Adjoint-Based Mixing Enhancement for Binary Fluids

a thesis presented for the degree of

Doctor of Philosophy of Imperial College London

and the

Diploma of Imperial College

by

Maximilian F. Eggl

Department of Mathematics
Imperial College
180 Queen's Gate, London SW7 2BZ

June 2019

I certify that this thesis, and the research to which it refers, are the product of my own work, and that any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline.

Signed: ________________________________

Copyright

Thesis advisor: Professor Peter J. Schmid Maximilian F. Eggl

Adjoint-Based Mixing Enhancement for Binary Fluids

Abstract

Mixing is a fundamental fluid process that dominates a great many natural phenomena and is present in a wide variety of industrial applications. Therefore, studying the characteristics and optimisation of this process may lead to a significant impact in many fields. This thesis presents an analytical and computational framework for optimising fluid mixing processes using embedded stirrers based on a non-linear direct-adjoint looping approach. The governing equations are the non-linear Navier-Stokes equations, augmented by an evolution equation for a passive scalar, which are solved by a Fourier-based spectral method. Stirrers are embedded in the computational domain by a Brinkman-penalisation technique, and shape and path gradients for the stirrers are computed from the adjoint solution. The relationship between this penalisation approach and the adjoint will be examined through the derivation of a dual system of equations, and three different optimisation scenarios of increasing complexity, each focusing on different optimisation parameters, are considered. Within the limits of the parameterisations of the geometry and the externally imposed bounds, significant improvements in mixing efficiency are achieved in all cases.

To my Opa, without whom this PhD would never have happened.

To my Dad, because I always wanted to be an academic like him.

To Barbara, who I have always wanted to make proud.

Acknowledgments

Firstly, I would like to express my sincerest gratitude to my supervisor Prof. Peter Schmid for the continuous support of my PhD study and related research, for his patience, motivation, and immense knowledge. His insight and encouragement over many a pint in the octoberfestpub, has driven this PhD more than anything else. I feel so incredibly honoured to have been able to learn from and work with such an exceptional researcher and supervisor.

Furthermore, I extend my thanks to my PhD examiners, Professors Sergei Chernyshenko and Colm-cille Caulfield. Their understanding and willingness to find a suitable date for my viva with conflicting and hectic schedules, their friendliness during the viva itself and their invaluable contributions to the thesis itself have made this doctoral thesis the success it is.

I would like to also thank my office mates, particularly Jordan Hauge and Calum Skene, as they have dealt with my frustrations and complaints throughout these long three years. Their discussions and advice are what have kept me sane and going throughout this academic journey. Our lunches, drinks and geeky discussions will never be forgotten.

I have been incredibly lucky to have such a great set of friends, including ICSM Drama, Old Meadonians football club and the Chess Giris, and all have been incredible in their support. However, I would like to especially thank Takashi and Grace, who have supported me in my personal journey throughout this PhD, and helped me switch off from the mathematics on a regular basis.

To a very special person, Emma, who has dealt with the fallout of the writing process of this thesis, I would also like to extend a thank you. She has been invaluable in making this happen and always manages to put a smile on my face.

Last, but certainly not the least, I would like to thank my family: my mother, my father, my brother, my aunt and my uncle for their unending support and willingness to lend a hand to help me out. I cannot imagine being here today if it wasn't for them.

List of figures

2.1 Example of a studied system: A binary fluid encompassed by a moving circular wall and mixed by embedded stirrers 24

3.1 Velocity field at t = 0, 1, 5, 10, 30, 64. As expected at this Re value we observe the formation of vortices through the actions of the wall and cylinders. 51

3.2 Passive scalar simulation for $Re = 500$,$Pe = 166$ and $n = 200$. These figures were both taken at $t = 63$, and clearly numerical noise can be observed on the left. 53

3.3 Test of scalar field implementation for $t = 8, 16, 24$ and 32. 54

3.4 Graphical representation of the checkpointing procedure. One sparsely saved full forward sweep of `FLuSI` is completed before several subsimulations are executed and all time steps are saved for use in the adjoint system. 57

3.5 A visual representation of the non-linear adjoint loop. 58

4.1 Sketch of the initial configurations for the four test cases. Case 1 (top left) : one centred cylinder rotating about its centre. Case 2 (top right): two cylinders on the horizontal axis, rotating in opposite directions. Case 3 (bottom left): five rotating cylinders, placed such that three cylinders are aligned along the horizontal axis, while the remaining two are vertically offset. Case 4 (bottom right): one rotating cylinder moving from left to right with the velocity of the horizontal movement dictated by a cos-function. 66

4.2 Case 1: mixing optimisation using one stationary rotating stirrer. A highly penalised optimisation setting ($\lambda = 10^{-3}$ from equation (4.1)) has been used. (a) Rotational speed ω and axis a versus the number of direct-adjoint iterations. (b) Variance of the passive scalar versus time $t \in [0,\ T]$. 68

4.3 Case 1: mixing optimisation using one stationary rotating stirrer. (a) Variance, as defined in equation (2.81), of the scalar field θ versus time $t \in [0,\ T]$. (b) Zoomed-in view to illustrate the decrease in variance for the first eight iterations. (c) Contour of cylinder shapes as result of successive iterations. 70

4.4 Case 1: mixing optimisation using one stationary rotating stirrer. Left column: unoptimised configuration, with snapshots at $t = 8, 16, 24, 32$ (top to bottom). Middle Column: after four direct-adjoint optimisations, with snapshots at $t = 8, 16, 24, 32$ (top to bottom). Right column: after nine direct-adjoint optimisations, with snapshots at $t = 8, 16, 24, 32$ (top to bottom). For videos of these scenarios please refer to `1Before.mp4`, `1Intermediate.mp4` and `1After.mp4` for the left, middle and right column, respectively. 71

4.5 Case 2: mixing optimisation using two stationary rotating stirrers. A highly penalised optimisation setting ($\lambda = 10^{-3}$ from equation (4.1)) has been used. (a) Rotational speed ω and axis length a versus the number of direct-adjoint iterations for the first (left) cylinder. (b) Variance of the passive scalar versus time $t \in [0,\ T]$. 73

4.6 Case 2: mixing optimisation using two stationary rotating stirrers. Variance, as defined in equation (2.81), of the scalar field θ versus time $t \in [0,\ T]$. The solid lines represent a local optimum, where the left stirrer remains rather inactive. The dashed line represents an improved optimum, by mirroring the control variables onto the left stirrer before continuing the gradient-based optimisation. 74

4.7 Case 2: mixing optimisation using two stationary rotating stirrers. Left column: unoptimised configuration, with snapshots at $t = 8, 16, 24, 32$ (top to bottom). Middle column: after seven direct-adjoint optimisations, with snapshots at $t = 8, 16, 24, 32$ (top to bottom). Right column: Enforced minimum by mirroring the axis length across both cylinders at $t = 8, 16, 24, 32$. For videos of these scenarios please refer to `2Before.mp4`, `2After.mp4` and `2ImprovedMin.mp4` for the left, middle and right column, respectively. 75

4.8 Case 3: mixing optimisation using five stationary rotating stirrers. A highly penalised optimisation setting ($\lambda = 10^{-3}$ from equation (4.1)) has been used. (a) Rotational speed ω and axis a versus the number of direct-adjoint iterations for the first (left-most) cylinder. (b) Rotational speed ω and axis a versus the number of direct-adjoint iterations for the third (bottom) cylinder. (c) Rotational speed ω and axis a versus the number of direct-adjoint iterations for the fifth (centre) cylinder. (d) Variance of the passive scalar versus time $t \in [0,\ T]$. 77

4.9 Case 3: mixing optimisation using five stationary rotating stirrers. Variance, as defined in equation (2.81), of the scalar field θ versus time $t \in [0,\ T]$. 79

4.10 Case 3: mixing optimisation using five stationary rotating stirrers. Left column: unoptimised configuration, with snapshots at $t = 8, 16, 24, 32$ (top to bottom). Right column: after seven direct-adjoint optimisations, with snapshots at $t = 8, 16, 24, 32$ (top to bottom). For videos of these scenarios please refer to `5Before.mp4` and `5After.mp4` for the left and right column, respectively. 80

4.11 Case 4: mixing optimisation using one horizontally moving and rotating stirrer. Variance, as defined in equation (2.81), of the scalar field θ versus time $t \in [0,\ T]$. 82

4.12 Case 4: mixing optimisation using one horizontally moving and rotating stirrer. Left column: unoptimised configuration, with snapshots at $t = 8, 16, 24, 32$ (top to bottom). Right column: after five direct-adjoint optimisations, with snapshots at $t = 8, 16, 24, 32$ (top to bottom). For videos of these scenarios please refer to `CosBefore.mp4` and `CosAfter.mp4` for the left and right column, respectively. . . . 83

5.1 Unconstrained shape optimisation of two rotating stirrers. We can see that the left shape has generated two pinched off shapes throughout the optimisation. 90

5.2 Untwisting algorithm applied to the left shape from figure 5.1. The new curve is seen in red. 91

5.3 Application of thickening routine to untwisted shape. 92

5.4 Sketch of the initial configurations for the three shape optimisation cases. Case 1 (top left): one centred stirrer rotating about its centre. Case 2 (top right): two stirrers on the horizontal axis, rotating in the clockwise directions. Case 3 (bottom centre): five rotating stirrers, placed such that three cylinders are aligned along the horizontal axis, while the remaining two are vertically offset. 94

5.5 Case 1: mixing optimisation using one stationary rotating stirrer. No untwisting or thickening was applied. (a) Evolution of the shape throughout the optimisation steps. The red line is the initial configuration and green the final one. Arrows have been added in black to illustrate the changes in the shape. (b) Mix-norm of the passive scalar versus time $t \in [0,\ T]$. (c) Variance of the passive scalar versus time $t \in [0,\ T]$. 95

5.6 Case 1: mixing optimisation using one rotating stirrer. Left column: unoptimised configuration, with snapshots at $t = 2, 4, 6, 8$ (top to bottom). Right column: after eight direct-adjoint optimisations, with snapshots at $t = 2, 4, 6, 8$ (top to bottom). For videos of these scenarios please refer to `Shape1NoOpt.mp4` and `Shape1Opt.mp4` for the left and right column, respectively. 98

5.7 Case 2: mixing optimisation using two stationary rotating stirrers. (a) Evolution of the shape of the left stirrer throughout the optimisation steps. The red line is the initial configuration, green the final one and the blue dashed line refers to the thickening routine applied to the final optimised shape. Arrows have been added in black to illustrate the changes in the shape. (b) Evolution of the shape of the right stirrer throughout the optimisation steps. (c) Mix-norm of the passive scalar versus time $t \in [0,\ T]$. (d) Variance of the passive scalar versus time $t \in [0,\ T]$. 101

5.8 Case 2: mixing optimisation using two rotating stirrers. Left column: unoptimised configuration, with snapshots at $t = 2, 4, 6, 8$ (top to bottom). Right column: after 9 direct-adjoint optimisations, with snapshots at $t = 2, 4, 6, 8$ (top to bottom). For videos of these scenarios please refer to `Shape2NoOpt.mp4` and `Shape2Opt.mp4` for the left and right column, respectively. 102

5.9 Case 3: mixing optimisation using five stationary rotating stirrers. (a) Mix-norm of the passive scalar versus time $t \in [0,\ T]$. (b) Variance of the passive scalar versus time $t \in [0,\ T]$. 104

