilde{\theta}, \tilde{\mathbf{u}}\}$ is admissible. That said, this is a non-convex optimization problem so this is not a trivial task.

4.2.3 First and total variation for energy constraint

The analysis for the energy-constrained problem with $\frac{1}{T} \int_0^T \|\mathbf{u}\|_{L^2}^2 = U^2 L^2$ is similar to that for the enstrophy-constrained problem. Here we simply state the analogous results. The augmented Lagrangian for energy constrained problem is

$$\mathcal{L} = \frac{1}{2}\int\limits_D d^d x|\nabla\Delta^{-1}\theta(\mathbf{x},T)|^2 + \int\limits_D d^d x\phi_0(\theta(\mathbf{x},0)-\theta_0(\mathbf{x}))$$
$$+\int\limits_D d^d x\int dt\left\{\phi(\partial_t\theta+\mathbf{u}\cdot\nabla\theta)+\frac{\mu}{2}(|\mathbf{u}|^2-U^2)+p(\nabla\cdot\mathbf{u})\right\}. \quad (4.9)$$

The Euler-Lagrange equations are

$$\frac{\delta\mathcal{L}}{\delta\theta(T)}=0 \quad\Rightarrow\quad \Delta^{-1}\theta(\mathbf{x},T)-\phi(\mathbf{x},T)=0 \quad (4.10a)$$

$$\frac{\delta\mathcal{L}}{\delta\theta}=0 \quad\Rightarrow\quad \partial_t\phi+\mathbf{u}\cdot\nabla\phi+\kappa\Delta\phi=0 \quad (4.10b)$$

$$\frac{\delta\mathcal{L}}{\delta u}=0 \quad\Rightarrow\quad \phi\nabla\theta-\nabla p+\mu\mathbf{u}=0. \quad (4.10c)$$

$$\frac{\delta\mathcal{L}}{\delta\phi}=0 \quad\Rightarrow\quad \partial_t\theta+\mathbf{u}\cdot\nabla\theta-\kappa\Delta\theta=0 \quad (4.10d)$$

$$\frac{\delta\mathcal{L}}{\delta p}=0 \quad\Rightarrow\quad \nabla\cdot\mathbf{u}=0 \quad (4.10e)$$

$$\frac{\delta\mathcal{L}}{\delta\mu}=0 \quad\Rightarrow\quad \int\limits_0^T\int d^d x dt|\mathbf{u}|^2-U^2L^dT=0 \quad (4.10f)$$

$$\frac{\delta\mathcal{L}}{\delta\phi_0}=0 \quad\Rightarrow\quad \theta(\mathbf{x},0)-\theta_0(\mathbf{x})=0 \quad (4.10g)$$

and the total variation at an extrema is

$$\Delta C \;=\; \iint\left\{\nabla\phi\cdot\delta\mathbf{u}\delta\theta+\mu|\delta\mathbf{u}|^2\right\}d\mathbf{x}dt+\int\limits_D|\nabla^{-1}\delta\theta_T|^2 d\mathbf{x} \quad (4.11)$$

for all perturbations $\delta\theta=\tilde{\theta}-\theta$ and $\delta\mathbf{u}=\tilde{\mathbf{u}}-\mathbf{u}$ about the candidate solution $\{\theta,\mathbf{u}\}$ where $\{\tilde{\theta},\tilde{\mathbf{u}}\}$ is admissible.

4.2.4 Optimal control problem with inequality constraints

One may be interested in the following *inequality* constraint on $\mathbf{u}$ in place of equations (4.4):

$$\int\limits_0^T \int d^d x dt |\nabla \times \mathbf{u}|^2 \leq \Gamma^2 L^d T \tag{4.12}$$

This formulation allows for the possibility of not consuming the entire stirring budget. Intuitively, this seems an unlikely scenario since it seems inefficient not to use of one's stirring resources. We provide the following argument that the inequality constraint is likely saturated for $\kappa = 0$ but not necessarily for $\kappa \neq 0$.

Consider the enstrophy constraint with $\kappa = 0$. Suppose $\{\theta^*, u^*\}$ are minima where (4.12) is not saturated and $\int_0^T \int d^d x dt |\nabla \times \mathbf{u}|^2 = m\Gamma^2 L^d T$ with $0 \leq m < 1$. Then one can construct new variables $\tilde{\theta}, \tilde{\mathbf{u}}$ as $\tilde{\theta}(\mathbf{x}, t) = \theta^*(\mathbf{x}, ct)$ and $\tilde{\mathbf{u}}(\mathbf{x}, t) = cu^*(\mathbf{x}, ct)$ defined for $t \in [0, T/c]$. Any intermediate c $(1 < c < \frac{1}{m})$ will produce the original final-time mix-norm value but at an earlier time $t = T/c$ with available budget $(1 - c^2 m^2)\Gamma^2 L^d T$ left over. Therefore, if there exists *any* $\mathbf{u}$ defined for the remaining time $(t \in [T/c, T])$ that is able to decrease the mix-norm by even the slightest amount with budget remaining, then this new candidate solution ($\tilde{\mathbf{u}}$ for $t \in [0, T/c]$ and $\mathbf{u}$ for $t \in [T/c, T]$) defeats the supposed minima $\{\theta^*, \mathbf{u}^*\}$. The instantaneous optimal strategy from the previous chapter is a good candidate, however there are cases where the first derivative of the mix-norm can not be controlled. This case must be dealt with to make this argument rigorous.

Nevertheless, it still remains worthwhile to formulate the inequality-constrained optimal control problem. We may have confidence that the bound is saturated in the $\kappa = 0$ case (given the argument above), but it not obvious that it will always be saturated with diffusion. In fact, there are trivial cases where $\mathbf{u} = 0$ is a reasonable choice. Consider the enstrophy case with initial data $\theta(\mathbf{x}, 0)$ given by a Fourier mode with large wavenumber k_0 and with a corresponding wavelength that is much smaller than the Batchelor wavenumber $k_B = \sqrt{\Gamma/\kappa}$. If $\mathbf{u} = 0$, then the advection-diffusion equation becomes simply the diffusion equation, and the mix-norm exhibits exponential decay with rate $-\kappa k_0^2$. However from the local-in-time optimization study of the previous chapter, we found that when advection and diffusion are both actively attempting to minimize the mix-norm, the scalar field develops length scales comparable to the Batchelor scale in the long run. Therefore, a rough approximation of the long-term exponential rate is likely to be $-\kappa k_B^2 = -\Gamma$. Therefore in the case were $\Gamma < \kappa k_0^2$, then $\mathbf{u} = 0$ may be a reasonable choice. In other words, stirring (when $\mathbf{u} \neq 0$) may move some spectral mass to wave numbers less than the Batchelor wavenumber and be disruptive. This argument gives reason to believe that the optimal strategy may not always benefit from saturation of the budget.

4.3 Numerical method for pure advection ($\kappa = 0$)

4.3.1 Gradient descent algorithm

Here we describe a numerical method for solving the Euler-Lagrange equations (4.7a) – (4.7g) corresponding the enstrophy-contained problem in 2 dimensions. We use a gradient-based method with line search. The overall strategy is to solve (4.7a) – (4.7g) per iteration except (4.7c). The left hand side of (4.7c) is in fact the gradient of the cost with respect to the velocity field over space and time that is valuable for our iterative update scheme on $\mathbf{u}$. To see that this is the gradient, consider the first variation of the entire augmented functional. If all first variations vanish except the variation with respect to $\mathbf{u}$, then

$$\delta\mathcal{L} \;=\; \iint (\phi\nabla\theta - \nabla p - \mu\Delta\mathbf{u}) \cdot \delta\mathbf{u}\; d\mathbf{x}dt.$$

If in addition $\theta + \delta\theta, u + \delta u$ remains admissible, then

$$\frac{\delta C}{\delta\mathbf{u}} = \frac{\delta L}{\delta\mathbf{u}} = \phi\nabla\theta - \nabla p - \mu\Delta\mathbf{u} \tag{4.13}$$

for variations respecting the constraints.

We discretize space as an N by N grid and time from 0 to T uniformly with M time points. Thus, $\mathbf{u}$ is an array of shape $(M, 2, N, N)$. θ and ϕ are arrays of shape (M, N, N). The approach requires satisfying all the Euler-Lagrange equations except (4.7c) per iteration. The update is of the following form:

$$\mathbf{u}^{k+1} = \mathbb{N}\left(\mathbf{u}^k - \eta\frac{\delta C}{\delta\mathbf{u}}(\mathbf{u}^k)\right). \tag{4.14}$$

where η is the step size. Note that this updates the velocity field at all points of space and time. $\mathbb{N}(\mathbf{v})$ enforces the enstrophy constraint: it is defined as $\mathbb{N}(\mathbf{v}) = \alpha\mathbf{v}$ where α is a normalizing factor and chosen so $\frac{1}{T}\int_0^T dt\|\nabla\mathbb{N}(\mathbf{v})(\,\cdot\,,t)\|_2^2 = \Gamma^2L^2$.

To calculate $\frac{\delta C}{\delta\mathbf{u}}(\mathbf{u}^k) = \phi^k\nabla\theta^k - \nabla p^k - \mu^k\Delta\mathbf{u}^k$, we must find θ^k, ϕ^k, μ^k and p^k. θ^k is determined at each iteration from $\mathbf{u}^k$ by integrating the advection-diffusion equation with initial condition θ_0. Then the terminal condition $\phi^k(\mathbf{x},T) = \Delta^{-1}\theta^k(\mathbf{x},T)$ is used to provide an 'initial condition' for ϕ^k. The velocity $\mathbf{u}^k$ is used to evolve adjoint equation for ϕ^k backwards in time given $\phi^k(\mathbf{x},T)$. Both integrations forward and backwards in time are done with a Fourier basis with N modes to represent the spatial domain and a 2nd order Heun's time-stepping method. N is chosen sufficiently large to to resolve spatially for all time. dt is chosen small enough to satisfy both advective and diffusive CFL conditions. This criteria is compactly stated as $dt = 0.25 \min[1/(\Gamma N), L^2/(\kappa N^2)]$. If we demand that the updated $\mathbf{u}^{k+1}$ be incompressible, then this requires that $\frac{\delta C}{\delta \mathbf{u}}(\mathbf{u}^k)$ be incompressible. Towards this goal, we take the divergence of both sides of (4.13) and require that it vanish. This gives the following choice of $p^k = \Delta^{-1}\nabla \cdot (\phi^k \nabla \theta^k)$ which is equivalent to applying the divergence-free projection operator $\mathbb{P}$ to $\frac{\delta C}{\delta \mathbf{u}}(\mathbf{u}^k)$ where the projector $\mathbb{P}(\mathbf{v})$ is defined as $\mathbb{P}(\mathbf{v}) = \mathbf{v} - \nabla\Delta^{-1}(\nabla \cdot \mathbf{v})$. Lastly μ^k is chosen so that the update obeys the enstrophy constraint. This restriction is captured in the condition $\int_0^T \int_D \Delta \mathbf{u} \cdot \delta \mathbf{u}^k d\mathbf{x}dt = 0$ where $\delta \mathbf{u} = \mathbf{u}^{k+1} - \mathbf{u}^k$. According to our update and for small $\delta \mathbf{u}$ or equivalently η, this is approximately given by $\delta \mathbf{u} \approx \eta \frac{\delta C}{\delta \mathbf{u}}(\mathbf{u}^k)$. We then calculate:

$$\begin{aligned}
0 &= \int_0^T \int_D d\mathbf{x}dt\, \Delta \mathbf{u}^k \cdot \delta \mathbf{u} && (4.15)\\
&= \eta \int_0^T \int_D d\mathbf{x}dt\, \Delta \mathbf{u}^k \cdot (\phi^k \nabla \theta^k - \nabla p^k - \mu^k \Delta \mathbf{u}^k) && (4.16)\\
&= \eta \int_0^T \int_D d\mathbf{x}dt\, \Delta \mathbf{u}^k \cdot (\mathbb{P}(\phi^k \nabla \theta^k) - \mu^k \Delta \mathbf{u}^k) && (4.17)\\
& && (4.18)
\end{aligned}$$

Therefore, we find that

$$\mu^k = \frac{\int_0^T \int_D d\mathbf{x}dt\, \Delta \mathbf{u}^k \cdot \mathbb{P}(\phi^k \nabla \theta^k)}{\int_0^T \int_D d\mathbf{x}dt\, \Delta \mathbf{u}^k \cdot \Delta \mathbf{u}^k}. \quad (4.19)$$

This choice μ^k is chosen to enforce the enstrophy constraint. Note that (4.15) is the *linearized* constraint condition. This is why we enforced this constraint by introducing the normalizing operator $\mathbb{N}()$. The method as a whole is summarized in Figure 4.1.