5.10 Case 3: mixing optimisation using five stationary rotating stirrers. (a) Evolution of the shape of stirrer 1 throughout the optimisation steps. The red line is the initial configuration, green the final one and the blue dashed line refers to the thickening routine applied to the final optimised shape. Arrows have been added in black to illustrate the changes in the shape. (b) Evolution of the shape of stirrer 2 throughout the optimisation steps. (c) Evolution of the shape of stirrer 5 throughout the optimisation steps. 105

5.11 Case 3: mixing optimisation using five rotating stirrers. Left column: unoptimised configuration, with snapshots at $t = 2, 4, 6, 8$ (top to bottom). Right column: after four direct-adjoint optimisations, with snapshots at $t = 2, 4, 6, 8$ (top to bottom). For videos of these scenarios please refer to `Shape5NoOpt.mp4` and `Shape5Opt.mp4` for the left and right column, respectively. 106

6.1 Sketch of time horizons for the optimisation problem. A control strategy (red) is applied over two control horizons, $T = 1$ (for short-time control) and $T = 8$ (for long-time control). The gradient information about the flow development (green), encoded in the adjoint variables, is gathered over a predictive horizon of $T = 8$. The final simulation, based on the optimised strategy, is performed over $T = 32$ non-dimensional time units (blue). 109

6.2 Various mixing strategies, from snapshots of the simulations. (a,b,c) Plunging of the stirrer through the interface, (d,e,f) casting of start-stop vortices towards the interface (vortex cannon), (g,h,i) collision of vortices, (j,k,l) collision with the vessel wall, and (m,n,o) breakup of vortical structures by stationary stirrers (obstruction). 112

6.3 Mixing optimisation based on only energy constraints for the stirrers. The time horizon for applying control is $T_{\text{control}} = 1$. Shown are iso-contours of the passive scalar at selected instances. The optimisation algorithm includes information over a time window of $T = 8$. For the video of this scenario please refer to `EnergyPen1.mp4`. . . . 114

6.4 Case 1: Energy penalised mixing optimisation under lesser control horizon $T_{\text{control}} = 1$. The black lines signify the control and information time horizons. (a) Mix-norm of the passive scalar versus extended time horizon $t \in [0,\ T = 32]$. (b)Variance of the passive scalar versus extended time horizon $t \in [0,\ T = 32]$. 115

6.5 Same as figure 6.3, but with an extended control window of $T_{\text{control}} = 8$. For the video of this scenario please refer to `EnergyPen8.mp4`. . . 116

6.6 Case 2: Same as figure 6.4, but with an extended control window of $T_{\text{control}} = 8$. 116

6.7 Mixing optimisation based on energy and velocity constraints for the stirrers. The time horizon for applying control is $T_{\text{control}} = 1$. Shown are iso-contours of the passive scalar at selected instances. The optimisation algorithm includes information over a time window of $T = 8$. For the video of this scenario please refer to `VelocityPen1.mp4`. 119

6.8 Case 3: Energy and velocity penalised mixing optimisation under lesser control horizon $T_{\text{control}} = 1$. The black lines signify the control and information time horizons. (a) Mix-norm of the passive scalar versus extended time horizon $t \in [0,\ T = 32]$. (b)Variance of the passive scalar versus extended time horizon $t \in [0,\ T = 32]$. 119

6.9 Same as figure 6.7, but with an extended control window of $T_{\text{control}} = 8$. For the video of this scenario please refer to `VelocityPen8.mp4`. 120

6.10 Case 4: Same as figure 6.8, but with an extended control window of $T_{\text{control}} = 8$. 120

6.11 Mixing optimisation based on energy, velocity and acceleration constraints for the stirrers. The time horizon for applying control is $T_{\text{control}} = 1$. Shown are iso-contours of the passive scalar at selected instances. The optimisation algorithm includes information over a time window of $T = 8$. For the video of this scenario please refer to `AccelerationPen1.mp4`. 122

6.12 Case 5: Energy, velocity and acceleration penalised mixing optimisation under lesser control horizon $T_{\text{control}} = 1$. The black lines signify the control and information time horizons. (a) Mix-norm of the passive scalar versus extended time horizon $t \in [0,\ T = 32]$. (b)Variance of the passive scalar versus extended time horizon $t \in [0,\ T = 32]$. 123

6.13 Same as figure 6.11, but with an extended control window of $T_{\text{control}} = 8$. For the video of this scenario please refer to `AccelerationPen8.mp4`.123

6.14 Case 6: Same as figure 6.12, but with an extended control window of $T_{\text{control}} = 8$. 124

List of tables

4.1 Consistency check based on the sign distribution of the control variable gradient: (left) based on a finite-difference approximation of the forward problem, (right) based on the adjoint system. 64

Contents

1 Introduction **17**

2 Mathematical Framework **23**

2.1 Governing equations and general assumptions 23

2.2 Fluid-solid interactions 25

2.3 Adjoint derivation 28

2.3.1 Cost functional 28

2.3.2 Penalisation and adjoint 29

2.3.3 Penalisation of adjoint derivation 29

2.3.4 Adjoint of penalised system 36

2.4 Shape parameterisation 45

2.5 Choosing a specific cost 46

2.5.1 Defining mixing 47

2.5.2 Mixing cost 47

3 Numerical Framework **49**

3.1 FLuSI and modifications 49

3.1.1 Mask implementation 50

3.1.2 Scalar field implementation 52

3.2 Adjoint implementation 55

3.2.1 Problem description 55

3.2.2 Backward-in-time integration 55

3.2.3 Technical considerations 56

3.2.4 Overview of non-linear adjoint looping 57

3.3 Code structure and algorithm 57
3.4 Stopping criteria 59
3.5 Summary 60

4 Results 1: Shape and rotation optimisation 61

4.1 Optimisation specifics 61
4.1.1 Optimisation parameters 62
4.1.2 Introduction of cost functional 62
4.2 Validation of gradient direction and optimisation results 63
4.2.1 A simple gradient check 63
4.2.2 Definition of test configurations 64
4.2.3 Case 1: one stationary rotating stirrer 65
4.2.4 Case 2: two stationary rotating stirrers 72
4.2.5 Case 3: five stationary rotating stirrers 76
4.2.6 Case 4: one horizontally moving, rotating stirrer 81

5 Results 2: Fourier-based shape optimisation 84

5.1 Optimisation specifics 84
5.1.1 Optimisation parameters 84
5.1.2 Shape parameterisation 84
5.1.3 Shape constraints 89
5.1.4 Introduction of the cost functional 91
5.2 Geometric configuration 92
5.3 Presentation of optimisation results 93
5.3.1 Case 1: Shape optimisation of one stirrer 93
5.3.2 Case 2: Shape optimisation of two stirrers 99
5.3.3 Case 3: Shape optimisation of five stirrers 103

6 Results 3: Optimisation of stirrer velocity along a path 107

6.1 Set-up of the problem 107

6.1.1 Path parameterisation and optimality condition 107

6.1.2 Cost functional and constraints 109

6.2 Presentation of optimisation results 110

6.2.1 Overview of test cases 111

6.2.2 Cases 1 and 2: optimisation under energy constraints 113

6.2.3 Cases 3 and 4: optimisation under energy and velocity constraints 117

6.2.4 Cases 5 and 6: optimisation under energy, velocity and acceleration constraints 121

7 Conclusion **125**

References **136**

Appendix A Arithmetic of Hadamard product **137**

Appendix B Full adjoint derivation of non-penalised governing equations **140**

Appendix C Full adjoint derivation of penalised governing equations **142**

Appendix D Backward time derivation **150**

1
Introduction

The mixing of binary fluids is a problem of fundamental concern in fluid dynamics, as its mechanisms play an important role in a wide variety of industrial fields and in many fluid processes encountered in daily life. Therefore, the study of this phenomenon has drawn great interest in the research community.

The study of mixing can broadly be split into two areas, theoretical and application-driven. The theoretical aspects of mixing have been studied in great detail, and there exists a vast body of literature pertaining to all aspects of mixing for binary fluids. Aspects that have been studied include the formulation of emulsions and the stirring mechanics used to create them[1], mixing in stratified flows[2,3] as well as the optimisation of mixing[4]. For example, Stokes mixing, i.e., the mixing of highly viscous fluids, has drawn great interest in the research community, exemplified by papers by Spencer *et al.*[5] who identified stretching and cutting or folding as the primary mechanisms of mixing, Chien *et al.*[6] and Ottino *et al.*[7], who found that chaotic mixing can be induced from laminar mixing by varying the oscillations of the walls, and Mohr *et al.*[8] who argued for a simplified analysis of viscous mixing by ignoring interfacial tension. Their simplification formed the basis for most subsequent developments in the mixing of viscous fluids. Concurrently, at the other end of the spectrum, turbulent mixing, i.e., the mixing of binary fluids mostly by inertial effects, has been studied extensively, such as in the articles by Corrsin *et al.*[9,10] and Batchelor[11]. More recently, mixing has been linked to chaotic processes and chaotic advection, as the accepted consensus is that turbulence naturally leads to 'better' mixing[12]. This consensus was subsequently studied widely and verified experimentally[13]. Thus, a trend has arisen where mixing is achieved by using techniques that inject energy into the system to generate turbulence and, therefore, achieve better mixing[14,15]. Turbulence has also been a subject of study in mixing of stratified fluids, e.g., Linden[16], who suggested that turbulence existed for all values of the ratio of buoyancy and flow shear and argued that there must be some overall stratification where mixing is most optimal. This was also studied by Tang *et al.*[17], who put a value on this optimal stratification and confirmed Linden's supposition.

Another area of particular interest, due to our reliance on this specific concept throughout this thesis, is the issue of choosing an appropriate mixing measure. This includes the early work by Danckwerts[18], who showed that mixtures can be expressed by two statistically-defined quantities, the scale and the intensity of segregation, by Tryggvason *et al.*[19] who utilised the stretching of the interface as a mixing measure, and, more recently, by Mathew *et al.*[20] who took inspiration from mathematical measure theory.

This small selection of articles is intended to demonstrate the incredible breadth of study that has gone into mixing, and we have by no means covered or exhausted the available literature. For those who wish to consult a more in-depth review of the different aspects of mixing theory, we can suggest the books by Uhl[21], Paul *et al.*[22] and Ottino[23].

These monographs and investigations focus on the more theoretical aspects of fluid mixing, but are less representative of the studies that concentrate on the applications of mixing in natural or technological phenomena. Samples of the latter category include the spreading and mixing of pollutants in the ocean[24], the ventilation of a building[25,26], the mixing of air and fuel for subsequent combustion[27] or the mixing in microfluidic devices[28,29]. Further fields of application include the food processing, pharmaceutical and consumer-product industry where even modest improvements of mixing efficiency would translate into immediate and significant cost-reductions as well as a more consistent quality of the final product.

Mixing in industrial applications is often accomplished by stirrers, i.e., moving bodies of a given shape embedded in a mixing vessel, whose task it is to produce long filaments[30] which subsequently diffuse. This two-step shearing-diffusion process, which is principally at the core of binary fluid mixing by stirrers, has been recognised as a fundamental mechanism[5] and has been studied extensively to gain insight and to guide control strategies. Following this strategy, we will consider in this thesis the case of stirrer-induced mixing, its challenges and its optimisation.

From a computational point of view, several difficulties arise from the treatment of the embedded stirrers, in particular the treatment of the fluid-structure interaction between these moving bodies and the binary fluid. Various options exist, and have been pursued by previous studies, such as body-fitted meshes[31] or fictitious domains[32]. However, these techniques are often restricted to simple configurations or are excessively costly due to the need to remesh at each time step (after the solid body has been moved). An attractive alternative to these methods is the Brinkman-penalisation approach which will be adopted in this thesis and will provide the necessary flexibility and efficiency to treat more complex geometries and flow configurations. Introduced by Arquis *et al.*[33], solutions obtained by the penalisation method have been rigorously shown to converge to the corresponding solutions of the Navier-Stokes equations[34] for the respective complex domain. Supporting studies, including an asymptotic analysis of this convergence process, are summarised in Liu *et al.*[35], and applications to high Mach-number flows[36] and turbulent flow past cylinders[37], among many other examples, have demonstrated the effectiveness and flexibility of this approach. Brinkman-penalisation has also been invoked by Chantalat *et al.*[38], coupled with a level-set technique to express optimal geometries, and applied by Bruneau *et al.*[39] to optimise the shape of stirrers. The appeal of this method lies in its simple derivation and straightforward numerical implementation.

A second difficulty arises when considering the choice of parameterisation for the stirrers. Even though the point-by-point definition of the penalisation method allows for an extensive and diverse number of solid shapes[40], this excessive dimensionality does not lend itself to realistic and applicable optimisation outcomes. The choice of reducing the dimensions of the design space alters the final optimisation result, as more exotic, yet optimal, shapes are excluded. Consequently, care must be applied in selecting a suitable parameterisation to represent the embedded stirrers. Examples of shape parameterisations are Bézier curves[41] or Hermite splines[42]. However, while these methods allow for a wide range of possible complex shapes, they quickly reach a break-even point between flexibility and computational cost. Instead, we turn to a Fourier-based shape parameterisation, also referred to as Fourier descriptors[43]. Relying on Fourier series, we are thus able to choose the dimensionality of the control space without substantially compromising the design space. The effectiveness of this approach will be demonstrated in a later section.

A further complication of the mixing problem lies in the non-linear nature of its underlying governing equations. Optimising stirring strategies using a gradient-based approach will have to cope with the solution of non-linear equations and the checkpointing problem for the dual/adjoint system. The complexity of the flow does not furnish equilibrium points about which to linearise, rather the full non-linear problem has to be tackled, and non-linear adjoint looping techniques[44] have to be employed for the stirrer geometry and/or stirrer paths.

With these challenges identified, and respective algorithms chosen, we can then attempt to enhance mixing via stirrer-based strategies using a non-linear direct-adjoint optimisation framework. Several previous studies have attempted the control and optimisation of mixing processes and these include, among others, optimal control of mixing via entropy maximisation of a flow governed by two orthogonal shear flows[45], and optimal mixing of binary fluids by optimising the mix-norm[46,47,48]. A concise review of measuring mixing as well as mixing optimisation is given in Ottino[49] as well as in Thiffeault[50]. The specific combination of Brinkman-style penalisation and adjoint techniques applied to the optimisation of mixing will be the methods of choice considered in this study. The same combination has been employed to address natural convection problems[51], applied to the Lattice-Boltzmann method[52] and used in the study of shape optimisation of square deferentially heated cavities[53,54], which most closely approaches the focus of this thesis; a pure application to mixing processes is lacking at this point.

A fundamental question of using this combined approach relates to the interchangeability of the numerical representation of the geometry by Brinkman-penalisation and the resulting adjoint variables. Specifically, we have to consider (i) the optimisation of the penalised dual (backward) system or (*ii*) the dual of the penalised forward Navier-Stokes equations. The former case retains the freedom of employing other fluid-solid interaction methods, while in the latter, the adjoint system naturally inherits complex-geometry boundary conditions without additional considerations for the derivation of the adjoint system.

In either case, we aim at a fully computational framework to inherit the flexibility, efficiency and accuracy of the fluid-structure treatment by Brinkman-penalisation combined with the effective convergence of the PDE-constrained optimisation method to reach a local minimum in a given mixing norm, and thus achieve a better mixed state of the binary system.

The considered test cases and their underlying geometries, namely a circular container with one or more embedded moving stirrers, are taken with a view to industrial applications, where cylindrical mixing containers with rotating stirrers in motion are commonplace[55]. Initially, we solely consider simple configurations within which the algorithm can prove its mettle and provide noticeable improvements in mixing efficiency. We then turn to including more complex shape parameterisations and time dependent optimisation parameters, where the framework successfully identifies strategies which greatly increase mixing in all cases.

The rest of this thesis is organised in the following manner; in chapter §2 we present the governing equations of the fluid system, in conjunction with the geometrical configuration and basic assumptions. Additionally, we consider the application of the Brinkman-penalisation method to these governing equations to model the fluid-solid interactions. This is followed by an in-depth discussion of the adjoint, including the two alternative adjoint derivations mentioned above. Furthermore, the numerical treatment of the governing equations is discussed, and we propose a formulation that is particularly amenable to deriving the dual/adjoint set of equations. Both the approaches introduce the concept of an augmented Lagrangian, albeit of slightly different forms. Next, we consider the parameterisation of the stirrer geometry together with our mixing system, before concluding the chapter by reviewing mixing measures. We then turn to the numerical algorithm and implementation thereof in chapter §3. We present the modifications that were required to adapt the open-source fluid-solid interaction code (`FLuSI`[56]) to fit the requirements of our study. We then proceed by discussing the implementation of the dual/adjoint system and combining it with the forward solver in a non-linear adjoint looping routine. Furthermore, we present a step-by-step summary of the optimal-mixing algorithm. In chapter §4, we demonstrate the efficacy of the algorithm by presenting test cases of initially simple geometries. The optimisation of the rotation and eccentricity of our stirrers are intended to validate our adjoint derivation and implementation. These examples have been chosen to probe and assess the convergence and optimisation behaviour of the algorithm, and results range from simple (and anticipated) to more difficult (and less intuitive). They are intended to gain experience with the optimisation strategy, which in turn will guide us in the following chapters where more complex setups are studied. The first study of these more complex geometries is presented in chapter §5, where we consider the optimisation of stirrer shapes. Due to the flexible nature of the parameterisation the results are less intuitive and less predictable, yet result in significant increase in mixing efficiency. Lastly, we present our most complex optimisation cases in chapter §6, namely the optimisation of a velocity profile along a given circular path. Despite the complexity of this system, our approach leads to significant results and much promise for future mixing optimisation endeavours. Conclusions are offered in chapter §7. The appendices will provide details on various derivations from the text.

2

Mathematical Framework

2.1 Governing equations and general assumptions

Throughout this thesis we will consider the mixing processes of a binary, miscible fluid. The governing equations below are sufficiently general to include non-Newtonian effects and some of the adjoint derivations will refer to these factors. The underlying governing equations that define the motion of our fluid are the incompressible Navier-Stokes equations, which in primitive form read

$$\begin{aligned} \partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p - Re^{-1} \nabla \cdot (f(\boldsymbol{u}) \nabla \boldsymbol{u}) &= 0, & (2.1a) \\ \nabla \cdot \boldsymbol{u} &= 0, & (2.1b) \end{aligned}$$

with $\boldsymbol{u}$ defined as the velocity vector, p as the pressure field and $f(\boldsymbol{u})$ generally denotes a tensor that can describe non-Newtonian effects. We note that if we set $f(\boldsymbol{u}) = 1$, we recover the Navier-Stokes equations for Newtonian fluids. We have stated the equations in non-dimensional form, using an appropriate characteristic length (L_0) and velocity (u_0) based upon which the dependent and independent variables are rendered dimensionless. This process introduces the non-dimensional fluid viscosity as the inverse Reynolds number with $Re = \frac{u_0 L_0}{\nu}$ and ν as the kine-

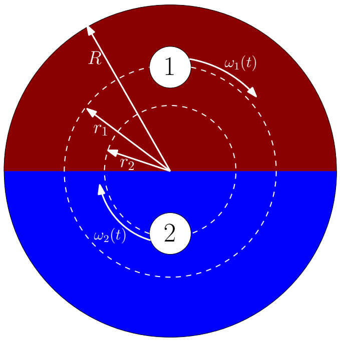


Figure 2.1: Example of a studied system: A binary fluid encompassed by a moving circular wall and mixed by embedded stirrers

matic viscosity.

To fully describe a mixing process an additional equation is introduced that tracks the two fluids (and their mixing) in the form of a passive scalar θ. This passive scalar, which takes on the value of $\theta = 0$ in fluid one, $\theta = 1$ in fluid two, and values in-between for a mixture of the two fluids, is governed by an advection-diffusion equation of the form

$$\partial_t \theta + \boldsymbol{u} \cdot \nabla \theta - Pe^{-1} \nabla^2 \theta = 0. \tag{2.2}$$

Another non-dimensional parameter, namely the Péclet number, Pe, is defined as $Pe = \frac{u_0 L_0}{\kappa}$ and represents the ratio of the rate of advection of the scalar field by the velocity field to the rate of diffusion, κ. The passive scalar will be used to track the composition of the binary fluid, to measure mixedness of our fluid and to subsequently design active stirring strategies to optimise this mixedness.

The mixing of the fluid will primarily be achieved using solid stirrers embedded within the fluid. To this end, we introduce the quantities, $\boldsymbol{x}_R$ and $\boldsymbol{x}_{r_i}$, which refer to the position of the outer wall and i embedded stirrers, respectively. With the introduction of these quantities, it is now possible to define boundary conditions for our scalar and velocity fields as follows

$$\begin{aligned}
\boldsymbol{u}(\boldsymbol{x}_R, t) &= \boldsymbol{u}_R, && (2.3a)\\
\boldsymbol{u}(\boldsymbol{x}_{r_i}, t) &= \boldsymbol{u}_{r_i}, && (2.3b)\\
\partial_R \theta(\boldsymbol{x}_R, t) = \partial_r \theta(\boldsymbol{x}_{r_i}, t) &= 0, && (2.3c)\\
\partial_R p(\boldsymbol{x}_R, t) = \partial_r p(\boldsymbol{x}_{r_i}, t) &= 0, && (2.3d)
\end{aligned}$$

i.e., we impose no-slip conditions on the boundaries of the solids as well as no out-flow for the scalar field. Furthermore, for the purpose of this study we impose the following initial conditions

$$\begin{aligned}
\boldsymbol{u}(\boldsymbol{x}, 0) &= 0, && (2.4a)\\
\theta(\boldsymbol{x}, 0) &= \theta_0(\boldsymbol{x}). && (2.4b)
\end{aligned}$$

where θ_0 is a step function with one in the upper half-plane and zero in the lower half-plane. An example set-up with two embedded stirrers moving on a circular path can be seen in figure 2.1.

2.2 Fluid-solid interactions

As stirrers will be used to achieve fluid mixing, our problem can be categorised as a fluid-structure interaction problem. To model the boundary conditions, as well as the interaction of the solids with the fluids, we will use a Brinkman-penalisation method[33]. Its appeal lies in the simple modification of the governing Navier-Stokes equations by adding external forcing terms. These terms model our solid bodies as Brinkman-style porous media with vanishing permeability C_η. This method has been shown to converge to the exact solid-fluid solution as C_η tends to zero[34]; furthermore, it is able to enforce Dirichlet as well as Neumann boundary conditions on the respective flow variables. The advantage of this method lies in its rather simple implementation, its flexibility in imposing complex boundary conditions, and its numerical efficiency. Moreover, moving solids are straightforwardly treated by remapping masks to a new position, without any need for remeshing or sophisticated grid operations. We will provide a brief overview here; for alternative applications of this method, or a more in-depth coverage of this method, the reader is referred to Schneider *et al.*[57].