We perform the following validation of the gradient by considering its projection onto the search direction $\mathbf{d} = -\frac{\delta C}{\delta \mathbf{u}}$ at $\mathbf{u}$ (chosen to be local-in-time velocity field). We first computing the directional analytical gradient by

$$G_a = \frac{\int_0^T \int_D d\mathbf{x}dt \, \frac{\delta C}{\delta \mathbf{u}} \cdot \mathbf{d}}{\int_0^T \int_D d\mathbf{x}dt \, \mathbf{d} \cdot \mathbf{d}}. \tag{4.20}$$

and the finite-difference approximation with step ϵ given by

$$G_\epsilon = \frac{C(\mathbf{u} + \epsilon \mathbf{d}) - C(\mathbf{u} - \epsilon \mathbf{d})}{2\epsilon} \tag{4.21}$$

We can then compare the analytic gradient G_a and its approximation G_ϵ to find the relative error R defined as

$$R = \frac{|G_a - G_\epsilon|}{|G_a|} \tag{4.22}$$

as function of ϵ as shown in figure 4.2. Note that we get a decay in the relative error with smaller ϵ which eventually plateau to $R \approx 10^{-3}$ which is likely due to truncation error associated with our choices of N and M. If N and M are increased, we would expect this saturated plateau to decrease. Improvements to determining an accurate gradient will be investigated further since it is important to have an accurate gradient for an efficient gradient descent search.

4.3.2 Target algorithm

The previous gradient descent algorithm picks the search direction d to be proportional to the negative gradient. Here we purpose a different search direction that has improved convergence at least demonstrated in practice for the problem

```
function GRADIENT_DESCENT($u_0$,$\theta_0$,$tol$)
    $u \leftarrow$ zeros array of shape (M,2,N,N)
    $\theta \leftarrow$ zeros array of shape (M,N,N)
    $\phi \leftarrow$ zeros array of shape (M,N,N)

    $u[0] \leftarrow u_0$

    while $\|\frac{\delta C}{\delta u}\| \geq tol$ do
        $\theta \leftarrow$ integrate_forward($u$ ,$\theta_0$)
        $\phi[M-1] \leftarrow \Delta^{-1}(\theta[M-1])$
        $\phi \leftarrow$ integrate_backward($u$ ,$\phi[M-1]$)

        $\frac{\delta C}{\delta u} \leftarrow$ compute_gradient($\theta, \phi, u$)
        $d \leftarrow$divergence_free_projection($-\frac{\delta C}{\delta u}$)
        $\eta \leftarrow$ line_search($u$,$d$)
        $u \leftarrow$ normalize($u + \eta d$)
    return $u$
```

Figure 4.1: Gradient descent for final-time optimization

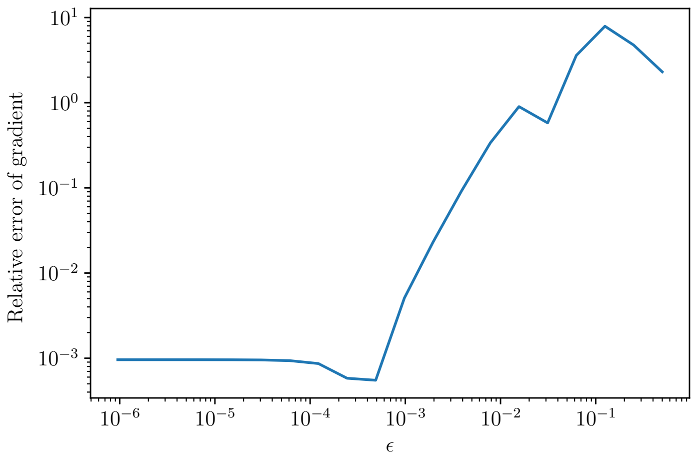

Figure 4.2: The relative numerical error between analytical and finite-difference gradient as a function of the finite-difference step size ϵ.

of interest and parameters chosen. The update is

$$\mathbf{u}^{k+1} = \mathbb{N}(\mathbf{u}^k + \eta \mathbf{d}^k) \tag{4.23}$$

where $\mathbf{d}^k = \mathbf{u}_t^k - \mathbf{u}^k$ and $\mathbf{u}_t^k$ is the 'target' solution given by

$$\mathbf{u}_t^k = \mathbb{N}(\Delta^{-1}\mathbb{P}(\phi^k \nabla \theta^k)). \tag{4.24}$$

ϕ^k and θ^k are determined by integrating forward and backwards in time with $\mathbf{u}^k$ as seen before in the gradient descent algorithm. The above expression can be viewed as the solution to $\phi^k \nabla \theta^k - \nabla p^k - \mu^k \Delta \mathbf{u}_t^k = 0$ where the $\mathbf{u}_t^k$ takes the place of $\mathbf{u}^k$ in the gradient expression while keeping θ^k and ϕ^k as solutions to the state and adjoint equations corresponding to $\mathbf{u}^k$. μ^k and p^k are embodied in the normalization and divergence-free projection operators that require $\mathbf{u}_t^k$ to satisfy incompressibility and the intensity constraint. The algorithm is summarized in figure 4.3.

4.4 Results for pure advection ($\kappa = 0$)

4.4.1 Optimal budget use is uniform in time

Recall that the velocity field is required to have a fixed *mean* enstrophy of Γ over time. Surprisingly, we find that it is optimal to expend enstrophy uniformly

```
**function** TARGET($u_0$,$\theta_0$,$tol$)
    $u \leftarrow$ zeros array of shape (M,2,N,N)
    $\theta \leftarrow$ zeros array of shape (M,N,N)
    $\phi \leftarrow$ zeros array of shape (M,N,N)

    $u[0] \leftarrow u_0$

    **while** $\|d\| \geq tol$ **do**
        $\theta \leftarrow$ integrate_forward($u$ ,$\theta_0$)
        $\phi[M-1] \leftarrow \Delta^{-1}(\theta[M-1])$
        $\phi \leftarrow$ integrate_backward($u$ ,$\phi[M-1]$)

        $u_{target} \leftarrow$ compute_target($\theta, \phi, u$)
        $d \leftarrow u_{target} - u$
        $\eta \leftarrow$ line_search($u$,$d$)
        $u \leftarrow$ normalize($u + \eta d$)
    **return** $u$
```

Figure 4.3: Target algorithm for final-time optimization

in time. This is shown by the calculation using the Euler-Lagrange equations:

$$
\begin{aligned}
\frac{d}{dt}\int\limits_D d^dx|\nabla\times\mathbf{u}|^2 &= -\int\limits_D d^dx\Delta\frac{\partial\mathbf{u}}{\partial t}\cdot\mathbf{u} \\
&= -\frac{1}{\mu}\int\limits_D d^dx\left(\frac{\partial\phi}{\partial t}\nabla\theta+\phi\nabla\frac{\partial\theta}{\partial t}-\nabla\frac{\partial p}{\partial t}\right)\cdot\mathbf{u} \\
&= -\frac{1}{\mu}\int\limits_D d^dx\left(\frac{\partial\phi}{\partial t}\nabla\theta+\phi\nabla\frac{\partial\theta}{\partial t}\right)\cdot\mathbf{u} \\
&= -\frac{1}{\mu}\int\limits_D d^dx\left(\frac{\partial\phi}{\partial t}\mathbf{u}\cdot\nabla\theta-\mathbf{u}\cdot\nabla\phi\frac{\partial\theta}{\partial t}\right) \\
&= 0
\end{aligned}
$$

Thus, enstrophy is utilized uniformly in time for a minimizer to problem (4.1) (provided that the minimizer satisfies the Euler-Lagrange equations). This shows another commonality between the partial differential equation and shell model.

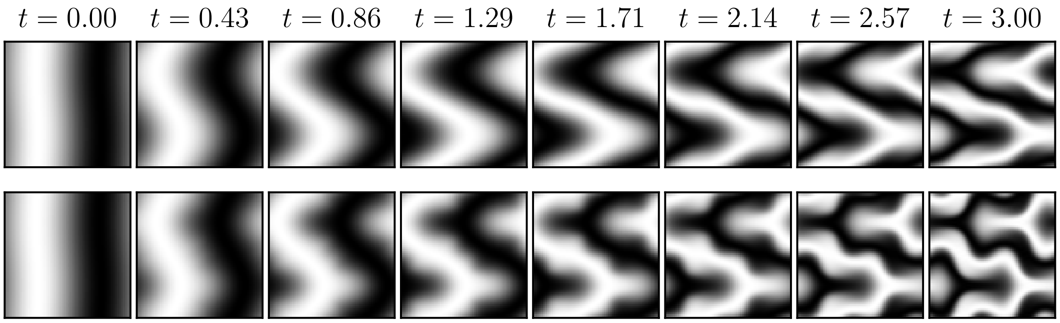

Figure 4.4: The top filmstrip is the local-in-time strategy while the bottom filmstrip is the global-in-time strategy. $\Gamma = 1.0$.

A similar calculation reveals that energy is conserved in time for the energy-constrained problem.

4.4.2 Comparison with local-in-time optimization

We investigate the performance of global-in-time optimization relative to instantaneous optimization for $\Gamma = 1.0$, $\kappa = 0$, $L = 1.0$, $M = 1000$, $N = 64$ and $T = 3.0$. We consider the performance of mixing the initial condition $\theta_0(\mathbf{x}) = \sin(2\pi x/L)$. We use the numerical scheme described in section 4.3.2. Python code is provided in Appendix C.2. The resulting flow is shown in the bottom filmstrip of Figure 4.4 while the local-in-time flow is shown in the top filmstrip for comparison. Quantities of interests for this global-in-time optimal flow are shown in Figure 4.5. The top subplot of Figure 4.5 shows how the H^{-1} mix-norm varies in time. Note how initially the local-in-time flow outperforms the global-in-time optimal flow for short times, obviously due to the fact that it is a greedy algorithm maximizing mixing in the near future. However, note that the global-in-time flow eventually outperforms local-in-time by the final time as expected. Furthermore, observe that enstrophy is expended uniformly in time as

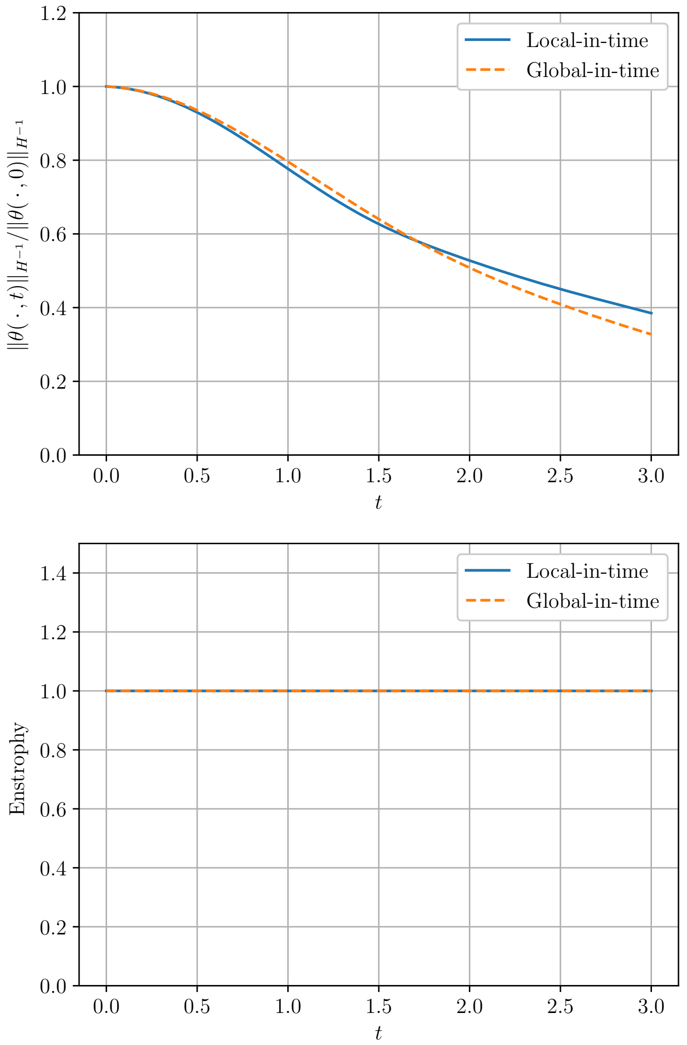

Figure 4.5: The top subplot shows a comparison of local-in-time agains global-in-time optimization for fixed enstrophy ($\Gamma = 1.0$). The bottoms subplot shows uniform expenditure in time of the stirring budget as expected.

expected by the previous analytical result.

CHAPTER V

Conclusion and Future Directions

In this dissertation, new results on the optimization of mixing are uncovered. In the first study on optimization of a shell model, it is discovered that the mixing rate is limited by the presence of diffusion. We investigated both local- and global-in-time optimization for various shell model truncations. The 3-shell model was particularly informative since analytical solutions for Euler-Lagrange equations were obtained by using methods familiar to the theory of nuclear magnetic resonance. These analyses demonstrated clearly how the global-in-time optimization strategy can outperform the local-in-time optimization scheme by a clever 'rotation' in state space.