In preparation for the fact that we will deal with multiple embedded solids with independent characteristics we will introduce χ_i, a mask function of the i-th solid, where a solid can be either the wall or a stirrer, by defining

$$\chi_i(\boldsymbol{x}, t) = \begin{cases} 1, & \text{if } \boldsymbol{x} \in \Omega_{s,i}, \\ 0, & \text{if } \boldsymbol{x} \in \Omega_f, \\ 0, & \text{if } \boldsymbol{x} \in \Omega_{s,j \neq i}, \end{cases} \tag{2.5}$$

with $\Omega_{s,i}$ denoting the i-th solid domain and Ω_f stands for the fluid domain. The global mask χ for our computational domain is then given as

$$\chi(\boldsymbol{x}, t) = \sum_i \chi_i(\boldsymbol{x}, t). \tag{2.6}$$

The mask χ acts as an indicator function which distinguishes between the solid part ($\chi = 1$) and the fluid part ($\chi = 0$) of the computational domain. This indicator function then allows us to supplement the Navier-Stokes equations (2.1) by external driving terms that impose the given velocity of the i-th solid, denoted by $\boldsymbol{u}_{s,i}$, on the fluid and thus model the motion of individual bodies through the fluid.

Assuming Einstein summation over identical indices, we can then state the penalised Navier-Stokes equations as

$$\begin{aligned} \partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \frac{\chi}{C_\eta} \boldsymbol{u} - \frac{\chi_i}{C_\eta} \boldsymbol{u}_{s,i} + \nabla p - Re^{-1} \nabla \cdot (f(u) \nabla \boldsymbol{u}) &= 0, && \text{(2.7a)} \\ \nabla \cdot \boldsymbol{u} &= 0, && \text{(2.7b)} \end{aligned}$$

where C_η is the permeability of the solids (which, for simplicity, we take identical for all solids). It has been shown[56] that an optimal value of C_η is proportional to $(\Delta x)^2$, and this leads to accurate numerical results. Recalling the definition of χ_i, we note that the equations above reduce to the Navier-Stokes equations (2.1) in the fluid domain ($\chi = 0$).

Proceeding to the governing equations for the passive scalar θ, we also have to apply penalisation terms to enforce no-flux boundary conditions (2.3) at the various solids. The scalar field equation (2.2) then becomes[58]

$$\partial_t\theta + (1-\chi)\,\boldsymbol{u}\cdot\nabla\theta + \chi_i\,(\boldsymbol{u}_{s,i}\cdot\nabla\theta) - \nabla\cdot\left(\left[Pe^{-1}\,(1-\chi) + \frac{\chi}{C_\eta}\right]\nabla\theta\right) = 0. \tag{2.8}$$

The terms $(1-\chi)\,\boldsymbol{u}\cdot\nabla\theta$ and $\nabla\cdot([Pe^{-1}\,(1-\chi)+\chi/C_\eta]\,\nabla\theta)$ prevent the passive advection or diffusion of the passive scalar field θ, respectively, into any of the solids; the term $\chi_i\,(\boldsymbol{u}_{s,i}\cdot\nabla\theta)$ transports the scalar field with the velocity of the i-th solid.

To complete the penalisation process, we need to define the solid properties of the stirrers, i.e., the solid velocity of the stirrers themselves. This is equivalent to the boundary conditions defined in (2.3). For the sake of brevity and consistency with the literature, we will replace $\boldsymbol{u}_r$ by $\boldsymbol{u}_s$ and define it as follows

$$\begin{aligned}(\boldsymbol{u}_{s,i})_1 &= \mathbf{1}(\omega_{C_i})_1 + \omega_i\boldsymbol{h}_1(\phi), && \text{(2.9a)}\\ (\boldsymbol{u}_{s,i})_2 &= \mathbf{1}(\omega_{C_i})_2 + \omega_i\boldsymbol{h}_2(\phi), && \text{(2.9b)}\\ \chi_i &= g_i(\boldsymbol{x},t), && \text{(2.9c)}\end{aligned}$$

where ω_{C_i} is the velocity of the centre of solid i, ω_i the rotational velocity of the solid i and ϕ the angle between $x - x_0$ (x_0 being the centre of the stirrer) and the horizontal. Lastly, we introduce the vector-valued function $\boldsymbol{h}$, since rotation is more conveniently defined in polar coordinates (yet we work in Cartesian coordinates). This transforms the radial velocity from one coordinate system to the other as follows

$$\begin{aligned}\boldsymbol{h}_1(\phi) &= \sqrt{(\boldsymbol{x}^2 - \boldsymbol{x}_0^2)}\,\sin\phi, && \text{(2.10a)}\\ \boldsymbol{h}_2(\phi) &= \sqrt{(\boldsymbol{x}^2 - \boldsymbol{x}_0^2)}\,\cos\phi. && \text{(2.10b)}\end{aligned}$$

2.3 Adjoint derivation

We begin by introducing the following inner products which will be used extensively throughout the derivation

$$||\boldsymbol{a}(\boldsymbol{x},t)||_2^2 = \frac{1}{\mathrm{Vol}_\Omega}\int_\Omega \boldsymbol{a}(\boldsymbol{x},t)^2 \ \mathrm{d}\Omega, \tag{2.11}$$

$$\langle a(\boldsymbol{x},t), b(\boldsymbol{x},t)\rangle = \int_0^T \int_\Omega a(\boldsymbol{x},t)^H b(\boldsymbol{x},t) \ \mathrm{d}\Omega\,\mathrm{d}t, \tag{2.12}$$

$$(c(\boldsymbol{x}), d(\boldsymbol{x})) = \int_\Omega c(\boldsymbol{x})^H d(\boldsymbol{x}) \ \mathrm{d}\Omega, \tag{2.13}$$

$$[e(x,t), f(x,t)]_R = \int_0^T \oint_{\delta_R} e(x,t)^H f(x,t) \ \mathrm{d}x\,\mathrm{d}t, \tag{2.14}$$

where Ω is our given domain and $(\cdot)^H$ refers to the conjugate transpose of the relevant matrix or vector.

2.3.1 Cost functional

The key to any optimisation problem is the successful choice of a given cost functional to be minimised or maximised. To begin, we will define our state and constraint vectors as follows

$$\boldsymbol{q}_i = \begin{pmatrix} \boldsymbol{u}_i \\ \theta \end{pmatrix}, \ \boldsymbol{s}_i = \begin{pmatrix} \boldsymbol{u}_{s,i} \\ \chi_i \end{pmatrix}. \tag{2.15}$$

Then we can introduce a cost functional of the following form

$$\mathcal{J} = g(\boldsymbol{q_i}, \boldsymbol{s_i}) \tag{2.16}$$

where g is some general function. We leave $\mathcal{J}$ purposely as general as possible, as we may choose to optimise with respect to different quantities and introduce a variety of constraints. We now turn to the definition of the adjoint variables and the derivation of their respective governing equations.

2.3.2 Penalisation and adjoint

Before we begin with the adjoint derivation there is one consideration we need to make, namely the treatment of boundary conditions in the adjoint process. There are two approaches we can take; the first of which involves deriving the adjoint of the unpenalised Navier-Stokes equations (2.1) with the corresponding boundary conditions (2.3). From there we can then apply the penalisation technique to these adjoint equations to model the interaction of the adjoint variables with the solid stirrers. This is an adjoint-first, penalisation-second approach.

On the other hand, we can consider the fully penalised Navier-Stokes equations (2.7) and (2.9), and derive the adjoint equations from these. In this case, the boundary conditions are already inherently embedded in the governing equations, and therefore by extension into the adjoint formulation. While this is very elegant and flexible, added complexity comes from the extra terms that need to be considered and treated. This is a penalisation-first, adjoint-second approach.

While we wish to emphasise that both approaches are valid and do not want to introduce a bias towards one or the other, we observed that the latter strategy (penalisation-first) was significantly easier and intuitive than the former (adjoint-first). The adjoint-first approach was derived and tested numerically, however there were complications that will be elaborated later on. However, we demonstrate the derivation nonetheless for completeness sake. Therefore, we begin by studying the derivation of the adjoint-first, penalisation-second approach.

2.3.3 Penalisation of adjoint derivation

The first step in the optimisation process is deriving the relevant adjoint equations from the governing equations (2.7). As the derivation does not change significantly we will choose the tensor f as the identity matrix, i.e., we consider a Newtonian fluid for this adjoint derivation.

At this point we wish to make a note; our cost functional $\mathcal{J}$ has been defined as generally as possible. However, depending on certain choices, it may only have exclusive dependencies on either $\boldsymbol{u}$ or θ, which means that explicit dependencies of some or all of our state variables might be missing from our cost functional.

The minimisation of our cost functional, $\mathscr{J}$, has to be accomplished subject to our state variables satisfying our governing equations. This PDE-based constraint can be achieved by embedding the governing equations as well as $\mathscr{J}$ in a single combined functional, i.e., an augmented cost functional $\mathcal{L}$. This augmented cost functional then takes the following form

$$\begin{aligned}\mathcal{L} = \mathscr{J} - &\langle \boldsymbol{u}^\dagger, \partial_t \boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nabla p - Re^{-1}\nabla^2\boldsymbol{u}\rangle - \langle p^\dagger, \nabla\cdot\boldsymbol{u}\rangle \\ &- \langle \theta^\dagger, \partial_t\theta + \boldsymbol{u}\cdot\nabla\theta - Pe^{-1}\nabla^2\theta\rangle - [\boldsymbol{u}_R^\dagger, \boldsymbol{u}(\boldsymbol{x}_R, t) - \boldsymbol{u}_R]_R \\ &- [\boldsymbol{u}_{r_1}^\dagger, \boldsymbol{u}(\boldsymbol{x}_{r_1}, t) - \boldsymbol{u}_{r_1}]_{r_1} - [\boldsymbol{u}_{r_2}^\dagger, \boldsymbol{u}(\boldsymbol{x}_{r_2}, t) - \boldsymbol{u}_{r_2}]_{r_2}. \end{aligned} \tag{2.17}$$

The above demonstrates that all governing equations and boundary conditions have been enforced using Lagrange multipliers denoted by $(\cdot)^\dagger$; these multipliers are the adjoint variables that will subsequently supply our optimisation with sensitivity information. The key to deriving the optimality conditions is then the minimisation of the augmented cost functional $\mathcal{L}$. This is accomplished by taking the first variation of (2.17) and setting it to zero, i.e., enforcing $\delta\mathcal{L} = 0$. However, as this term depends on several independent variables, we necessarily require all first variations with respect to all variables to be zero. In this vein, we note that the variation with regard to the adjoint variables recovers the governing equations, i.e., $\boldsymbol{u}^\dagger$ ensures the time evolution of $\boldsymbol{u}$, $p^\dagger$ enforces the continuity equation and $\theta^\dagger$ imposes the passive scalar equation. Lastly, the boundary conditions of the embedded wall and stirrers are ensured by the $\boldsymbol{u}_R$, $\boldsymbol{u}_{r_1}$ and $\boldsymbol{u}_{r_2}$ terms, respectively.

It is the variation of the state variables that leads to governing equations for the adjoint variables. Therefore, we begin by first considering the first variation with respect to $\boldsymbol{u}$

$$\begin{aligned}\left\langle \frac{\partial\mathcal{L}}{\partial\boldsymbol{u}}, \delta\boldsymbol{u}\right\rangle = \left\langle \frac{\partial\mathscr{J}}{\partial\boldsymbol{u}}, \delta\boldsymbol{u}\right\rangle - &\underbrace{\langle \boldsymbol{u}^\dagger, \partial_t\delta\boldsymbol{u} + \delta\boldsymbol{u}\cdot\nabla\boldsymbol{u} + \boldsymbol{u}\cdot\nabla\delta\boldsymbol{u} - Re^{-1}\nabla^2\delta\boldsymbol{u}\rangle}_{(\star)} \\ - \underbrace{\langle p^\dagger, \nabla\cdot\delta u\rangle}_{(\star\star)} - \underbrace{\langle\theta^\dagger, \delta u\cdot\nabla\theta\rangle}_{(\star\star\star)} &- [\boldsymbol{u}_R^\dagger, \delta u]_{\delta R} - [\boldsymbol{u}_{r_1}^\dagger, \delta u]_{\delta r_1} - [\boldsymbol{u}_{r_2}^\dagger, \delta u]_{\delta r_2}.\end{aligned} \tag{2.18}$$

As the starred terms are significant in length, we will consider them all individually before combining them in the final step. We will begin by considering $(\star)$ and, to make the ensuing derivation cleaner, we will move from a vector to component form. We use the Einstein summation convention of shared indices to recover the vector dot product. This results in

$$(\star) = \int_0^T \int_\Omega u_i^\dagger (\partial_t \delta u_i + \delta u_j \partial_j u_i + u_j \partial_j \delta u_i - Re^{-1} \partial_j \partial_j \delta u_i) \, \mathrm{d}\Omega \, \mathrm{d}t. \tag{2.19}$$

To derive the adjoint equation it is necessary to isolate δu in the expression above. This is accomplished by performing integration by parts across either the time or space integrals to finally obtain

$$\begin{aligned} &= (u_i^\dagger, \delta u_i)_{t=0}^{t=T} - \langle \partial_t u_i^\dagger, \delta u_i \rangle + \langle u_j^\dagger \partial_i u_j, \delta u_i \rangle + [u_i^\dagger u_j n_j^R, \delta u_i]_{\delta R} + [u_i^\dagger u_j n_j^{r_1}, \delta u_i]_{\delta r_1} \\ &\quad + [u_i^\dagger u_j n_j^{r_2}, \delta u_i]_{\delta r_2} - \langle u_j \partial_j u_i^\dagger, \delta u_i \rangle - [Re^{-1} u_i^\dagger n_j^R, \partial_j \delta u_i]_{\delta R} - [Re^{-1} u_i^\dagger n_j^{r_1}, \partial_j \delta u_i]_{\delta r_1} \\ &\quad - [Re^{-1} u_i^\dagger n_j^{r_2}, \partial_j \delta u_i]_{\delta r_2} + [Re^{-1} \partial_j u_i^\dagger n_j^R, \delta u_i]_{\delta R} + [Re^{-1} \partial_j u_i^\dagger n_j^{r_1}, \delta u_i]_{\delta r_1} \\ &\quad + [Re^{-1} \partial_j u_i^\dagger n_j^{r_2}, \delta u_i]_{\delta r_2} - \langle Re^{-1} \partial_j \partial_j u_i^\dagger, \delta u_i \rangle, \end{aligned} \tag{2.20}$$

where $n_j^\star$ is the outward pointing normal of solid $\star$. We can now collect all terms that contain the same inner products

$$\begin{aligned} &= (u_i^\dagger, \delta u_i)_{t=0}^{t=T} + \langle -\partial_t u_i^\dagger + u_j^\dagger \partial_i u_j - u_j \partial_j u_i^\dagger - Re^{-1} \partial_j \partial_j u_i^\dagger, \delta u_i \rangle \\ &\quad + [u_i^\dagger u_j n_j^R + Re^{-1} \partial_j u_i^\dagger n_j^R, \delta u_i]_{\delta R} + [u_i^\dagger u_j n_j^{r_1} + Re^{-1} \partial_j u_i^\dagger n_j^{r_1}, \delta u_i]_{\delta r_1} \\ &\quad + [u_i^\dagger u_j n_j^{r_2} + Re^{-1} \partial_j u_i^\dagger n_j^{r_2}, \delta u_i]_{\delta r_2} - [Re^{-1} u_i^\dagger n_j^R, \partial_j \delta u_i]_{\delta R} \\ &\quad - [Re^{-1} u_i^\dagger n_j^{r_1}, \partial_j \delta u_i]_{\delta r_1} - [Re^{-1} u_i^\dagger n_j^{r_2}, \partial_j \delta u_i]_{\delta r_2}. \end{aligned} \tag{2.21}$$

As the derivation of the other terms is done in an identical manner, we relegate the explicit calculations to appendix B and will just present the results of the rearrangement and simplifications here

$$(\star\star) = [p^\dagger n_i^R, \delta u_i]_{\delta R} + [p^\dagger n_i^{r_1}, \delta u_i]_{\delta r_1} + [p^\dagger n_i^{r_2}, \delta u_i]_{\delta r_2} - \langle \partial_i p^\dagger, \delta u_i \rangle, \tag{2.22}$$

$$(\star\star\star) = \langle \theta^\dagger \partial_i \theta, \delta u_i \rangle. \tag{2.23}$$

Substituting these results into (2.18) and collecting like terms we finally arrive at

$$\begin{aligned}\left\langle \frac{\partial \mathcal{L}}{\partial u_i}, \delta u_i \right\rangle = &\langle \frac{\partial \mathcal{J}}{\partial \boldsymbol{u}} + \partial_t u_i^\dagger - u_j^\dagger \partial_i u_j + u_j \partial_j u_i^\dagger + Re^{-1} \partial_j \partial_j u_i^\dagger + \partial_i p^\dagger - \theta^\dagger \partial_i \theta, \delta u \rangle \\ &- (u_i^\dagger, \delta u_i)_{t=0}^{t=T} - [u_i^\dagger u_j n_j^R + Re^{-1} \partial_j u_i^\dagger n_j^R + p^\dagger n_i^R + u_{R,i}^\dagger, \delta u_i]_{\delta R} \\ &+ [Re^{-1} u_i^\dagger n_j^R, \partial_j \delta u_i]_{\delta R} - [u_i^\dagger u_j n_j^{r_1} + Re^{-1} \partial_j u_i^\dagger n_j^{r_1} + p^\dagger n_i^{r_1} + u_{r_1,i}^\dagger, \delta u_i]_{\delta r_1} \\ &+ [Re^{-1} u_i^\dagger n_j^{r_1}, \partial_j \delta u_i]_{\delta r_1} - [u_i^\dagger u_j n_j^{r_2} + Re^{-1} \partial_j u_i^\dagger n_j^{r_2} + p^\dagger n_i^{r_2} + u_{r_2,i}^\dagger, \delta u_i]_{\delta r_2} \\ &+ [Re^{-1} u_i^\dagger n_j^{r_2}, \partial_j \delta u_i]_{\delta r_2}. \end{aligned} \tag{2.24}$$

As we wish to impose $\delta \mathcal{L} = 0$, we note that each of the terms under the different inner products must also equate to zero. Thus, the first line of (2.24) implies the following condition

$$\partial_t u_i^\dagger - u_j^\dagger \partial_i u_j + u_j \partial_j u_i^\dagger + Re^{-1} \partial_j \partial_j u_i^\dagger + \partial_i p^\dagger = \theta^\dagger \partial_i \theta - \frac{\partial \mathcal{J}}{\partial \boldsymbol{u}}. \tag{2.25}$$

In order to set the terms with $\partial_j \delta u_i$ to zero, we must enforce the following boundary condition

$$\boldsymbol{u}^\dagger = 0 \text{ on } \delta R, \delta r_1, \delta r_2. \tag{2.26}$$

Additionally, we observe that $p^\dagger$ takes the following value on the boundaries of the solids

$$p^\dagger n_i^r = -Re^{-1} \partial_j u_i^\dagger n_j^r - u_{r,i}^\dagger \text{ for } r = R, r_1, r_2. \tag{2.27}$$

We note that equation (2.27) can be rearranged to obtain $u_{r,i}^\dagger$ (the boundary adjoint velocity) on the boundaries

$$u_{r,i}^\dagger = -Re^{-1} \partial_j u_i^\dagger n_j^r - p^\dagger n_i^r \text{ for } r = R, r_1, r_2. \tag{2.28}$$

We observe that equation (2.27), and its rearranged form, contain two unknown quantities, $p^\dagger$ and $u_{r,i}^\dagger$, and as such the system is underspecified. We will need to impose some value for one of the two quantities to be able to calculate the other. However, as the quantity of interest is $u_{r,i}^\dagger$, and for which we do not wish to impose some value, we will choose to set $p^\dagger$. We will aim to mirror the system of equations from Foures *et al.*[59], who find the boundary adjoint velocity, $u^\dagger_{r,i}$, to be

$$u^\dagger_{r,i} = -Re^{-1}\partial_j u^\dagger_i n^r_j \text{ for } r = R, r_1, r_2. \tag{2.29}$$

This implies that we shall impose $p^\dagger = 0$ on the solid boundaries, but of course other values are possible and may lead to better solutions.

Lastly, we turn to the terms that are evaluated at specific times. These give us the final time condition for $\boldsymbol{u}^\dagger$ as follows

$$\boldsymbol{u}^\dagger(\boldsymbol{x}, T) = -\frac{\partial \mathcal{J}}{\partial \boldsymbol{u}|_T}. \tag{2.30}$$

Combining these boundary conditions with the governing equation (2.25) yields the full evolution equation for $\boldsymbol{u}^\dagger$. We now turn to the derivation of $p^\dagger$ by taking the first variation of $\mathcal{L}$ with respect to p

$$\left\langle \frac{\partial \mathcal{L}}{\partial p}, \delta p \right\rangle = -\langle \boldsymbol{u}^\dagger, \nabla \delta p \rangle \tag{2.31}$$

$$= -\int_0^T \int_\Omega u^\dagger_i \partial_i \delta p \; \mathrm{d}\Omega \, \mathrm{d}t \tag{2.32}$$

$$= -[u^\dagger_i n^R_i, \delta p]_{\delta R} - [u^\dagger_i n^{r_1}_i, \delta p]_{\delta r_1} - [u^\dagger_i n^{r_2}_i, \delta p]_{\delta r_2}, \delta p]_{\delta r_2} + \langle \partial_i u^\dagger_i, \delta p \rangle. \tag{2.33}$$

Based on our previous assumption that $\boldsymbol{u}^\dagger$ vanishes on the boundaries, we can see that this term simply reduces to

$$\nabla \cdot \boldsymbol{u}^\dagger = 0, \tag{2.34}$$

which mirrors the continuity equation of our forward velocity variable. To complete the set and arrive at the governing equation for $\theta^\dagger$ we turn to the first variation of $\mathcal{L}$ with respect to θ. The calculation does not differ in method from the previous two derivations, and so we will only present the resulting expression (the full derivation can be seen in appendix B)

$$\partial_t \theta^\dagger + u_i \partial_i \theta^\dagger + Pe^{-1} \partial_i \partial_i \theta^\dagger = -\frac{\partial \mathcal{J}}{\partial \theta}, \tag{2.35}$$

with boundary conditions

$$\nabla\theta^\dagger \cdot \boldsymbol{n} = \theta^\dagger = 0. \tag{2.36}$$

The terminal conditions for $\theta^\dagger$ are seen to be

$$\theta^\dagger(\boldsymbol{x}, T) = -\frac{\partial\mathcal{J}}{\partial\theta|_T}. \tag{2.37}$$

Therefore, if we summarise all computed evolution equations we have the following system governing the adjoint variables

$$\begin{aligned}
\partial_t u_i^\dagger - u_j^\dagger \partial_i u_j + u_j \partial_j u_i^\dagger + Re^{-1}\partial_j\partial_j u_i^\dagger + \partial_i p^\dagger &= \theta^\dagger\partial_i\theta - \frac{\partial\mathcal{J}}{\partial\boldsymbol{u}}, && (2.38a)\\
\nabla\cdot\boldsymbol{u}^\dagger &= 0, && (2.38b)\\
\partial_t\theta^\dagger + u_i\partial_i\theta^\dagger + Pe^{-1}\partial_i\partial_i\theta^\dagger &= -\frac{\partial\mathcal{J}}{\partial\theta}. && (2.38c)
\end{aligned}$$