In the local-in-time optimization study of the advection-diffusion equation, it is demonstrated numerically that a generalized Batchelor length scale places restrictions on the rate of mixing. Many other observations from the shell model also carried over to this setting. For the enstrophy constrained problem, the long-term mixing rate is shown to be independent of diffusion coefficient. For the energy-constrained problem, we found that the mixing rate was dependent on diffusion coefficient in a perhaps surprising way — increased diffusion can *decrease* the mixing rate as measured by the H^{-1} norm. Diffusion is usually thought to benefit mixing as it tends to homogenize dye throughout a fluid, so its detrimental effect on the process is unexpected.

Finally, results are presented on our ongoing project on global-in-time optimization. In this study, we found that it is optimal to use the stirring budget uniformly in time when we only demanded that the *time-averaged* enstrophy be a desired fixed value. We also found this to be the case for the energy-constrained problem. This is consistent with the work of G. Mathew *et al.* [32] that found a uniform use of the stirring budget when controlling a superposition of a restricted set of enstrophy and energy constrained flows. We also presented the global-in-time strategy for a short time period which (as expected) outperformed the local-in-time strategy at the end time. However, the improvement is not dramatic in this short time period.

Global-in-time optimization is computationally challenging due to the large dimensionality of the search space. Improvements are being considered such as employing other gradient descent and line search methods. The primary bottleneck in the algorithm is the computation of gradient with respect to $\mathbf{u}$ (and equally true for the computation of the 'target' velocity field) which requires time integration forwards and backwards in time. Methods for approximating this gradient would be valuable to speed up the computation, especially at earlier iterations where precision is not necessary until near the convergence point. The inclusion of diffusion will most likely require a formulation with inequality intensity constraints since it is not obvious that the intensity budget will be saturated given the argument of the previous chapter. In terms of future theoretical and analytic work on global-in-time optimization, it is still important to determine the existence and uniqueness of optimizers for the presented problem. What function space restriction is necessary to ensure that a minimizer exists?

These studies suggest the following questions to the mixing community:

- Can one demonstrate the mixing rate limitation of the Batchelor scale rigorously? What are the restrictions on the control $\mathbf{u}$ for this limitation to hold? Can one construct a flow that surpasses the Batchelor scale in the long-run?

- Related to the previous question, can one derive a single exponential lower bound of the H^{-1} norm with diffusion for energy or enstrophy stirring intensity constraints? Without diffusion, the rate is shown to depend on the support of the initial condition [24]. How does diffusion affect this dependence on the initial condition?

It is natural to ask "Are optimal flows as defined feasible?" and "How would one generate such flows in reality?" The purpose of this study is not to tackle these questions directly since our formulation is not entirely suitable for these questions. The purpose of this study is to consider idealized mixing to provide expectations in the best-case scenario with absolute control over the velocity field under the assigned constraints. In reality absolute control is generally not obtainable.

Although we do not fully address feasibility in the series of studies presented here, we acknowledge that feasibility is an important issue and encourage research in this direction. With the goal of feasibility in mind, we can work backwards from a desired flow $\mathbf{u}$ to obtain the required forcing $\mathbf{f}$ on a fluid. This can be found by simply substituting a discovered (local- or global-in-time) optimal velocity field $\mathbf{u}$ into the Navier-Stokes equation to find:

$$\mathbf{f} = \rho\partial_t\mathbf{u} + \rho\mathbf{u}\cdot\nabla\mathbf{u} + \nabla p - \mu\Delta\mathbf{u} \tag{5.1}$$

where ρ is the fluid density, μ is the viscosity, p is the fluid pressure, and $\mathbf{f}$ is the required forcing. If $\mathbf{f}$ and the initial condition $\mathbf{v}_0(\mathbf{x}) = \mathbf{u}(\mathbf{x}, 0)$ are given, the solution $\mathbf{v}$ to the Navier-Stokes equation, $\rho\partial_t\mathbf{v} + \rho\mathbf{v} \cdot \nabla\mathbf{v} = -\nabla p + \mu\Delta\mathbf{v} + \mathbf{f}$, is precisely the desired flow field: $\mathbf{v} = \mathbf{u}(\mathbf{x}, t)$. The next natural question is "How could you construct a mixing device to create the forcing $\mathbf{f}$?" Although we do not provide an answer, the derived forcing $\mathbf{f}$ at least gives us a target to aim for when tasked with the engineering problem of designing a mechanical mixer that realizes the flow $\mathbf{u}$.

The required mechanical power to operate a mixing device is also useful measure for evaluating feasibility. For instance if the mechanical power blows up in finite time, this would rule out its feasibility. The mechanical power P expended by an agent exerting the force $\mathbf{f}$ on the flow can be found by multiplying (5.1) by $\mathbf{u}$ and integrating over the domain D to arrive at

$$P = \int\limits_D \mathbf{f} \cdot \mathbf{u} \, d^d\mathbf{x} = \frac{d}{dt}\left(\frac{\rho}{2}\int\limits_D |\mathbf{u}|^2 \, d^d\mathbf{x}\right) + \mu\int\limits_D |\nabla\mathbf{u}|^2 \, d^d\mathbf{x} \tag{5.2}$$

Note that, for the enstrophy-constrained case, the last term on the right-hand side of (5.2) is constant. Thus the required mechanical power changes in time according to the rate of change of the total kinetic energy. For the energy-constrained case, the first term on the right-hand side of (5.2) vanishes. Therefore the mechanical power is proportional to the enstrophy of the flow which could potentially increase dramatically due to the development of small length scales. Note for inviscid flows the power in the energy case is zero.

APPENDICES

APPENDIX A

Shell model

A.1 Lower bound for non-diffusive enstrophy-constrained case

Recall equation (2.6) from section 2.3:

$$\frac{d}{dt}\|\theta(t)\|_{h^{-1}}^2 = 2\sum_{n=1}\left(k_{n+1}^{-2} - k_n^{-2}\right)\theta_n\theta_{n+1}k_n u_n \tag{A.1}$$

using $\kappa = 0$ since we are considering the non-diffusive case. Rewrite (A.1) as

$$\frac{d}{dt}\|\theta(t)\|_{h^{-1}}^2 = v \cdot w \tag{A.2}$$

where v and w are infinite-dimensional vectors with components $v_n = k_n u_n$ and $w_n = 2\left(k_{n+1}^{-2} - k_n^{-2}\right)\theta_n\theta_{n+1}$ respectively. Using Cauchy-Schwarz and $\|v\|_{l^2} = \|u\|_{h^1}$, we have

$$\frac{d}{dt}\|\theta(t)\|_{h^{-1}}^2 \geq -\|u\|_{h^1}\|w\|_{l^2}. \tag{A.3}$$

$\|w\|_{l^2}$ and $\|\theta\|_{h^{-1}}$ are related by the following estimates and manipulations:

$$\begin{aligned}
\|w\|_{l^2} &= \sqrt{4\sum_{n=1}(k_n^{-2} - k_{n+1}^{-2})^2\theta_n^2\theta_{n+1}^2} && \text{(A.4)}\\
&= \sqrt{9\sum_{n=1} k_n^{-2}k_{n+1}^{-2}\theta_n^2\theta_{n+1}^2} \text{ (using } k_{n+1} = 2k_n) && \text{(A.5)}\\
&= 3\sqrt{\sum_{n=1}(\theta_n^2 k_n^{-2})(\theta_{n+1}^2 k_{n+1}^{-2})} && \text{(A.6)}\\
&\leq 3\sqrt{\sum_{n=1}(\theta_n^2 k_n^{-2})\sum_{m=1}(\theta_m^2 k_m^{-2})} && \text{(A.7)}\\
&= 3\sqrt{\|\theta\|_{h^{-1}}^2\|\theta\|_{h^{-1}}^2} && \text{(A.8)}\\
&= 3\|\theta\|_{h^{-1}}^2 && \text{(A.9)}
\end{aligned}$$

Using this with (A.3), we conclude

$$\frac{d}{dt}\|\theta\|_{h^{-1}}^2 \geq -3\|u\|_{h^1}\|\theta\|_{h^{-1}}^2. \tag{A.10}$$

If $\|u(t)\|_{h^1}$ is in $L^1([0,T])$, then we can use Grönwall's inequality to deduce

$$\|\theta(t)\|_{h^{-1}} \geq \|\theta(0)\|_{h^{-1}} \exp\left(-\frac{3}{2}\int_0^t \|u(t')\|_{h^1}dt'\right). \tag{A.11}$$

A.2 Lower bound for non-diffusive energy-constrained case

Now we choose to represent equation (2.6) with $\kappa = 0$ in the following alternative form:

$$\frac{d}{dt}\|\theta(t)\|_{h^{-1}}^2 = u \cdot y \tag{A.12}$$

where y is an infinite-dimensional vectors with components

$$y_n = 2k_n \left(k_{n+1}^{-2} - k_n^{-2}\right) \theta_n \theta_{n+1}.$$

Using Cauchy-Schwarz, we find that

$$\frac{d}{dt}\|\theta(t)\|_{h^{-1}}^2 \geq -\|u\|_{l^2}\|y\|_{l^2} \tag{A.13}$$

By similar techniques seen for bounding $\|w\|_{l^2}$ in the previous section, we can relate $\|y\|_{l^2}$ to $\|\theta\|_{h^{-1}}$ and $\|\theta\|_{l^2}$ as follows:

$$\begin{aligned}
\|y\|_{l^2} &= \sqrt{4\sum_{n=1} k_n^2(k_n^{-2} - k_{n+1}^{-2})^2\theta_n^2\theta_{n+1}^2} && \text{(A.14)}\\
&= \sqrt{9\sum_{n=1} k_{n+1}^{-2}\theta_n^2\theta_{n+1}^2}\ \text{(using } k_{n+1} = 2k_n\text{)} && \text{(A.15)}\\
&= 3\sqrt{\sum_{n=1}(\theta_n^2)(\theta_{n+1}^2 k_{n+1}^{-2})} && \text{(A.16)}\\
&\leq 3\sqrt{\sum_{n=1}\theta_n^2\sum_{m=1}\theta_m^2 k_m^{-2}} && \text{(A.17)}\\
&= 3\sqrt{\|\theta\|_{l^2}^2\|\theta\|_{h^{-1}}^2} && \text{(A.18)}\\
&= 3\|\theta\|_{l^2}\|\theta\|_{h^{-1}} && \text{(A.19)}
\end{aligned}$$

Using the relation above with (A.12) and the fact that $\|\theta(t)\|_{l^2} = \|\theta(0)\|_{l^2}$, we find that

$$\frac{d}{dt}\|\theta\|_{h^{-1}} \;\; \geq \;\; -\frac{3}{2}\|u\|_{l^2}\|\theta(0)\|_{l^2}. \tag{A.20}$$

Therefore, we can obtain the following lower bound provided that $\|u(t)\|_{l^2}$ is in $L^1([0,T])$,

$$\|\theta(t)\|_{h^{-1}} \geq \|\theta(0)\|_{h^{-1}} - \frac{3}{2}\|\theta(0)\|_{l^2} \int\limits_0^t \|u(t')\|_{l^2} dt'. \tag{A.21}$$

A.3 Global-in-time optimal strategy requires uniform use of budget in time

It is useful to introduce matrix notation to assist our calculation. Let

$$B^{(n)} = \begin{bmatrix} \ddots & \ddots & \ddots & & & & & \\ & 0 & 0 & 0 & & & & \\ & & 0 & 0 & -k_n & & & \\ & & & k_n & 0 & 0 & & \\ & & & & 0 & 0 & 0 & \\ & & & & & \ddots & \ddots & \ddots \end{bmatrix}, K = \begin{bmatrix} k_1 & & & \\ & k_2 & & \\ & & k_3 & \\ & & & \ddots \end{bmatrix}, \tag{A.22}$$

$$M = K^{-1}K^{-1}, \text{ and } A = \sum_n u_n B^{(n)}. \tag{A.23}$$

With this notation, we can write can write (2.11) with $\kappa = 0$ as

$$M\theta(T) - \phi(T) = 0 \tag{A.24a}$$

$$\dot{\phi} - A\phi = 0 \tag{A.24b}$$

$$\dot{\theta} - A\theta = 0 \tag{A.24c}$$

$$\phi^T B^{(n)}\theta + \mu k_n^{2\alpha} u_n = 0 \tag{A.24d}$$

$$\frac{1}{T}\int\limits_0^T u^T K^{2\alpha} u - W^{(\alpha)} = 0 \tag{A.24e}$$

We calculate that

$$\begin{aligned}
\frac{d}{dt}(u^T K^{2\alpha} u) &= 2\sum_n u_n k_n^{2\alpha} \frac{du_n}{dt} \\
&= -\frac{2}{\mu}\sum_n u_n \left(\phi^T B^{(n)} \frac{d\theta}{dt} + \frac{d\phi^T}{dt} B^{(n)} \theta \right) && \text{(using (A.24d))} \\
&= -\frac{2}{\mu}\sum_n u_n \left(\phi^T B^{(n)} A\theta + \phi^T A^T B^{(n)} \theta \right) && \text{(using (A.24b and A.24c))} \\
&= -\frac{2}{\mu}\sum_n u_n \phi^T \left(B^{(n)} A - A B^{(n)} \right) \theta && (A^T = -A) \\
&= -\frac{2}{\mu}\sum_n u_n \phi^T \left[B^{(n)}, A \right] \theta && \text{([}\cdot,\cdot\text{] is the commutator.)} \\
&= -\frac{2}{\mu}\sum_n \sum_m u_n u_m \phi^T \left[B^{(n)}, B^{(m)} \right] \theta \\
&= 0 && \text{(antisym. w.r.t. } n \text{ \& } m\text{).}
\end{aligned}$$

Thus, $u^T K^{2\alpha} u$ is conserved in time.

A.4 Optimal control solution to 3-shell truncated model

By differentiating (2.11d) and simplifying, we find

$$\frac{d}{dt} k_1 u_1 = -\frac{1}{\mu T}(\phi_1 \theta_3 - \phi_3 \theta_1) k_2 u_2 \tag{A.25}$$

$$\frac{d}{dt} k_2 u_2 = \frac{1}{\mu T}(\phi_1 \theta_3 - \phi_3 \theta_1) k_1 u_1. \tag{A.26}$$

Differentiating the quantity $(\phi_1\theta_3 - \phi_3\theta_1)$ and using (2.11), you can show that

$$\frac{d}{dt}(\phi_1 \theta_3 - \phi_3 \theta_1) = 0. \tag{A.27}$$

Thus, equations (A.25) and (A.26) take the form

$$\frac{d}{dt}k_1u_1 = -\omega k_2 u_2 \tag{A.28}$$

$$\frac{d}{dt}k_2u_2 = \omega k_1 u_1 \tag{A.29}$$

where

$$\omega = \frac{1}{\mu T}(\phi_1\theta_3 - \phi_3\theta_1) \tag{A.30}$$

is a constant. The initial condition $\theta(0) = (1,0,0)^T$ translates into the following initial condition for $u = (\pm\frac{1}{\tau}, 0, 0)$ by making use of equations (2.11d) evaluated at $t = 0$ and the constraint $\frac{1}{T}\int_0^T \|u(t)\|_{h^1}^2 dt = \frac{1}{\tau^2}$. If we choose the u with a positive first component, then we find that

$$k_1u_1 = \frac{1}{\tau}\cos(\omega t) \quad k_2u_2 = \frac{1}{\tau}\sin(\omega t).$$

A.5 State solution to 3-shell truncated model

We solve (2.16) given the optimal control by making a unitary transformation, $\theta^r = \mathcal{U}^r\theta$ where $\mathcal{U}^r = \exp(-\omega t S_y)$. Using this transformation we find a new 'rotated' state equation,

$$\dot{\theta}^r = \mathcal{U}^r A\theta + \dot{\mathcal{U}}^r\theta = [\mathcal{U}^r A(\mathcal{U}^r)^{-1} + \dot{\mathcal{U}}^r(\mathcal{U}^r)^{-1}]\theta^r = \dot{\theta}^r = [\mathcal{U}^r A(\mathcal{U}^r)^{-1} - \omega S_y]\theta^r$$

where

$$\mathcal{U}^r A(\mathcal{U}^r)^{-1} = \exp(-\omega t S_y)\left(\frac{1}{\tau}\sin(\omega t)S_x + \frac{1}{\tau}\cos(\omega t)S_z\right)\exp(\omega t S_y).$$

The time derivative of $\mathcal{U}^r A(\mathcal{U}^r)^{-1}$ is calculated as

$$\begin{aligned}
\frac{d}{dt}\mathcal{U}^r A(\mathcal{U}^r)^{-1} &= \frac{d}{dt}\left(\exp(-\omega t S_y)\left(\frac{1}{\tau}\sin(\omega t)S_x + \frac{1}{\tau}\cos(\omega t)S_z\right) r\exp(\omega t S_y)\right) \\
&= \exp(-\omega t S_y)\Bigg(\frac{1}{\tau}\sin(\omega t)[S_x, \omega S_y] \\
&+ \frac{1}{\tau}\cos(\omega t)[S_z, \omega S_y]\Bigg)\exp(\omega t S_y) \\
&+ \exp(-\omega t S_y)\left(\frac{1}{\tau}\omega\cos(\omega t)S_x - \frac{1}{\tau}\omega\sin(\omega t)S_z\right)\exp(\omega t S_y) \\
&= \exp(-\omega t S_y)\left(\frac{1}{\tau}\omega\sin(\omega t)S_z - \frac{1}{\tau}\omega\cos(\omega t)S_x\right)\exp(\omega t S_y) \\
&+ \exp(-\omega t S_y)\left(\frac{1}{\tau}\omega\cos(\omega t)S_x - \frac{1}{\tau}\omega\sin(\omega t)S_z\right)\exp(\omega t S_y) \\
&= 0.
\end{aligned}$$

Thus, $\mathcal{U}^r A(\mathcal{U}^r)^{-1}$ is constant. Hence, we can evaluate it at any time. If we choose $t = 0$, we find that $\mathcal{U}^r A(\mathcal{U}^r)^{-1} = \frac{1}{\tau}S_z$. The rotated state equation becomes

$$\dot{\theta}^r = \left(\frac{1}{\tau}S_z - \omega S_y\right)\theta^r.$$

Since $\left(\frac{1}{\tau}S_z - \omega S_y\right)$ is time-independent, we can write the solution as

$$\theta^r(t) = \exp\left(-\omega t S_y + \frac{1}{\tau}t S_z\right)\theta^r(0).$$

We can write this in terms of θ to get

$$\theta(t) = \exp(\omega t S_y)\exp(-\omega t S_y + \frac{1}{\tau}t S_z)\theta(0).$$

The rotation of a vector x about an arbitrary axis $\hat{n} = (n_x, n_y, n_z)^T$ by an angle ψ is performed by acting on the vector x with the operator, $\exp(\psi\hat{n} \cdot \mathbf{S})$. If we define

$$Z = \begin{pmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{pmatrix},$$

the expression $\exp(\psi\hat{n} \cdot \mathbf{S})$ is given by [2]

$$\exp(\psi\hat{n} \cdot \mathbf{S}) = I + (\sin\psi)Z + (1 - \cos\psi)Z^2.$$

Using this fact, we rewrite our solution as

$$\theta(\omega, \tau, t) = \begin{pmatrix} \cos(\omega t)\cos(\nu t) + \frac{\omega}{\nu}\sin(\omega t)\sin(\nu t) \\ \frac{\rho}{\nu}\sin(\nu t) \\ -\sin(\omega t)\cos(\nu t) + \frac{\omega}{\nu}\cos(\omega t)\sin(\nu t) \end{pmatrix}$$

where we assume $\theta(0) = (1, 0, 0)^T$ and define $\nu \equiv \sqrt{\omega^2 + \frac{1}{\tau}^2}$.

A.6 Perfect mixing in finite time is impossible with diffusion

Let

$$\frac{d}{dt}\theta_n = k_{n-1}u_{n-1}\theta_{n-1} - k_n u_n \theta_{n+1} - \kappa\, k_n^2 \theta_n, \quad n = 1, 2, \ldots \tag{A.31}$$

and u be constrained by

$$\|u(t)\|_{h^\alpha} = W^{(\alpha)} \tag{A.32}$$

at all times t. We define

$$x_m = \left(\sum_{n=1}^{m} \theta_n^2\right)^{1/2}, \quad y_m = \left(\sum_{n=m+1}^{\infty} \theta_n^2\right)^{1/2}, \quad \text{and } \beta_m = y_m/x_m. \tag{A.33}$$

Also we define

$$\beta_m^{\pm} = \gamma_m \pm \sqrt{\gamma_m^2 - 1}$$

and $\gamma_m = \frac{3\kappa k_m^{1+\alpha}}{2W^{(\alpha)}}$.

Lemma 1:

$$\text{(a) If } \beta_m(0) \leq \beta_m^+ \text{ and } \gamma_m \geq 1 \text{ , then } \beta_m(t) \leq \beta_m^+ \text{ for all time } t. \tag{A.34}$$

$$\text{(b) If } \beta_m(0) \leq \beta_m^- \text{ and } \gamma_m \geq 1 \text{ , then } \beta_m(t) \leq \beta_m^- \text{ for all time } t. \tag{A.35}$$

Proof:

We find that

$$\frac{1}{2}\frac{d}{dt}x_m^2 = -k_m u_m \theta_m \theta_{m+1} - \kappa \sum_{n=1}^{m} k_n^2 \theta_n^2 \tag{A.36a}$$

$$\frac{1}{2}\frac{d}{dt}y_m^2 = k_m u_m \theta_m \theta_{m+1} - \kappa \sum_{n=m+1}^{\infty} k_n^2 \theta_n^2. \tag{A.36b}$$

Using the following inequalities,

- $|\theta_m| \leq x_m$
- $|\theta_{m+1}| \leq y_m$
- $|u_m| \leq \frac{W^{(\alpha)}}{k_m^\alpha}$

- $\sum_{n=1}^{m} k_n^2\theta_n^2 \leq k_m^2 x_m^2$

- $\sum_{n=m+1}^{\infty} k_n^2\theta_n^2 \geq k_{m+1}^2 y_m^2$,

we find that

$$\frac{dx_m}{dt} \geq -k_m^{1-\alpha}W^{(\alpha)}y_m - \kappa k_m^2 x_m \tag{A.37a}$$

$$\frac{dy_m}{dt} \leq k_m^{1-\alpha}W^{(\alpha)}x_m - \kappa k_{m+1}^2 y_m. \tag{A.37b}$$

Taking the derivative of y_m/x_m,

$$\begin{aligned}\frac{d}{dt}\left(\frac{y_m}{x_m}\right) = \frac{x_m y_m' - y_m x_m'}{x_m^2} &\leq \frac{1}{x_m}(k_m^{1-\alpha}W^{(\alpha)}x_m - \kappa k_{m+1}^2 y_m)\\ &- \frac{y_m}{x_m^2}(-k_m^{1-\alpha}W^{(\alpha)}y_m - \kappa k_m^2 x_m)\\ &= k_m^{1-\alpha}W^{(\alpha)}\left(1+\frac{y_m^2}{x_m^2}\right) - \kappa(k_{m+1}^2 - k_m^2)\frac{y_m}{x_m}\end{aligned} \tag{A.38}$$

Using the definition $\beta_m = y_m/x_m$, we write this as

$$\frac{d\beta_m}{dt} \leq k_m^{1-\alpha}W^{(\alpha)}\left(1+\beta_m^2\right) - \kappa(k_{m+1}^2 - k_m^2)\beta_m = k_m^{1-\alpha}W^{(\alpha)}(\beta_m - \beta_m^-)(\beta_m - \beta_m^+) \tag{A.39}$$

where $\beta_m^{\pm}$ are the roots of the right-hand side given by

$$\beta_m^{\pm} = \gamma_m \pm \sqrt{\gamma_m^2 - 1}.$$

The roots $\beta_m^{\pm}$ are real when $\gamma_m = \frac{3\kappa k_m^{1+\alpha}}{2W^{(\alpha)}} \geq 1$. The differential inequality (A.39) implies conditions (A.34) and (A.35).

Theorem 1:

Let there exist a smallest integer p such that $x_p(0) > 0$ and $y_p(0) < \infty$. It follows that perfect mixing in finite time is impossible ($\|\theta\|_{h^{-1}} > 0$ for all finite t). Furthermore, for

$$m = \max\left\{ \left\lceil \frac{1}{1+\alpha} \log_2 \left(\frac{2W^{(\alpha)}}{3\kappa k_0^{1+\alpha}} \gamma^* \right) \right\rceil, p \right\} \tag{A.40}$$

with $\gamma^* \equiv \max\left\{ \frac{1+\beta_p^2(0)}{2\beta_p(0)}, 1 \right\}$ and $\beta_p = x_p/y_p$, we have that

$$\|\theta(t)\|_{h^{-1}} \geq \frac{x_m(0)}{k_m} \exp(-(k_m^{1-\alpha} W^{(\alpha)} \beta_m^+ + \kappa k_m^2)t). \tag{A.41}$$

Proof:

Assume that there exists a smallest integer p such that $x_p(0) > 0$ and $y_p(0) < \infty$. Thus, $\beta_p(0) < \infty$. If we choose $m \geq p$ large enough so that $\gamma_m \geq \gamma^* \equiv \max\left\{ \frac{1+\beta_p^2(0)}{2\beta_p(0)}, 1 \right\}$, then this ensures that $\beta_m(0) \leq \beta_p(0) \leq \beta_m^+$ and the hypotheses of *Lemma 1.a* are satisfied. Therefore, by choosing

$$m = \max\left\{ \left\lceil \frac{1}{1+\alpha} \log_2 \left(\frac{2W^{(\alpha)}}{3\kappa k_0^{1+\alpha}} \gamma^* \right) \right\rceil, p \right\} \tag{A.42}$$

we have that

$$y_m(t) \leq \beta_m^+ x_m(t). \tag{A.43}$$

Note that (A.37a) remains true. We use (A.37a) with the above relation to give

$$\frac{dx_m}{dt} \geq -(k_m^{1-\alpha} W^{(\alpha)} \beta_m^+ + \kappa k_m^2) x_m. \tag{A.44}$$

By Grönwall's inequality, we have that

$$x_m(t) \geq x_m(0) \exp(-(k_m^{1-\alpha} W^{(\alpha)} \beta_m^+ + \kappa k_m^2)t). \tag{A.45}$$

Since

$$\|\theta\|_{h^{-1}}^2 \geq \sum_{n=1}^{m} \frac{\theta_n^2}{k_n^2} \geq \frac{1}{k_m^2} \sum_{n=1}^{m} \theta_n^2 = \frac{x_m^2}{k_m^2}, \tag{A.46}$$

we have that

$$\|\theta(t)\|_{h^{-1}} \geq \frac{x_m(0)}{k_m} \exp(-(k_m^{1-\alpha} W^{(\alpha)} \beta_m^+ + \kappa k_m^2)t) > 0. \tag{A.47}$$

Therefore, perfect mixing in finite time is impossible. The exponential decay rate is exactly equal to the eigenvalue λ_- (see definition (2.28)) from the local-in-time analysis with diffusion.