We note that these equations mirror the governing equations (2.7), with similar forms of the momentum, continuity and passive scalar equations. The diffusion terms both in $\boldsymbol{u}^\dagger$ and $\theta^\dagger$ carry a positive sign, and therefore these equations can be seen as anti-diffusive. For the geometry pictured in figure 2.1 the complete boundary conditions are as follows

$$\begin{aligned}
\boldsymbol{u}^\dagger &= 0, && (2.39a)\\
u_{r,i}^\dagger &= -Re^{-1}\partial_j u_i^\dagger n_j^r, && (2.39b)\\
\nabla\theta^\dagger\cdot\boldsymbol{n_r} = \theta^\dagger &= 0. && (2.39c)
\end{aligned}$$

Finally, we have temporal conditions of the form

$$\begin{aligned}
\theta^\dagger(\boldsymbol{x}, T) &= -\frac{\partial\mathcal{J}}{\partial\theta|_T}, && (2.40a)\\
\boldsymbol{u}^\dagger(\boldsymbol{x}, T) &= -\frac{\partial\mathcal{J}}{\partial\boldsymbol{u}|_T}. && (2.40b)
\end{aligned}$$

Next, we derive the optimality conditions which will enable us to update our mixing strategies and obtain optimal ones. This is achieved by taking the first variation of the cost functional with respect to the control quantities, i.e., $\boldsymbol{u}_R^\dagger$ and $\boldsymbol{u}_{r_1,r_2}^\dagger$, leaving us with

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}_R} &= \boldsymbol{u}_R^\dagger, && \text{(2.41a)}\\
\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}_{r_1,r_2}} &= \boldsymbol{u}_{r_1,r_2}^\dagger. && \text{(2.41b)}
\end{aligned}
$$

There is no particular reason that these quantities should be zero. However, we can note[60]

$$
\begin{aligned}
\nabla_{\boldsymbol{u}_R} \mathcal{J} &= \frac{\partial \mathcal{L}}{\partial \boldsymbol{u}_R}, && \text{(2.42a)}\\
\nabla_{\boldsymbol{u}_{r_1,r_2}} \mathcal{J} &= \frac{\partial \mathcal{L}}{\partial \boldsymbol{u}_{r_1,r_2}} && \text{(2.42b)}
\end{aligned}
$$

and therefore we can define an expression for the cost functional with respect to the control variables, i.e.,

$$
\begin{aligned}
\nabla_{\boldsymbol{u}_R} \mathcal{J} &= \boldsymbol{u}_R^\dagger, && \text{(2.43a)}\\
\nabla_{\boldsymbol{u}_{r_1,r_2}} \mathcal{J} &= \boldsymbol{u}_{r_1,r_2}^\dagger. && \text{(2.43b)}
\end{aligned}
$$

By employing a gradient-based optimisation algorithm, we are therefore able to obtain optimal control strategies for these quantities and thus obtain a minimal value for $\mathcal{J}$ where the norm of the above gradient (2.43) vanishes.

With the full adjoint derivation now completed, we need to combine these equations with the Brinkman-penalisation framework introduced in equations (2.7). Using a similar style of analysis as in Angot *et al.*[34], we arrive at the following penalised equations

$$
\partial_t \boldsymbol{u}^\dagger - \boldsymbol{u}^\dagger \cdot \nabla \boldsymbol{u}^T + \boldsymbol{u} \cdot \nabla \boldsymbol{u}^\dagger + Re^{-1} \nabla^2 \boldsymbol{u}^\dagger + \nabla p^\dagger = \theta^\dagger \nabla \theta - \frac{\partial \mathcal{J}}{\partial \boldsymbol{u}} - \frac{\chi}{C_n} \boldsymbol{u}^\dagger, \quad \text{(2.44a)}
$$

$$
\partial_t \theta^\dagger + (1-\chi) \boldsymbol{u} \cdot \nabla \theta^\dagger + \chi (\boldsymbol{u}_s \cdot \nabla \theta^\dagger) + \nabla \cdot \left([Pe^{-1}(1-\chi) + \kappa \chi] \nabla \theta^\dagger \right) = 0, \quad \text{(2.44b)}
$$

where $\boldsymbol{u}^\dagger \cdot \nabla \boldsymbol{u}^T = u_j^\dagger \partial_i u_j$. Note that we do not include $\boldsymbol{u}^\dagger - \boldsymbol{u}_s$ in (2.44) as we are not interested in the adjoint velocity ,$\boldsymbol{u}^\dagger$, in the interior of the solid and thus extend the boundary condition (2.39), i.e., we impose $\boldsymbol{u}^\dagger = 0$ in the various solids. Similarly to the forward solution we adopt an operator splitting approach to ensure $\nabla \cdot \boldsymbol{u}^\dagger = 0$ by enforcing

$$\nabla^2 p^\dagger = \nabla \cdot \left(\boldsymbol{u}^\dagger \cdot \nabla \boldsymbol{u}^T - \boldsymbol{u} \cdot \nabla \boldsymbol{u}^\dagger + \theta^\dagger \nabla \theta - \frac{\chi}{C_n} \boldsymbol{u}^\dagger \right). \tag{2.45}$$

The full system above provides a basis in achieving optimal control of our chosen problem. However, we note that choosing the boundary condition of $p^\dagger = 0$ on the boundaries of the solids leaves ambiguities in the proper treatment of the solid stirrers within the adjoint framework. Furthermore, as no $\boldsymbol{u}_s$ terms appear in the above formulation of $\boldsymbol{u}^\dagger$, it is not clear what the relation of the adjoint quantity to the movement of the solids is. A full numerical version was implemented, however, it showed numerical instabilities near the boundaries. These instabilities may arise from the treatment of $p^\dagger$, but finding the optimal value of this quantity at the solid boundaries is not trivial.

Considering these caveats, we present the above derivation to illustrate the principles of the continuous adjoint as well as produce a set of equations that are independent of any numerical framework and that can be used, e.g., in a body-fitted mesh approach[61].

In what follows, we will turn to the penalisation-first, adjoint-second approach mentioned previously. This choice avoids any unnecessary assumptions on the boundary conditions; all conditions inherently encoded into the governing equations are transformed naturally into the adjoint framework.

2.3.4 Adjoint of penalised system

Discretising the governing equations

As mentioned previously, the starting point for this adjoint derivation are the penalised governing equations (2.7). We wish to avoid the component-wise analysis of the last section due to its complicated and cluttered nature. Instead, we intend to first spatially discretise our equations. To this end, we will replace the spatial derivatives with a multiplication of the discretised velocity, pressure and scalar fields by a Fourier spectral differentiation matrix. We begin by introducing the discrete analogue

$$\frac{\partial}{\partial \boldsymbol{x}_j} \rightarrow \mathsf{A}_j, \tag{2.46}$$

where A_j is a $n^2 \times n^2$ matrix where n is the number of grid points in a single dimension. More specifically, we define A_j in the following terms

$$\mathsf{A}_j = W^H i K_j W, \tag{2.47}$$

where W is the symmetric Fourier matrix, K is a diagonal matrix of associated wave numbers and $i = \sqrt{-1}$. To represent the associated gradient vector ∇, we introduce the matrix $\mathbf{A}= [\mathsf{A}_1, \mathsf{A}_2, \mathsf{A}_3]$. Furthermore, this results in the state variables being similarly discretised, i.e., $\boldsymbol{u}$ and $\boldsymbol{u}_{s,i}$ become $3 \times n^2$ matrices containing the fluid and solid velocity components, respectively. We note that $(\boldsymbol{u}_{s,i})_j$ is a $1 \times n^2$ vector and refers to the j^{th} velocity component of the i^{th} solid. χ, θ and p are $1 \times n^2$ vectors that represent the mask, scalar field and pressure, respectively. With these new terms, and assuming Einstein summation convention, we introduce the spatially discretised form of the governing equations

$$\partial_t \boldsymbol{u} + \boldsymbol{u}_j \circ [\mathsf{A}_j \boldsymbol{u}] + \frac{\chi}{C_\eta} \circ \boldsymbol{u} - \frac{\chi_i}{C_\eta} \circ \boldsymbol{u}_{s,i} + \mathbf{A} p - Re^{-1} \mathsf{A}_i \left[\mathsf{F}_{ij}(\boldsymbol{u}) \circ (\mathsf{A}_j \boldsymbol{u}) \right] = 0, \tag{2.48}$$

$$\mathsf{A}_i \boldsymbol{u}_i = 0, \tag{2.49}$$

$$\begin{aligned} \partial_t \theta - \mathsf{A}_i \left(\left[Pe^{-1} \left(\mathbf{1} - \chi \right) + \kappa \chi \right] \circ \mathsf{A}_i \theta \right) + (\mathbf{1} - \chi) \circ \boldsymbol{u}_j \circ [\mathsf{A}_j \theta] \\ + \chi_i \circ \left(\boldsymbol{u}_{s,i} \right)_j \circ [\mathsf{A}_j \theta] = 0, \end{aligned} \tag{2.50}$$

where $\circ$ is defined as the Hadamard element-wise product (see Horn *et al.*[62]). More details on specific calculations and relations of the Hadamard product can also been found in appendix A. The variable F_{ij} denotes the discrete version of the tensor describing non-Newtonian material behaviour. Similarly to the continuous case, we enforce the continuity equation by utilising an operator-splitting approach in addition to a pressure Poisson equation

$$\mathsf{A}_j\mathsf{A}_j p + \mathsf{A}_i\left(\boldsymbol{u}_j \circ [\mathsf{A}_j\boldsymbol{u}_i]\right) + \mathsf{A}_i\left[\frac{\chi}{C_\eta}\circ\boldsymbol{u} - \frac{\chi_i}{C_\eta}\circ\boldsymbol{u}_{s,i}\right] - Re^{-1}\mathsf{A}_i\left[\mathsf{F}_{ij}(\boldsymbol{u})\circ(\mathsf{A}_j\boldsymbol{u})\right] = 0. \tag{2.51}$$

Numerical implementation of masks

The representation of solid bodies on an underlying Cartesian grid calls for a transfer of geometric information onto the background mesh. This transfer is accomplished by a mollified delta-function, smoothing the otherwise discontinuous mask onto the grid and thus avoiding numerical inaccuracies and instabilities[63]. We choose the widely used piece-wise defined function

$$\chi_i(\mathbf{x},t) = \begin{cases} 1, & \text{for} \quad |f| < r_i, \\ \frac{1}{2}\left(1+\cos\left(\frac{\pi(f-r_i)}{2h}\right)\right) & \text{for} \quad r_i < |f| < r_i + 2h, \\ 0, & \text{otherwise,} \end{cases} \tag{2.52}$$

where f denotes a parametric representation for the solids and r_i is the distance from the centre of solid i to grid point $\boldsymbol{x}$. We note that h is proportional to the grid size, $\sqrt{\Delta x}$, and as $\Delta x \to 0$ the function χ_i tends to a Heaviside function[56].