Theorem 2:

Let p be an integer such that $\theta_q = 0$ for all $q \geq p$. It follows that perfect mixing in finite time is impossible ($\|\theta\|_{h^{-1}} > 0$ for all finite t). Furthermore, for

$$m = \max\left\{ \left\lceil \frac{1}{1+\alpha} \log_2 \left(\frac{2W^{(\alpha)}}{3\kappa k_0^{1+\alpha}} \right) \right\rceil, p \right\} \tag{A.48}$$

we have that

$$\|\theta(t)\|_{h^{-1}} \geq \frac{x_m(0)}{k_m} \exp(-(k_m^{1-\alpha} W^{(\alpha)} \beta_m^- + \kappa k_m^2)t). \tag{A.49}$$

Proof:

Assume there exists an integer p such that $\theta_q(0) = 0$ for all $q \geq p$. Then, we have that $\beta_q(0) = 0$ for all $q \geq p$. By choosing

$$m = \max\left\{ \left\lceil \frac{1}{1+\alpha} \log_2 \left(\frac{2W^{(\alpha)}}{3\kappa k_0^{1+\alpha}} \right) \right\rceil, p \right\} \tag{A.50}$$

we have that $\gamma_m \geq 1$ and $\beta_m(0) = 0 \leq \beta_m^-$. Thus, we satisfy the hypotheses of *Lemma 1.b* and arrive at

$$y_m(t) \leq \beta_m^- x_m(t). \tag{A.51}$$

The remaining argument is identical to the argument from inequality (A.43) forward in the previous proof with the substitution of β_m^+ for β_m^-. We find that

$$\|\theta(t)\|_{h^{-1}} \geq \frac{x_m(0)}{k_m} \exp(-(k_m^{1-\alpha} W^{(\alpha)} \beta_m^- + \kappa k_m^2)t) > 0. \tag{A.52}$$

Therefore, perfect mixing in finite time is impossible. The exponential decay rate is exactly equal to the eigenvalue λ_+ (see definition (2.28)) from the local-in-time analysis with diffusion.

Remark: Note that *Theorem 1* proves perfect mixing in finite time is impossible for a larger set of initial conditions than that considered by *Theorem 2*. *Theorem 2*, however, provides a tighter lower bound than that given by *Theorem 1*.

APPENDIX B

Local-in-time optimization

B.1 Local-in-time optimization code: lit.py

```python
import tools
import numpy as np
import matplotlib.pyplot as plt
import copy
import time
import os
import sys
import pickle
import pprint


class sol(object):
    def __init__(self):
        self.M = 0  # Number of total steps
        self.N = 0  # Number of spatial grid points in a single dimension
        self.T = 0.  # Final time
        self.dt = 0.
        self.Pe = 0.  # Peclet
        self.L = 0.  # Length of box side
        self.hist_th_hm1 = []
        self.hist_th_l2 = []
        self.hist_th_h1 = []
        self.hist_th_time = []
        self.hist_th = []

        self.hist_u = []
        self.hist_u_time = []
        self.hist_u_h1 = []
        self.hist_u_l2 = []


def sim(N=128, M=1000, T=1.0, L=1.0, gamma=1.0, U=1.0, Pe=1024,
        T_kick=0.01, save_th_every=10, save_u_every=10, pickle_file=None,
        plot=False, constraint='enstrophy'):

    def f(th, u):
        th_d = st.dealias(th)
        return st.dealias(-1.0 * np.sum(vt.dealias(u) * st.grad(th_d), 0)
                          + kappa * st.lap(th_d))

    def f_lit(th):
        return f(th, u_lit(th))

    def u_lit_enstrophy(th):
        th_d = st.dealias(th)
        u_lit = th_d * st.grad_invlap(th_d)
        u_lit = - vt.invlap(vt.div_free_proj(u_lit))
        u_lit = gamma * L * u_lit / st.l2norm(vt.curl(u_lit))
        return u_lit

    def u_lit_energy(th):
        u_lit = st.dealias(th) * vt.dealias(st.grad_invlap(th))
        u_lit = vt.div_free_proj(u_lit)
        u_lit = U * L * u_lit / vt.l2norm(u_lit)
        return u_lit

    if constraint == 'enstrophy':
        u_lit = u_lit_enstrophy
    elif constraint == 'energy':
        u_lit = u_lit_energy

        # Parameters
    h = L / N
    kappa = 1. / Pe

    # ## Double precision
    ftype = np.float64
    ctype = np.complex128
    total_steps = M
    dt = T / M
    print('dt = ', dt)
    final_time_ind = total_steps
    total_time_pts = total_steps + 1

    X = np.mgrid[:N, :N].astype(ftype) * h
    Nf = N // 2 + 1
    kx = np.fft.fftfreq(N, 1. / N).astype(int)
    ky = kx[:Nf].copy()
    ky[-1] *= -1
    K = np.array(np.meshgrid(kx, ky, indexing='ij'), dtype=int)

    st = tools.ScalarTool(N, L)
    vt = tools.VectorTool(N, L)

    th0 = np.sin(2. * np.pi * X[0] / L)
    th = copy.copy(th0)

    # Initial kick
    # The default value for T_kick equal to 0.01 is sufficient to
    # initiate LIT optimization.

    num_steps_kick = int(max(round(T_kick / dt), 10))
    dt_kick = T_kick / num_steps_kick

    u_kick = np.zeros((2, N, N), dtype=ftype)
    u_kick[0, :, :] = np.sin(2. * np.pi * X[1] / L)

    for i in range(num_steps_kick):
        k1 = f(th, u_kick)
        k2 = f(th + 0.5 * dt_kick * k1, u_kick)
        k3 = f(th + 0.5 * dt_kick * k2, u_kick)
        k4 = f(th + dt_kick * k3, u_kick)
        th = th + dt_kick * (1.0 / 6.0) * (k1 + 2.0 * k2 + 2.0 * k3 + k4)

    time = 0.0

    hist_th = [th]
    hist_th_time = [time]
    hist_th_hm1 = [st.hm1norm(th)]
    hist_th_l2 = [st.l2norm(th)]
    hist_th_h1 = [st.h1norm(th)]

    u = u_lit(th)
    hist_u = [u]
    hist_u_time = [time]
    hist_u_h1 = [vt.h1norm(u)]
    hist_u_l2 = [vt.l2norm(u)]
    if plot:
        plt.figure()
        st.plot(th)
        plt.title('time = %2.3f' % time)
        plt.show()

    u0 = copy.copy(u)
    assert total_steps == M

    for i in range(1, total_steps + 1):
        k1 = f_lit(th)
        k2 = f_lit(th + 0.5 * dt * k1)
        k3 = f_lit(th + 0.5 * dt * k2)
        k4 = f_lit(th + dt * k3)
        th = th + dt * (1.0 / 6.0) * (k1 + 2.0 * k2 + 2.0 * k3 + k4)
        time += dt

        if np.mod(i, save_th_every) == 0:

            hist_th.append(th)
            hist_th_time.append(time)
            hist_th_hm1.append(st.hm1norm(th))
            hist_th_l2.append(st.l2norm(th))
            hist_th_h1.append(st.h1norm(th))

            if plot:
                plt.figure()
                st.plot(th)
                plt.title('time = %2.3f' % time)
                plt.show()

                vt.plot(u)
                plt.show()

        if np.mod(i, save_u_every) == 0:
            u = u_lit(th)
            hist_u.append(u)
            hist_u_time.append(time)
            hist_u_h1.append(vt.h1norm(u))
            hist_u_l2.append(vt.l2norm(u))

    sol_save = sol()
    sol_save.M = total_time_pts
    sol_save.N = N
    sol_save.T = T
    sol_save.dt = dt
    sol_save.Pe = Pe
    sol_save.L = L

    sol_save.hist_th = hist_th
```

```python
    sol_save.hist_th_time = hist_th_time
    sol_save.hist_th_hm1 = hist_th_hm1
    sol_save.hist_th_l2 = hist_th_l2
    sol_save.hist_th_h1 = hist_th_h1

    sol_save.hist_u = hist_u
    sol_save.hist_u_time = hist_u_time
    sol_save.hist_u_h1 = hist_u_h1
    sol_save.hist_u_l2 = hist_u_l2

    if pickle_file != None:
        output = open(pickle_file, 'wb')
        pickle.dump(sol_save, output)

    return sol_save


def movie(time, scalar_hist, N, L, output_path='output/'):
    os.system('mkdir ' + output_path)
    os.system('mkdir ' + output_path + 'images/')
    st = tools.ScalarTool(N, L)
    # st.plot(scalar_hist[i])
    # plt.savefig(outputPath + "image%.4d.png" % i, format='png')
    for i in range(len(time)):
        fig = plt.figure()
        st.plot(np.real(scalar_hist[i]))
        plt.title('Time = %.3f' % time[i])
        plt.savefig(output_path + 'images/' + "image%.4d.png" %
                    i, format='png')
        # plt.savefig("image.png", format='png')
        plt.close(fig)

    os.system("ffmpeg -y -framerate 20 -i " + output_path + 'images/'
              "image%04d.png -c:v libx264 -pix_fmt yuv420p " + output_path +
              "movies.mp4")

    # os.system('rm -r ' + output_path + 'images/')


def compute_norms(scalar_hist, N, L):
    st = tools.ScalarTool(N, L)
    time_length, _, _ = np.shape(scalar_hist)

    hm1norm_hist = np.zeros(time_length)
    l2norm_hist = np.zeros(time_length)
    h1norm_hist = np.zeros(time_length)

    for i, scalar in enumerate(scalar_hist):
        hm1norm_hist[i] = st.hm1norm(scalar)
        l2norm_hist[i] = st.l2norm(scalar)
        h1norm_hist[i] = st.h1norm(scalar)

    return [hm1norm_hist, l2norm_hist, h1norm_hist]


def plot_norms(time, scalar_hist, N, L, high_quality=False, graph='log'):
    if high_quality:
        plt.rc('text', usetex=True)
        plt.rc('font', family='serif', size=12)
    else:
        plt.rc('text', usetex=False)
        plt.rc('font', family='sans-serif', size=12)
    hm1norm_hist, l2norm_hist, h1norm_hist = compute_norms(
   ↪ scalar_hist, N, L)

    if graph == 'log':
        plt.semilogy(time, hm1norm_hist,
                     label=r'$H^{-1}$', linestyle='-', color='k')
        plt.semilogy(time, l2norm_hist, label=r'$L^2$',
                     linestyle='--', color='k')
        plt.semilogy(time, h1norm_hist,
                     label=r'$H^{1}$', linestyle=':', color='k')
    elif graph == 'linear':
        plt.plot(time, hm1norm_hist,
                 label=r'$H^{-1}$', linestyle='-', color='k')
        plt.plot(time, l2norm_hist, label=r'$L^2$',
                 linestyle='--', color='k')
        plt.plot(time, h1norm_hist,
                 label=r'$H^{1}$', linestyle=':', color='k')

    plt.legend()
    plt.xlabel('Time')
    plt.grid(alpha=0.5)


if __name__ == "__main__":

    # PLEASE READ BEFORE RUNNING ON FLUX SERVER!
    #
    # Transfer this script lit.py , tools.py, and flux_pbs_enstrophy.
   ↪ sh (or flux_pbs_energy.sh)
```

```
# to flux.
#
# This is specifically built to interface with the pbs scrpts to work on
# univeristy of Michigan's flux server. Make sure to use the following command
# ‘qsub -t 1-27 flux_pbs_enstrophy.sh’ where 27 is chosen since it is the (Length
# of Pe_list[=9]) times (number of trials per Peclet[=3]). Similarly
# for the energy constraint run ‘qsub -t 1-33 flux_pbs_energy.sh’

sim_num = int(sys.argv[1])
constraint = str(sys.argv[2])
T = float(sys.argv[3])

print(sim_num)

if constraint == "enstrophy":
    Pe_list = [128.0, 256.0, 512.0, 1024.0,
               2048.0, 4096.0, 8192.0, 16384.0, np.inf]
elif constraint == "energy":
    Pe_list = [1.0, 2.0, 4.0, 8.0, 16.0, 32.0,
               64.0, 128.0, 256.0, 512.0, np.inf]

Pe = Pe_list[(sim_num - 1) // 3]
kappa = 1.0 / Pe

# THESE PARAMETERS SHOULD NOT BE CHANGE!
L = 1.0
U = 1.0
```

```
    gamma = 1.0

    # Determine N and dt_cfl
    if constraint == "enstrophy":
        if Pe == np.inf:
            N = 512
            dt_cfl = 0.25 * (L / N) / (gamma * L)
        else:
            lb = (kappa / gamma)**0.5
            l_smallest = 0.25 * lb  # a quarter of batchelor scale
            print('lb = ', lb)
            num_wavelengths = L / l_smallest
            print('N_boyd = ', tools.N_boyd(num_wavelengths))
            N = min(tools.N_boyd(num_wavelengths), 512)
            dt_cfl = 0.25 * min((L / N)**2. / kappa, (L / N) / (gamma
↪  * L))

    elif constraint == "energy":

        if Pe == np.inf:
            N = 512
            dt_cfl = 0.25 * (L / N) / (U)
        else:
            lb = (kappa / U)
            l_smallest = 0.25 * lb  # a quarter of batchelor scale
            print('lb = ', lb)
            num_wavelengths = L / l_smallest
            print('N_boyd = ', tools.N_boyd(num_wavelengths))
            N = min(tools.N_boyd(num_wavelengths), 512)
            dt_cfl = 0.25 * min((L / N)**2. / kappa, (L / N) / U)

    print('N = ', N)
    print('dt CFL = ', dt_cfl)

    # Determine M_list given dt_cfl
    # Run 3 different simulations with 2 and 4 times as many time num_steps
    # This will be used to calculate convergence metrics.
    M0 = round(T / dt_cfl)  # approx number of time steps according to CFL
    M0 = int(2**np.ceil(np.log2(M0)))  # make power of two
    M_list = [M0, int(2 * M0), int(4 * M0)]

    # Select M based off of sim_num
    M_index = (sim_num - 1) % 3
    M = M_list[M_index]

    output_folder = "output-pe=" + str(Pe) + "-M=" + str(M) + "/"
    os.system('mkdir ' + output_folder)
    pickle_file = output_folder + "pe=" + str(Pe) + "-M=" + str(M) + ".pkl"

    save_th_every = max(int(round(M / 128)), 1)
    save_u_every = max(int(round(M / 8)), 1)

    solution = sim(N=N, M=M, T=T, L=L, U=U, gamma=gamma, Pe=Pe, constraint=constraint,
                   save_th_every=save_th_every, save_u_every=save_u_every,
                   pickle_file=pickle_file, plot=False)

    movie(solution.hist_th_time, solution.hist_th,
          N, L, output_path=output_folder)

    plt.figure()

    st = tools.ScalarTool(N, L)
    st.plot(solution.hist_th[-1])
    plt.savefig(output_folder + 'plot_final_frame-pe=' + str(Pe) + '.png')

    plt.figure()
    plot_norms(solution.hist_th_time, solution.hist_th, N, L)
    plt.savefig(output_folder + 'plot_norms-pe=' + str(Pe) + '.png')
```

B.2 Local-in-time optimization code: tools.py

```
import numpy as np
import pyfftw.interfaces.numpy_fft as fft
# from numpy import fft
from pyfftw.interfaces import cache
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
cache.enable()


def N_boyd(M):
    """ Boyd's rule of thumb. M is the number of wave lengths
    given by M = L/l where L is the box size and l is the smallest
    scale to be resolved """
    return int(2**np.ceil(np.log2(4 * (M - 1) + 6)))

```

```

class ScalarTool(object):
    """
    Description:
    ScalarTool contains a collection of functions necessary to
    ↪ compute basic
    operations such as gradients and norms on scalars defined on a 2D
    ↪  periodic
    square domain of length L and discretized in each dimension into
    ↪ N
    intervals.

    Inputs:
    N - number of discretized points in each dimension
    L - length of side
    """

    def __init__(self, N, L):
        self.N = N
        self.L = L
        self.h = self.L / self.N
        self.X = np.mgrid[:self.N, :self.N].astype(float) * self.h

        self.Nf = self.N // 2 + 1
        self.kx = np.fft.fftfreq(self.N, 1. / self.N).astype(int)
        self.ky = self.kx[:self.Nf].copy()
        self.ky[-1] *= -1
        self.K = np.array(np.meshgrid(
            self.kx, self.ky, indexing='ij'), dtype=int)
        self.K2 = np.sum(self.K * self.K, 0, dtype=int)
        self.KoverK2 = self.K.astype(
            float) / np.where(self.K2 == 0, 1, self.K2).astype(float)
        self.oneoverK2 = 1.0 / \
            np.where(self.K2 == 0.0, 1.0, self.K2).astype(float)
        self.mean_zero_array = self.K2 != 0.0
        self.kmax_dealias = 2. / 3. * (self.N / 2 + 1)
        self.dealias_array = np.array((abs(self.K[0]) < self.
   ↪ kmax_dealias) * (
            abs(self.K[1]) < self.kmax_dealias), dtype=bool)
        self.num_threads = 1

    def l2norm(self, scalar):
        self.scalar_input_test(scalar)
        return np.sum(np.ravel(scalar)**2.0 * self.h**2.0)**0.5

    def grad(self, scalar):
        self.scalar_input_test(scalar)

        scalar_hat = self.fft(scalar)
        return fft.irfftn(1.0j * self.K * (2 * np.pi / self.L) *
   ↪ scalar_hat, axes=(1, 2), threads=self.num_threads)

    def h1norm(self, scalar):
        self.scalar_input_test(scalar)
        grad_scalar = self.grad(scalar)
        grad_scalar_sq = np.sum(grad_scalar * grad_scalar, 0)
        integrand = np.ravel(grad_scalar_sq)
        return np.sum(integrand * self.h**2.0)**0.5

    def lap(self, scalar):
        self.scalar_input_test(scalar)
        scalar_hat = self.fft(scalar)
        return self.ifft((-1.0) * self.K2 * (2 * np.pi / self.L)**2.0
↪  * scalar_hat)

    def invlap(self, scalar):
        self.scalar_input_test(scalar)
        scalar_hat = self.fft(scalar)
        return np.real(self.ifft(-1.0 * (2.0 * np.pi / self.L)
↪ **(-2.0) *
                                     self.oneoverK2 * self.
↪ mean_zero_array * scalar_hat))

    def grad_invlap(self, scalar):
        self.scalar_input_test(scalar)
        scalar_hat = self.fft(scalar)
        return fft.irfftn(-1.0j * self.KoverK2 * (2 * np.pi / self.L)
↪ **(-1.0) * scalar_hat, axes=(1, 2), threads=self.num_threads)

    def hm1norm(self, scalar):
        self.scalar_input_test(scalar)
        grad_invlap_scalar = self.grad_invlap(scalar)
        grad_invlap_scalar_sq = np.sum(grad_invlap_scalar *
                                       grad_invlap_scalar, 0)  # dot
↪ product
        integrand = np.ravel(grad_invlap_scalar_sq)
        return np.sum(integrand * self.h**2.0)**0.5

    def plot(self, scalar, high_quality=False, fixed_vertical_axis=
↪ False):

        if high_quality:
            plt.rc('text', usetex=True)
            plt.rc('font', family='serif', size=12)
        else:
            plt.rc('text', usetex=False)
            plt.rc('font', family='sans-serif', size=12)

        if fixed_vertical_axis:
            vmin = -1
            vmax = 1
        else:
            vmin = np.amin(scalar)
            vmax = np.amax(scalar)
        self.scalar_input_test(scalar)
        im = plt.imshow(np.transpose(scalar),
                        cmap=plt.cm.gray,
                        extent=(0, self.L, 0, self.L),
                        origin="lower",
                        vmin=vmin,
                        vmax=vmax)
        plt.xlabel(r'$x$')
        plt.ylabel(r'$y$')
        plt.colorbar(im)

    def scalar_input_test(self, scalar):
        if np.shape(scalar) != (self.N, self.N):
            print(np.shape(scalar))
            raise InputError("Scalar field array does not have correct shape.")
        if not np.all(np.isrealobj(scalar)):
            raise InputError("Scalar field array should be real.")

    def scalar_hat_input_test(self, scalar_hat):
```

```python
        if np.shape(scalar_hat) != (self.N, self.Nf):
            print(np.shape(scalar_hat))
            raise InputError("Scalar field array does not have correct shape.")

    def sint(self, scalar):
        """ Performs spatial integration """
        self.scalar_input_test(scalar)
        return np.sum(np.ravel(scalar) * self.h**2.0)

    def dealias(self, scalar):
        """ Perform 1/3 dealias on scalar """
        self.scalar_input_test(scalar)
        temp_hat = self.fft(scalar) * self.dealias_array
        return self.ifft(temp_hat)

    def fft(self, scalar):
        """ Performs fft of scalar field """
        self.scalar_input_test(scalar)
        return fft.rfftn(scalar, threads=self.num_threads)

    def ifft(self, scalar_hat):
        """ Performs inverse fft of scalar field """
        self.scalar_hat_input_test(scalar_hat)
        return fft.irfftn(scalar_hat, threads=self.num_threads)

    def subtract_mean(self, scalar):
        """ subtract off mean """
        self.scalar_input_test(scalar)
        scalar_hat = self.fft(scalar)
        return np.real(self.ifft(scalar_hat * self.mean_zero_array))

    def get_spectrum(self, scalar):
        """ gets spectrum """
        self.scalar_input_test(scalar)
        scalar_hat = self.fft(scalar)
        amp = 2.0 * np.absolute(scalar_hat) / \
            self.N**2.0  # corrects normalization
        k_list = np.arange(0, self.N // 2 + 1, 1)  # beginning of bin intervals
        K_inf = np.maximum(abs(self.K[0]), abs(self.K[1]))  # infinity norm
        amp_list = []

        for k in k_list:
            K_shell_bool = k == K_inf
            max_amp_in_shell = np.amax(amp * K_shell_bool)
            amp_list.append(max_amp_in_shell)

        return [k_list, amp_list]

    def isblocked(self, scalar, k_frac=0.85, amp_thres=10.**(-10)):
        """ determines if spectral blocking is present """
        self.scalar_input_test(scalar)
        k_thres = int(k_frac * (self.N / 2))
        [k_list, amp_list] = self.get_spectrum(scalar)
        amp_beyond_k_thres = [amp_list[i]
                              for i in range(len(k_list)) if k_list[i] > k_thres]
        return max(amp_beyond_k_thres) > amp_thres