Derivation of adjoint system

Returning to our state, $\boldsymbol{q}$, and constraint vectors, $\boldsymbol{s}$, as defined previously in (2.15), we can also introduce a spatially discretised cost functional of the form

$$\mathcal{J} = G(\boldsymbol{q}_i, \boldsymbol{s}_i) \tag{2.53}$$

where G is the spatially discretised version of the previous general function g. To include the dependencies of the various state variables on the optimisation variables, it is necessary to form an augmented Lagrangian as before. We do not need to include the boundary conditions as separate terms as these are implicitly encoded into the governing equations by our spatial discretisation. However, if we wish to optimise the shape, the velocity along a path and/or the rotational speed, these extra terms need to be incorporated into $\mathcal{L}$

$$\begin{aligned}\mathcal{L} = \mathcal{J} - \int_0^T & \underbrace{(\boldsymbol{u}^\dagger)_k^H \mathsf{M}\,\{\text{Equation (2.48)}\}_k}_{\star} + \underbrace{p^{\dagger,H} \mathsf{M}\,\{\text{Equation (2.51)}\}}_{\star\star} \\ & + \underbrace{\theta^{\dagger,H} \mathsf{M}\,\{\text{Equation (2.50)}\}}_{\star\star\star} + \omega_{C_i}^{\dagger,H} \mathsf{M}[\omega_{C_i} - w] + \omega_i^{\dagger,H} \mathsf{M}[\omega_i - z] \\ & + \chi_i^{\dagger,H} \mathsf{M}[\chi_i - g_i(\boldsymbol{x}, t)]\ \mathrm{d}t, \end{aligned} \tag{2.54}$$

where w, z and g_i are the given previous values of ω_{C_i}, ω_i and χ_i, respectively. M is a symmetric, positive-definite weight matrix taking into account grid resolution and spatial integration weights. As we are working on a equidistant grid of $n \times n$ mesh points, the weight matrix is simply a diagonal matrix of the form

$$\mathsf{M} = \frac{1}{n^2} I, \tag{2.55}$$

where I is the identity matrix. We note that the $\boldsymbol{u}^\dagger, p^\dagger$ and $\theta^\dagger$ variables perform the same role as in the previous derivation. The new terms $\omega_{C_i}^\dagger, \omega_i^\dagger$ and $\chi_i^\dagger$ express the initial conditions of the mask variables associated with these quantities.

We note that the optimality condition is obtained by taking the first variation of $\mathcal{L}$ with respect to all its dependent variables to zero. As the variation with respect to the adjoint variables recovers the governing equations, we will proceed to turn to the first variation of the state variables to obtain the evolution equations of the discretised adjoint variables. As there is no longer an explicit spatial dependence in this derivation, the isolation of the variational terms, $(\delta(\cdot))$, can be solely performed by vector manipulations and integration by parts for the temporal derivatives.

We will demonstrate the manipulation of the $\frac{\partial}{\partial \boldsymbol{u}}$ term as a representative example. We note that the mask variables have no explicit $\boldsymbol{u}$ dependence and thus vanish.

We are then left with

$$\int_0^T \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}_i}\right) \delta \boldsymbol{u}_i \ \mathrm{d}t = \int_0^T \left(\frac{\partial \mathcal{J}}{\partial \boldsymbol{u}_i}\right) \delta \boldsymbol{u}_i \ \mathrm{d}t - \int_0^T \left(\frac{\partial (\star)}{\partial \boldsymbol{u}_i}\right) \delta \boldsymbol{u}_i \ \mathrm{d}t \\ - \int_0^T \left(\frac{\partial (\star\star)}{\partial \boldsymbol{u}_i}\right) \delta \boldsymbol{u}_i \ \mathrm{d}t - \int_0^T \left(\frac{\partial (\star\star\star)}{\partial \boldsymbol{u}_i}\right) \delta \boldsymbol{u}_i \ \mathrm{d}t. \quad (2.56)$$

We follow the same strategy as the previous derivation by considering all terms separately, beginning with $(\star)$

$$\int_0^T \left(\frac{\partial (\star)}{\partial \boldsymbol{u}_i}\right) \delta \boldsymbol{u}_i \ \mathrm{d}t = \int_0^T \left(\frac{\partial}{\partial \boldsymbol{u}_i}\left\{(\boldsymbol{u}^\dagger)_k^H \mathsf{M}(\partial_t \boldsymbol{u} + \boldsymbol{u}_j \circ [\mathsf{A}_j \boldsymbol{u}] + \frac{\chi}{C_n} \circ \boldsymbol{u} + \frac{\chi_i}{C_n} \circ \boldsymbol{u}_{s,i} \right.\right. \\ \left.\left. + \mathbf{A}p - Re^{-1}\left[\mathsf{A}_i[F_{ij}(\boldsymbol{u}) \circ (\mathsf{A}_j \boldsymbol{u})]\right])\right\}\right) \delta \boldsymbol{u}_i \ \mathrm{d}t \quad (2.57)$$

$$= \int_0^T (\boldsymbol{u}_i^\dagger)^H \partial_t \delta \boldsymbol{u}_i + (\boldsymbol{u}_i^\dagger)^H \delta \boldsymbol{u}_j \circ [\mathsf{A}_j \boldsymbol{u}_i] + (\boldsymbol{u}_i^\dagger)^H \{\boldsymbol{u}_j \circ [\mathsf{A}_j \delta \boldsymbol{u}_i]\} \\ + (\boldsymbol{u}_i^\dagger)^H \left[\frac{\chi}{C_n} \circ (\delta \boldsymbol{u}_i)\right] - (\boldsymbol{u}_i^\dagger)^H Re^{-1} \mathsf{A}_l [F_{lj}(\boldsymbol{u}) \circ (\mathsf{A}_j \delta \boldsymbol{u}_i)] \\ - (\boldsymbol{u}_k^\dagger)^H \left[Re^{-1} \mathsf{A}_l \left[\left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial \boldsymbol{u}_i} \delta \boldsymbol{u}_i\right) \circ \mathsf{A}_j \boldsymbol{u}_k\right]\right] \ \mathrm{d}t. \\ (2.58)$$

The task is to isolate the first variation $\delta \boldsymbol{u}_i$ from all terms and transfer any operator acting on it onto the remaining terms of the scalar product. In this effort, we take advantage of the algebra for the Hadamard product, explained in appendix A. Continuing from above we obtain

$$= [\boldsymbol{u}_i^\dagger \delta \boldsymbol{u}]_0^T + \int_0^T -(\partial_t \boldsymbol{u}_i^\dagger)^H \delta \boldsymbol{u}_i + (\boldsymbol{u}_k^\dagger \circ [\mathsf{A}_j \boldsymbol{u}_k])^H \delta \boldsymbol{u}_j + \left(\mathsf{A}_j^H [\boldsymbol{u}_j \circ \boldsymbol{u}_i^\dagger]\right)^H \delta \boldsymbol{u}_i \\ - Re^{-1} \left[\mathsf{A}_j^H [F_{kj}(\boldsymbol{u}) \circ \mathsf{A}_k^H \boldsymbol{u}_i^\dagger]\right]^H \delta \boldsymbol{u}_i \\ - Re^{-1} \left[\left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial \boldsymbol{u}_i}\right)^H (\mathsf{A}_l^H \boldsymbol{u}_k^\dagger \circ (\mathsf{A}_j \boldsymbol{u}_k))\right]^H \delta \boldsymbol{u}_i \ \mathrm{d}t. \quad (2.59)$$

Next, we collect matching terms and gather them into their respective time domains to arrive at

$$
\begin{aligned}
= [\boldsymbol{u}_i^\dagger \delta \boldsymbol{u}]_0^T + \int_0^T \Bigg[&-(\partial_t \boldsymbol{u}_i^\dagger) + (\boldsymbol{u}_k^\dagger \circ [\mathsf{A}_i \boldsymbol{u}_k]) + \mathsf{A}_j^H [\boldsymbol{u}_j \circ \boldsymbol{u}_i^\dagger] + \frac{\chi}{C_n} \circ (\boldsymbol{u}_i^\dagger) \\
&- Re^{-1} \mathsf{A}_j^H [F_{kj}(\boldsymbol{u}) \circ \mathsf{A}_k^H \boldsymbol{u}_i^\dagger] - Re^{-1} \left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial \boldsymbol{u}_i} \right)^H (\mathsf{A}_l^H \boldsymbol{u}_k^\dagger \circ (\mathsf{A}_j \boldsymbol{u}_k)) \Bigg]^H \delta \boldsymbol{u}_i \, \mathrm{d}t.
\end{aligned}
\tag{2.60}
$$

We note that the calculation for the other two terms, $(\star\star)$ and $(\star\star\star)$, proceed in a similar fashion, and therefore we will relegate the explicit calculation to appendix C. Nevertheless, we will present the result of these two terms below

$$
\begin{aligned}
\int_0^T \left(\frac{\partial (\star\star)}{\partial \boldsymbol{u}_i} \right) \delta \boldsymbol{u}_i \, \mathrm{d}t = \int_0^T \Bigg[&\mathsf{A}_k^H p^\dagger \circ [\mathsf{A}_i \boldsymbol{u}_k] + \mathsf{A}_j^H [\boldsymbol{u}_j \circ \mathsf{A}_i^H p^\dagger] + \frac{\chi}{C_\eta} \circ \mathsf{A}_i^H p^\dagger \\
&- Re^{-1} \mathsf{A}_j^H (F_{kj}(\boldsymbol{u}) \circ \mathsf{A}_k^H \mathsf{A}_i^H p^\dagger) \\
&- Re^{-1} \left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial \boldsymbol{u}_i} \right)^H \mathsf{A}_l^H \mathsf{A}_k^H p^\dagger \circ (\mathsf{A}_j \boldsymbol{u}_k) \Bigg]^H \delta \boldsymbol{u}_i \, \mathrm{d}t,
\end{aligned}
\tag{2.61}
$$

$$
\int_0^T \left(\frac{\partial (\star\star\star)}{\partial \boldsymbol{u}_i} \right) \delta \boldsymbol{u}_i \, \mathrm{d}t = \int_0^T \left[(\mathbf{1} - \chi) \circ \theta^\dagger \circ [\mathsf{A}_i \theta] \right]^H \delta \boldsymbol{u}_i \, \mathrm{d}t. \tag{2.62}
$$

Substituting these terms into equation (2.56) and collecting like terms we arrive at

$$
\begin{aligned}
\int_0^T \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}_i} \right) \delta \boldsymbol{u}_i \, \mathrm{d}t = -[\boldsymbol{u}_i^\dagger \delta \boldsymbol{u}]_0^T - \int_0^T \Bigg[&- \left(\frac{\partial \mathcal{J}}{\partial \boldsymbol{u}_i} \right)^H - (\partial_t \boldsymbol{u}_i^\dagger) + (\boldsymbol{u}_k^\dagger \circ [\mathsf{A}_i \boldsymbol{u}_k]) \\
&+ \mathsf{A}_j^H [\boldsymbol{u}_j \circ \boldsymbol{u}^\dagger] + \frac{\chi}{C_n} \circ (\boldsymbol{u}_i^\dagger) - Re^{-1} \mathsf{A}_j^H [F_{kj}(\boldsymbol{u}) \circ \mathsf{A}_k^H \boldsymbol{u}_i^\dagger] \\
&- Re^{-1} \left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial \boldsymbol{u}_i} \right)^H (\mathsf{A}_l^H \boldsymbol{u}_k^\dagger \circ (\mathsf{A}_j \boldsymbol{u}_k)) + \mathsf{A}_k^H p^\dagger \circ [\mathsf{A}_i \boldsymbol{u}_k] \\
&+ [\boldsymbol{u}_j \circ \mathsf{A}_j]^H \mathsf{A}_i^H p^\dagger + \frac{\chi}{C_\eta} \circ \mathsf{A}_i^H p^\dagger - Re^{-1} \mathsf{A}_j^H (F_{kj}(\boldsymbol{u}) \circ \mathsf{A}_k^H \mathsf{A}_i^H p^\dagger) \\
&- Re^{-1} \left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial \boldsymbol{u}_i} \right)^H \mathsf{A}_l^H \mathsf{A}_k^H p^\dagger \circ (\mathsf{A}_j \boldsymbol{u}_k) \\
&+ (\mathbf{1} - \chi) \circ \theta^\dagger \circ [\mathsf{A}_i \theta] \Bigg]^H \delta \boldsymbol{u}_i \, \mathrm{d}t.
\end{aligned}
\tag{2.63}
$$