```

```python
class VectorTool(object):
    """
    Description:
    VectorTool contains a collection of functions necessary to compute basic
    operations such as norms on scalars defined on a 2D periodic
    square domain of length L and discretized in each dimension into N
    intervals.

    Inputs:
    N - number of discretized points in each dimension
    L - length of side
    """

    def __init__(self, N, L):
        self.N = N
        self.L = L
        self.h = self.L / self.N
        self.X = np.mgrid[:self.N, :self.N].astype(float) * self.h
        self.Nf = self.N // 2 + 1
        self.kx = np.fft.fftfreq(self.N, 1. / self.N).astype(int)
        self.ky = self.kx[:self.Nf].copy()
        self.ky[-1] *= -1
        self.K = np.array(np.meshgrid(
            self.kx, self.ky, indexing='ij'), dtype=int)
        self.K2 = np.sum(self.K * self.K, 0, dtype=int)
        self.KoverK2 = self.K.astype(
            float) / np.where(self.K2 == 0, 1, self.K2).astype(float)
        self.oneoverK2 = 1.0 / \
            np.where(self.K2 == 0.0, 1.0, self.K2).astype(float)
        self.mean_zero_array = self.K2 != 0.0
        self.kmax_dealias = 2. / 3. * (self.N / 2 + 1)
        self.dealias_array = np.array((abs(self.K[0]) < self.
↪ kmax_dealias) * (
            abs(self.K[1]) < self.kmax_dealias), dtype=bool)
        self.num_threads = 1

    def div(self, vector):
        """ Take divergence of vector """
        self.vector_input_test(vector)
        vector_hat = self.fft(vector)
        return fft.irfftn(np.sum(1j * self.K * (2 * np.pi / self.L) *
↪  vector_hat, 0), threads=self.num_threads)

    def fft(self, vector):
        """ Performs fft of vector field """
        self.vector_input_test(vector)
        return fft.rfftn(vector, axes=(1, 2), threads=self.
↪ num_threads)

    def ifft(self, vector_hat):
        """ Performs inverse fft of vector hat field """
        self.vector_hat_input_test(vector_hat)
        return fft.irfftn(vector_hat, axes=(1, 2), threads=self.
↪ num_threads)

    def plot(self, vector, high_quality=False):
        """ Plots a quiver plot of the vector field """
        self.vector_input_test(vector)
        if high_quality:
            plt.rc('text', usetex=True)
            plt.rc('font', family='serif', size=12)
        else:
            plt.rc('text', usetex=False)
            plt.rc('font', family='sans-serif', size=12)

        m = max(round(self.N / 25), 1)
        Q = plt.quiver(self.X[0][1:-1:m, 1:-1:m],
                       self.X[1][1:-1:m, 1:-1:m],
                       vector[0][1:-1:m, 1:-1:m],
                       vector[1][1:-1:m, 1:-1:m], linewidths=2.0)
        plt.quiverkey(
            Q, 0.8, 1.03, 2, r'%.2f $\frac{m}{s}$' % np.amax(vector), labelpos='E',)
        plt.xlabel(r'$x$')
        plt.ylabel(r'$y$')
        plt.title('')
        plt.xlim(0.0, self.L)
        plt.ylim(0.0, self.L)
        plt.axis('scaled')

    def dealias(self, vector):
        """ Dealias vector """
        self.vector_input_test(vector)
        vector_hat = self.fft(vector)
        vector_hat = vector_hat * self.dealias_array
        return np.real(self.ifft(vector_hat))

    def l2norm(self, vector):
        """ L2 norm of a vector field """
        self.vector_input_test(vector)
        integrand = np.sum(vector * vector, 0)

        return np.sum(np.ravel(integrand) * self.h**2)**0.5

    def h1norm(self, vector):
        """ L2 norm of a vector field """
        self.vector_input_test(vector)
        vector_hat = self.fft(vector)
        grad_vx = fft.irfftn(1.0j * self.K * (2 * np.pi / self.L)
                             * vector_hat[0], axes=(1, 2), threads=
↪ self.num_threads)
        grad_vy = fft.irfftn(1.0j * self.K * (2 * np.pi / self.L)
                             * vector_hat[1], axes=(1, 2), threads=
↪ self.num_threads)

        integrand = (grad_vx[0]**2.0 + grad_vx[1]**2.0
                     + grad_vy[0]**2.0 + grad_vy[1]**2.0)

        return np.sum(np.ravel(integrand) * self.h**2)**0.5

    def vector_input_test(self, vector):
        """ Determines if vector is correct size """
        if np.shape(vector) != (2, self.N, self.N):
            print(np.shape(vector))
            raise InputError("Vector field array does not have
↪ correct shape")

        if not np.all(np.isrealobj(vector)):
            raise InputError("Scalar field array should be real.")

    def vector_hat_input_test(self, vector_hat):
        """ Determines if vector is correct size """
        if np.shape(vector_hat) != (2, self.N, self.Nf):
            print(np.shape(vector_hat))
            raise InputError("Vector field array does not have
↪ correct shape")

    def is_incompressible(self, vector):
        self.vector_input_test(vector)
        return np.allclose(self.div(vector), 0)

    def div_free_proj(self, vector):
        """ performs leray divergence-free projection """
        self.vector_input_test(vector)
        vector_hat = self.fft(vector)
        return self.ifft(vector_hat - self.KoverK2 * np.sum(self.K *
↪ vector_hat, 0))

    def curl(self, vector):
        """ Perform curl of vector """
        self.vector_input_test(vector)
        vector_hat = self.fft(vector)
        w = fft.irfftn(
            1j * self.K[0] * (2.0 * np.pi / self.L) * vector_hat[1]
            - 1j * self.K[1] * (2.0 * np.pi / self.L) * vector_hat
↪ [0], threads=self.num_threads)
        return w

    def invlap(self, vector):
        """ Inverse laplacian of vector """
        self.vector_input_test(vector)
        vector_hat = self.fft(vector)
        return np.real(self.ifft(-1.0 * (2.0 * np.pi / self.L)
   ↪ **(-2.0) *
                                 self.oneoverK2 * self.
   ↪ mean_zero_array * vector_hat))

    def lap(self, vector):
        """ Laplacian of vector """
        self.vector_input_test(vector)
        vector_hat = self.fft(vector)
        return np.real(self.ifft(-1.0 * (2.0 * np.pi / self.L)**(2.0)
   ↪  * (self.K2) * vector_hat))

    def subtract_mean(self, vector):
        """ subtract off mean """
        self.vector_input_test(vector)
        vector_hat = self.fft(vector)
        return np.real(self.ifft(vector_hat * self.mean_zero_array))


class InputError(Exception):
    """ Input Error """

    def __init__(self, message):
        self.message = message
```

APPENDIX C

Global-in-time optimization

C.1 Total variation calculation

Let $\theta(\mathbf{x},t)$ and $\mathbf{u}(\mathbf{x},t)$ be arbitrary functions on $D \times [0,T]$ and define a cost functional be

$$C\{\theta\} = \|\theta(\mathbf{x},T)\|^2_{H^{-1}} = \int\limits_D d\mathbf{x} |\nabla^{-1}\theta(\mathbf{x},T)|^2.$$

Define the quantities:

$$\begin{aligned} g_1\{\theta,\mathbf{u}\} &= \partial_t\theta + \mathbf{u}\cdot\nabla\theta - \kappa\Delta\theta \\ g_2\{\mathbf{u}\} &= \nabla\cdot\mathbf{u} \\ g_3\{\mathbf{u}\} &= \int\limits_0^T\int\limits_D d\mathbf{x}dt|\nabla\mathbf{u}|^2 - TL^d\Omega^2. \end{aligned}$$

Let $\phi(\mathbf{x},t)$, and $q(\mathbf{x},t)$ be arbitrary functions on $D \times [0,T]$ and let μ be a scalar. Define the functional G as

$$G\{\theta, \mathbf{u}, \phi, q, \mu\} = \iint [\phi(\mathbf{x}, t) g_1\{\theta, \mathbf{u}\} + q(\mathbf{x}, t) g_2\{\mathbf{u}\}] d\mathbf{x} dt \\ + \mu g_3\{\mathbf{u}\}$$

Let the pair $\{\theta_0, \mathbf{u}_0\}$ be an extrema of C and $\{\theta_1, \mathbf{u}_1\}$ be an admissible pair. Note that since $\{\theta_0, \mathbf{u}_0\}$ and $\{\theta_1, \mathbf{u}_1\}$ are admissible, $G\{\theta_0, \mathbf{u}_0, \phi, q, \mu\} = 0$ and $G\{\theta_1, \mathbf{u}_1, \phi, q, \mu\} = 0$. Hence, the total variation of G is also zero,

$$\Delta G = G\{\theta_1, \mathbf{u}_1, \phi, q, \mu\} - G\{\theta_0, \mathbf{u}_0, \phi, q, \mu\} = 0.$$

Let $\delta\theta = \theta_1 - \theta_0$ and $\delta\mathbf{u} = \mathbf{u}_1 - \mathbf{u}_0$.

$$\begin{aligned} \Delta G = & \iint [\phi(\mathbf{x}, t) g_1\{\theta_1, \mathbf{u}_1\} + q(\mathbf{x}, t) g_2\{\mathbf{u}_1\}] d\mathbf{x} dt + \mu g_3\{\mathbf{u}_1\} \\ & - \iint [\phi(\mathbf{x}, t) g_1\{\theta_0, \mathbf{u}_0\} + q(\mathbf{x}, t) g_2\{\mathbf{u}_0\}] d\mathbf{x} dt - \mu g_3\{\mathbf{u}_0\} \end{aligned}$$

$$\begin{aligned} = & \iint [\phi \left(\partial_t(\theta_0 + \delta\theta) + (\mathbf{u}_0 + \delta\mathbf{u}) \cdot \nabla(\theta_0 + \delta\theta) - \partial_t\theta_0 - \mathbf{u}_0 \cdot \nabla\theta_0 \right) \\ + & q \left(\nabla \cdot (\mathbf{u}_0 + \delta\mathbf{u}) - \nabla \cdot \mathbf{u_0} \right) \\ + & \mu \left(|\nabla(\mathbf{u}_0 + \delta\mathbf{u})|^2 - |\nabla\mathbf{u}_0|^2 \right)] d\mathbf{x} dt \end{aligned}$$

$$
\begin{aligned}
\Longrightarrow \ \Delta G &= \iint \{(-\partial_t \phi - \mathbf{u}_0 \cdot \nabla \phi)\delta\theta + (\phi \nabla \theta_0 - \nabla q - \mu \Delta \mathbf{u}_0) \cdot \delta \mathbf{u} \\
&+ \nabla \phi \cdot \delta \mathbf{u} \delta \theta + \mu |\nabla \delta \mathbf{u}|^2 t\} d\mathbf{x} dt \\
&+ \int\limits_D \phi(x,T) \delta\theta(x,T) d\mathbf{x} = 0
\end{aligned}
$$

Consider the total variation of C

$$\Delta C = C\{\theta_1\} - C\{\theta_0\}$$

$$= \int\limits_D \left\{ |\nabla^{-1}\theta_1(\mathbf{x},T)|^2 - |\nabla^{-1}\theta_0(\mathbf{x},T)|^2 \right\} d\mathbf{x}$$

$$= \int\limits_D \left\{ |\nabla^{-1}\left[\theta_0(\mathbf{x},T) + \delta\theta(\mathbf{x},T)\right]|^2 - |\nabla^{-1}\theta_0(\mathbf{x},T)|^2 \right\} d\mathbf{x}$$

$$= \int\limits_D \left\{ \nabla^{-1}\theta_0(\mathbf{x},T) \cdot \nabla^{-1}\delta\theta(\mathbf{x},T) + |\nabla^{-1}\delta\theta(\mathbf{x},T)|^2 \right\} d\mathbf{x}$$

$$= \int\limits_D \left\{ -\Delta^{-1}\theta_0(\mathbf{x},T)\delta\theta(\mathbf{x},T) + |\nabla^{-1}\delta\theta(\mathbf{x},T)|^2 \right\} d\mathbf{x}$$

Since $\Delta G = 0$, we can add ΔG to ΔC without any consequence. Let the index "$_T$" be shorthand for the arguments $(\mathbf{x}, T)$. (e.g. $\phi_T = \phi(\mathbf{x}, T)$)

$$
\begin{aligned}
\Delta C &= \Delta C + \Delta G \\
&= \iint \{(-\partial_t \phi - \mathbf{u}_0 \cdot \nabla\phi - \kappa\Delta\phi)\delta\theta \\
&+ (\phi\nabla\theta_0 - \nabla q - \mu\Delta\mathbf{u}_0) \cdot \delta\mathbf{u} + \nabla\phi \cdot \delta\mathbf{u}\delta\theta + \mu|\nabla\delta\mathbf{u}|^2\} d\mathbf{x}dt \\
&+ \int_D \left\{(\phi_T - \Delta^{-1}\theta_{0,T})\delta\theta_T + |\nabla^{-1}\delta\theta_T|^2\right\} d\mathbf{x}
\end{aligned}
$$

Since $\{\theta_0, \mathbf{u}_0\}$ is an extrema, the first variation vanishes. The total variation about the extrema $\{\theta_0, \mathbf{u}_0\}$ becomes

$$
\Delta C \;=\; \iint \left\{\nabla\phi \cdot \delta\mathbf{u}\delta\theta + \mu|\nabla\delta\mathbf{u}|^2\right\} d\mathbf{x}dt + \int_D |\nabla^{-1}\delta\theta_T|^2 d\mathbf{x}. \qquad \text{(C.1)}
$$

C.2 Global-in-time optimization code: git.py

```
import tools
import numpy as np
import matplotlib.pyplot as plt
import copy
import time
import os
import pickle
import pprint
import lit


def increase_res(scalar, N_grid):
    st_grid = tools.ScalarTool(N_grid, L)
    th_hat = np.fft.rfft(scalar, axis=1)
```

```python

    th_hat1 = np.zeros((N, N_grid // 2 + 1), dtype=ctype)
    th_hat1[:, 0:(N // 2 + 1)] = N_grid / N * th_hat

    th_hat2 = np.fft.fft(th_hat1, axis=0)

    th_hat3 = np.zeros((N_grid, N_grid // 2 + 1), dtype=ctype)
    th_hat3[0:N // 2, :] = N_grid / N * th_hat2[0:N // 2, :]
    th_hat3[(N_grid - N // 2):, :] = N_grid / N * th_hat2[N // 2:, :]
    th_new = st_grid.ifft(th_hat3)
    return th_new


def f(th, u):
    th_d = st.dealias(th)
    return st.dealias(-1.0 * np.sum(vt.dealias(u) * st.grad(th_d), 0)
                      + kappa * st.lap(th_d))


def g(phi, u):
    phi_d = st.dealias(phi)
    return st.dealias(-1.0 * np.sum(vt.dealias(u) * st.grad(phi_d),
   ↪ 0)
                      - kappa * st.lap(phi_d))


def line_search(u, d, eta_array):
    J_array = np.zeros(np.shape(eta_array))
    for i, eta in enumerate(eta_array):
        J_array[i] = get_obj(normalize(div_free_proj(dealias(u + eta
   ↪ * d))))
```

```
    plt.figure()
    plt.loglog(eta_array, J_array)
    plt.show()
    return eta_array[np.argmin(J_array)]


def get_obj(u):
    # Forward integration
    th = integrate_forward(u, th0)
    return 0.5 * st.hm1norm(th[M - 1])**2


def integrate_forward(u, th0):
    th = np.zeros((M, N, N))
    th[0] = th0
    for i in range(M - 1):
        # Heun's method
        k1 = f(th[i], u[i])
        th_euler = th[i] + dt * k1
        th[i + 1] = th[i] + 0.5 * dt * (k1 + f(th_euler, u[i + 1]))
    return th


def integrate_backward(u, phiT):
    phi = np.zeros((M, N, N))
    phi[M - 1] = phiT
    for i in reversed(range(M - 1)):
        # Heun's method
        k1 = g(phi[i + 1], u[i + 1])
        phi_euler = phi[i + 1] - dt * k1
        phi[i] = phi[i + 1] - 0.5 * dt * (k1 + g(phi_euler, u[i]))
    return phi


def compute_gradJ_and_J(u):
    # Forward integration
    th = integrate_forward(u, th0)

    # Compute objective
    obj = 0.5 * st.hm1norm(th[M - 1])**2

    # Backward integration
    phiT = st.invlap(th[M - 1])
    phi = integrate_backward(u, phiT)

    # Compute gradient
    grad = np.zeros((M, 2, N, N))
    for i in range(M):
        grad[i] = st.dealias(phi[i]) * vt.dealias(st.grad(th[i]))
        grad[i] = vt.div_free_proj(vt.dealias(grad[i]))
    lapu = lap(u)
    mu = dot(lapu, grad) / dot(lapu, lapu)
    grad = grad - mu * lapu

    return grad, obj


def compute_d(u):
    # Forward integration
    th = integrate_forward(u, th0)

    # Compute objective
    obj = 0.5 * st.hm1norm(th[M - 1])**2

    # Backward integration
    phiT = st.invlap(th[M - 1])
    phi = integrate_backward(u, phiT)

    # Compute d
    d = np.zeros((M, 2, N, N))
    for i in range(M):
        d[i] = st.dealias(phi[i]) * vt.dealias(st.grad(th[i]))
        d[i] = vt.invlap(vt.div_free_proj(vt.dealias(d[i])))
    d = normalize(d) - u

    return d


def lap(v):
    lapv = np.zeros((M, 2, N, N))
    for i in range(M):
        lapv[i] = vt.lap(v[i])
    return lapv


def invlap(v):
    invlapv = np.zeros((M, 2, N, N))
    for i in range(M):
        invlapv[i] = vt.invlap(v[i])
    return invlapv


def normalize(v):
    return v * (gamma * L) / mean_enstrophy(v)**0.5


def mean_enstrophy(v):
    integ = 0
    for i in range(M):
        integ += vt.h1norm(v[i])**2. * dt
    integ = (1. / T) * integ
    return integ


def is_incompressible(v):
    cond = True
    for i in range(M):
        cond = cond * vt.is_incompressible(v[i])
    return cond == 1


def div_free_proj(v):
    for i in range(M):
        v[i] = vt.div_free_proj(vt.dealias(v[i]))
    return v


def dealias(v):
    for i in range(M):
        v[i] = vt.dealias(v[i])
    return v


def dot(v, u):
    dot = 0
    for i in range(M):
        dot += st.sint(sum(v[i] * u[i], 0)) * dt

    return dot


if __name__ == "__main__":
    N = 64
    M = 1000
    L = 1.0
    h = L / N
    T = 3.0
    dt = T / M
    kappa = 0.0
    gamma = 1.0
    ftype = np.float64
    ctype = np.complex128

    st = tools.ScalarTool(N, L)
    vt = tools.VectorTool(N, L)

    sol_lit = lit.sim(N=N, M=M - 1, Pe=np.inf, plot=False,
                      T=T, save_th_every=M - 1, save_u_every=1)
    u = np.array(sol_lit.hist_u)
    th0 = sol_lit.hist_th[0]
    print(mean_enstrophy(u))
    time_array = sol_lit.hist_u_time

    eta_array = np.array([0.1, .01, 0.001])
    num_iterations = 100
```

```python
    for i in range(num_iterations):
        gradJ, J = compute_gradJ_and_J(u)
        d = compute_d(u)
        eta = line_search(u, d, eta_array)
        u = normalize(div_free_proj(dealias(u + eta * d)))
        print('eta=', eta,
              'mag of gradJ = ', dot(gradJ, gradJ)**0.5,
              'mag d = ', dot(d, d)**0.5,
              'J=', J,
              'incompressible?', is_incompressible(u),
              'mean enstrophy', mean_enstrophy(u))
```

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Giovanni Alberti, Gianluca Crippa, and Anna L. Mazzucato. Exponential self-similar mixing and loss of regularity for continuity equations. *Comptes Rendus Mathematique*, 352(11):901–906, 2014.

[2] Simon L. Altmann. *Rotations, quaternions, and double groups*. Courier Corporation, Mineola, NY, 2005.

[3] Sanjeeva Balasuriya. Optimal perturbation for enhanced chaotic transport. *Physica D: Nonlinear Phenomena*, 202(3-4):155–176, 2005.

[4] George K. Batchelor. Small-scale variation of convected quantities like temperature in turbulent fluid Part 1: General discussion and the case of small conductivity. *Journal of Fluid Mechanics*, 5(01):113–133, 1959.

[5] Dimitri P. Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.

[6] John P. Boyd. Chebyshev and Fourier Spectral Methods. *New York*, page 688, 2000.

[7] Stanley Corrsin. On the spectrum of isotropic temperature fluctuations in an isotropic turbulence. *Journal of Applied Physics*, 22(4):469–473, 1951.

[8] Luca Cortelezzi, Alessandra Adrover, and Massimiliano Giona. Feasibility, efficiency and transportability of short-horizon optimal mixing protocols. *Journal of Fluid Mechanics*, 597:199–231, 2008.

[9] Gianluca Crippa and Camillo De Lellis. Regularity and compactness for the DiPerna Lions flow. In *Hyperbolic problems: theory, numerics, applications*, volume 1, pages 423–430. Springer, Berlin, 2008.

[10] Domenico D'Alessandro, Mohammed Dahleh, and Igor Mezić. Control of mixing in fluid flow: a maximum entropy approach. *IEEE Transactions on Automatic Control*, 44(10):1852–1863, 1999.

[11] P. V. Danckwerts. The definition and measurement of some characteristics of mixtures. *Applied Scientific Research, Section A*, 3(4):279–296, 1952.

[12] Paul E. Dimotakis. Turbulent Mixing. *Annual Review of Fluid Mechanics*, 37:329–356, 2005.

[13] Peter D. Ditlevsen. *Turbulence and shell models.* Cambridge University Press, 2010.

[14] Carl Eckart. An analysis of the stirring and mixing processes in incompressible fluids. *Journal of Marine Research*, 7:265–275, 1948.

[15] Lawrence C. Evans. Partial differential equations. *Graduate studies in mathematics*, 31(3):749, 2010.

[16] Mohammad Farazmand. Optimal initial condition of passive tracers for their maximal mixing in finite time. *Physical Review Fluids*, 2(5):1–16, 2017.

[17] Richard Feynman, Robert B. Leighton, and Matthew Sands. *The Feynman lectures on physics: The new millennium edition, volume III: quantum mechanics.* 2010.

[18] D. P. G. Foures, C. P. Caulfield, and P. J. Schmid. Optimal mixing in two-dimensional plane Poiseuille flow at finite Peclet number. *Journal of Fluid Mechanics*, 748:241–277, 2014.

[19] Charles Fox. *An introduction to the calculus of variations.* Courier Corporation, 1950.

[20] I. M. Gelfand, S. V. Fomin, and Richard A. Silverman. Calculus of variations. 2000.

[21] E. B. Gledzer. System of hydrodynamic type admitting two quadratic integrals of motion. *Soviet Physics Doklady*, 18:216, 1973.

[22] D. M. Hobbs and F. J. Muzzio. Optimization of a static mixer using dynamical systems techniques. *Chemical Engineering Science*, 53(18):3199–3213, 1998.

[23] Mark Holzer and Eric D. Siggia. Turbulent mixing of a passive scalar. *Physics of Fluids*, 5(May):1820–1837, 1994.

[24] Gautam Iyer, Alexander Kiselev, and Xiaoqian Xu. Lower bounds on the mix norm of passive scalars advected by incompressible enstrophy-constrained flows. *Nonlinearity*, 27(5):973, 2014.

[25] M. H. Jensen, G. Paladin, and A. Vulpiani. Shell model for turbulent advection of passive-scalar fields. *Physical Review A*, 45(10):7214–7221, may 1992.

[26] Donald E. Kirk. Optimal Control Theory: An Introduction, 2004.

[27] Daniel Liberzon. *Calculus of variations and optimal control theory: a concise introduction.* Princeton University Press, 2011.

[28] Zhi Lin, Jean-Luc Thiffeault, and Charles R. Doering. Optimal stirring strategies for passive scalar mixing. *Journal of Fluid Mechanics*, 675:465–476, 2011.

[29] Weijiu Liu. Mixing enhancement via flow optimization. *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 5323–5328, 2006.

[30] Evelyn Lunasin, Zhi Lin, Alexei Novikov, Anna Mazzucato, and Charles R. Doering. Optimal mixing and optimal stirring for fixed energy, fixed power, or fixed palenstrophy flows. *Journal of Mathematical Physics*, 53:1–15, 2012.

[31] George Mathew, Igor Mezić, Symeon Grivopoulos, Umesh Vaidya, and Linda Petzold. Optimal control of mixing in Stokes fluid flows. *Journal of Fluid Mechanics*, 580:261, 2007.

[32] George Mathew, Igor Mezić, and Linda Petzold. A multiscale measure for mixing. *Physica D: Nonlinear Phenomena*, 211(1):23–46, 2005.

[33] Christopher J. Miles and Charles R. Doering. A Shell Model for Optimal Mixing. *Journal of Nonlinear Science*, pages 1–34, 2017. https://doi.org/10.1007/s00332-017-9400-7

[34] Christopher J. Miles and Charles R. Doering. Diffusion-limited mixing by incompressible flows. *accepted to Nonlinearity*, pages 1–14, 2018.

[35] A. M. Obukhov. Structure of the temperature field in turbulent flow. *Izvestiia Akademii Nauk SSSR, Geogr. i Geofiz*, 13:58–69, 1949.

[36] Edward L. Paul, Victor A. Atiemo-Obeng, and Suzanne M. Kresta. *Handbook of industrial mixing: science and practice.* John Wiley & Sons, Inc., 2004.

[37] Chi-Cheung Poon. Unique continuation for parabolic equations. *Comm. in Partial differential Equations*, 21(3-4):521–539, 1996.

[38] Isidor Isaac Rabi, N. F. Ramsey, and J. Schwinger. Use of rotating coordinates in magnetic resonance problems. *Reviews of Modern Physics*, 26(2):167, 1954.

[39] J. J. Sakurai. *Modern Quantum Mechanics.* Addison-Wesley Publishing Company, revised ed edition, 1994.

[40] Christian Seis. Maximal mixing by incompressible fluid flows. *Nonlinearity*, 26(12):3279, 2013.

[41] R. Shankar. *Principles of quantum mechanics*. Springer Science & Business Media, 2nd edition, 1994.

[42] B. I. Bi Shraiman and E. D. Ed Siggia. Scalar turbulence. *Nature*, 405(6787):639–646, 2000.

[43] Jean-Luc Thiffeault. Using multiscale norms to quantify mixing and transport. *Nonlinearity*, 25(2):R1, 2012.

[44] Fredi Tröltzsch. *Optimal control of partial differential equations: theory, methods, and applications*, volume 112. American Mathematical Soc., 2010.

[45] A. Vikhansky. Enhancement of laminar mixing by optimal control methods. *Chemical Engineering Science*, 57(14):2719–2725, 2002.

[46] Cédric Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.

[47] A. Wirth and L. Biferale. Anomalous scaling in random shell models for passive scalars. *Physical Review E*, 54(5):4982, 1996.

[48] Michio Yamada and Koji Ohkitani. Lyapunov spectrum of a model of two-dimensional turbulence. *Physical Review Letters*, 60(11):983–986, 1988.

[49] Yao Yao and Andrej Zlatoš. Mixing and Un-mixing by Incompressible Flows. *J. Eur. Math. Soc. (JEMS). To appear. arXiv:1407.4163.*, page 37, 2014.