The optimality condition arises from the requirement that all these terms are equal to zero and, therefore, we can conclude the following

$$u_i^\dagger|_T = \left.\frac{\partial \mathcal{J}}{\partial \boldsymbol{u}_i}\right|_T, \tag{2.64}$$

and

$$\begin{aligned}\partial_t \boldsymbol{u}_i^\dagger - (\boldsymbol{u}_k^\dagger + \mathsf{A}_k^H p^\dagger) \circ [\mathsf{A}_i \boldsymbol{u}_k] - \mathsf{A}_j^H [\boldsymbol{u}_j \circ (\boldsymbol{u}_i^\dagger + \mathsf{A}_i^H p^\dagger)] - \frac{\chi}{C_n} \circ (\boldsymbol{u}_i^\dagger + \mathsf{A}_i^H p^\dagger) \\ + Re^{-1} \mathsf{A}_j^H \left(F_{kj}(\boldsymbol{u}) \circ \mathsf{A}_k^H \left(\boldsymbol{u}_i^\dagger + \mathsf{A}_i^H p^\dagger \right) \right) - (\mathbf{1} - \chi) \circ \theta^\dagger \circ [\mathsf{A}_i \theta] \\ + Re^{-1} \left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial \boldsymbol{u}_i} \right)^H (\mathsf{A}_l^H (\boldsymbol{u}_k^\dagger + \mathsf{A}_k^H p^\dagger) \circ (\mathsf{A}_j \boldsymbol{u}_k)) + \left(\frac{\partial \mathcal{J}}{\partial \boldsymbol{u}_i} \right)^H = 0.\end{aligned} \tag{2.65}$$

In the above expression there arises a recurring term, which we can replace to simplify (2.65), i.e., by defining $\Pi_i^\dagger = \boldsymbol{u}_i^\dagger + \mathsf{A}_i^H p^\dagger$, we arrive at

$$\begin{aligned}\partial_t \boldsymbol{u}_i^\dagger - \Pi_k^\dagger \circ [\mathsf{A}_i \boldsymbol{u}_k] - \mathsf{A}_j^H [\boldsymbol{u}_j \circ \Pi_i^\dagger] - \frac{\chi}{C_n} \circ \Pi_i^\dagger + Re^{-1} \mathsf{A}_j^H (F_{kj}(\boldsymbol{u}) \circ \mathsf{A}_k^H \Pi_i^\dagger) \\ + Re^{-1} \left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial \boldsymbol{u}_i} \right)^H (\mathsf{A}_l^H \Pi^\dagger \circ (\mathsf{A}_j \boldsymbol{u}_k)) - (\mathbf{1} - \chi) \circ \theta^\dagger \circ [\mathsf{A}_i \theta] + \left(\frac{\partial \mathcal{J}}{\partial \boldsymbol{u}_i} \right)^H = 0.\end{aligned} \tag{2.66}$$

The above expression states the temporal evolution equation for $\boldsymbol{u}^\dagger$. No assumptions had to be made regarding boundary conditions and the additional complexity due to the penalisation terms was tolerable in this derivation. Furthermore, since any information about the solids is implicitly encoded into the masks, we have arrived at a flexible formulation that is independent of the number of solids we choose to embed. Next, we turn to the variations with respect to p and θ to arrive at the remaining adjoint equations. The procedure for deriving these expressions is identical to the one discussed above, and therefore shall be omitted here (the full derivation can be found in appendix C). Nonetheless, we present the resulting equations for completeness sake

$$
\begin{align}
p\ &:\ \mathsf{A}_j^H\Pi_j^\dagger - \left(\frac{\partial \mathcal{J}}{\partial p}\right)^H = 0, \tag{2.67}\\
\theta\ &:\ \partial_t\theta^\dagger - \mathsf{A}_j^H[(\mathbf{1}-\chi)\circ \boldsymbol{u}_j \circ \theta^\dagger] - \mathsf{A}_j^H[\chi_i \circ (\boldsymbol{u}_{s,i})_j\ \ \circ \theta^\dagger] \nonumber\\
&\qquad + \mathsf{A}_i^H([Pe^{-1}(\mathbf{1}-\chi) + \kappa\chi]\circ \mathsf{A}_i^H\theta^\dagger) + \left(\frac{\partial \mathcal{J}}{\partial \theta}\right)^H = 0. \tag{2.68}
\end{align}
$$

Combining the above set of equation into one system, we arrive at the full adjoint system

$$
\begin{align}
&\partial_t\boldsymbol{u}_i^\dagger - \Pi_k^\dagger \circ [\mathsf{A}_i\boldsymbol{u}_k] - \mathsf{A}_j^H[\boldsymbol{u}_j \circ \Pi_i^\dagger] - \frac{\chi}{C_n}\circ \Pi_i^\dagger + Re^{-1}\mathsf{A}_j^H(F_{kj}(\boldsymbol{u})\circ \mathsf{A}_k^H\Pi_i^\dagger) \nonumber\\
&\quad + Re^{-1}\left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial \boldsymbol{u}_i}\right)^H (\mathsf{A}_l^H\Pi_k^\dagger \circ (\mathsf{A}_j\boldsymbol{u}_k)) - (\mathbf{1}-\chi)\circ\theta^\dagger \circ [\mathsf{A}_i\theta] + \left(\frac{\partial \mathcal{J}}{\partial \boldsymbol{u}_i}\right)^H = 0, \tag{2.69}\\
&\mathsf{A}_j^H\Pi_j^\dagger - \left(\frac{\partial \mathcal{J}}{\partial p}\right)^H = 0, \tag{2.70}\\
&\partial_t\theta^\dagger - \mathsf{A}_j^H[(\mathbf{1}-\chi)\circ \boldsymbol{u}_j \circ \theta^\dagger] - \mathsf{A}_j^H[\chi_i \circ (\boldsymbol{u}_{s,i})_j\ \ \circ \theta^\dagger] \nonumber\\
&\quad + \mathsf{A}_i^H([Pe^{-1}(\mathbf{1}-\chi) + \kappa\chi]\circ \mathsf{A}_i^H\theta^\dagger) + \left(\frac{\partial \mathcal{J}}{\partial \theta}\right)^H = 0, \tag{2.71}
\end{align}
$$

where $\Pi_i^\dagger$ is defined as before, and the final time conditions are defined by

$$
\boldsymbol{u}^\dagger(\boldsymbol{x},T) = \left.\frac{\partial \mathcal{J}}{\partial \boldsymbol{u}_i}\right|_T, \qquad \theta^\dagger(\boldsymbol{x},T) = \left.\frac{\partial \mathcal{J}}{\partial \theta}\right|_T. \tag{2.72}
$$

We see that the adjoint equations mirror the penalised governing equations we started with. In particular, $\Pi^\dagger$ can be regarded as an 'augmented' adjoint pressure, which satisfies the continuity equation. This term arises due to the operator splitting approach we utilise when solving the governing equations. We make one final observation before turning to the optimality conditions, namely the value of $\mathsf{F}_{kj}(\boldsymbol{u})$. If $\mathsf{F}_{kj}(\boldsymbol{u})$ is of the form such that $\nabla\cdot(\nabla^H f(\boldsymbol{u})\nabla)\boldsymbol{u} = 0$ (which arises from the right hand side of the Poisson equation) modifications are necessary as certain terms no longer exists. Therefore, we make the caveat that when $f(\boldsymbol{u})$ is of this form, the $\Pi^\dagger$ terms in equation (2.69) are replaced by $\boldsymbol{u}^\dagger$ (shown in red in the equation below) , as the non-Newtonian term in the pressure Poisson equation (2.51) is identically zero, i.e.,

$$\partial_t \boldsymbol{u}_i^\dagger - \Pi_k^\dagger \circ [\mathsf{A}_i \boldsymbol{u}_k] - \mathsf{A}_j^H[\boldsymbol{u}_j \circ \Pi_i^\dagger] - \frac{\chi}{C_n} \circ \Pi_i^\dagger + Re^{-1}\mathsf{A}_j^H(F_{kj}(\boldsymbol{u}) \circ \mathsf{A}_k^H \boldsymbol{u}_i^\dagger) \\ + Re^{-1}\left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial \boldsymbol{u}_i}\right)^H (\mathsf{A}_l^H \boldsymbol{u}_k^\dagger \circ (\mathsf{A}_j \boldsymbol{u}_k)) - (\boldsymbol{1} - \chi) \circ \theta^\dagger \circ [\mathsf{A}_i \theta] + \left(\frac{\partial \mathcal{J}}{\partial \boldsymbol{u}_i}\right)^H = 0. \tag{2.73}$$

With the adjoint equations in place, it remains to derive the optimality conditions to close our optimisation problem, i.e., the equations that define $\chi^\dagger, \omega^\dagger$ and $\boldsymbol{u}_C^\dagger$

$$\chi_i^\dagger = \left[\theta^\dagger \circ [\mathsf{A}_j \theta] - \frac{\Pi_j^\dagger}{C_\eta}\right] \circ (\boldsymbol{u}_j - (\boldsymbol{u}_{s,i})_j) + (\kappa - Pe^{-1})\mathsf{A}_j^H \theta^\dagger \circ \mathsf{A}_j \theta + \left(\frac{\partial \mathcal{J}}{\partial \chi_i}\right)^H, \tag{2.74}$$

$$(\omega_{C_i}^\dagger)_j = \chi_i^H \left(\frac{\Pi_j^\dagger}{C_n} - (\theta^\dagger \circ [\mathsf{A}_j \theta])\right) + \left(\frac{\partial \mathcal{J}}{\partial (\boldsymbol{u}_{s,i})_j}\right)^H, \tag{2.75}$$

$$\omega_i^\dagger = (\chi_i \circ \boldsymbol{h}_j(\phi))^H \left(\frac{\Pi_j^\dagger}{C_n} - (\theta^\dagger \circ [\mathsf{A}_j \theta])\right) + \left(\frac{\partial \mathcal{J}}{\partial (\boldsymbol{u}_{s,i})_j}\right)^H. \tag{2.76}$$

We have now presented the full optimisation framework for a geometrically penalised non-Newtonian fluid. In the context of this thesis we will use this set of equations to optimise mixing, but it is also possible to consider entirely different physical phenomena and their optimisation. We have purposely left the definition of $\mathcal{J}$ as general as possible as we want to demonstrate that the optimisation framework is largely independent of the chosen measures and constraints. Throughout this thesis, we choose a range of cost functionals that have been tailored to the quantities of interest.

When considering the above system of equations, we note that we have to simultaneously solve the direct and adjoint equations, as well as the optimality condition. Rather than following this procedure, it is customary to solve the direct and adjoint equations exactly, and to iterate on the optimality condition, until a user-specified criterion is satisfied. With this approach we use the gradient with respect to the control variables to advance the solution towards an optimum. We recall that our system is non-linear, which necessarily implies that we may not achieve convergence to a global optimum; instead, only a local optimum may be guaranteed.

As a last note we wish to state that we only considered Newtonian fluids in our computational experiments, i.e., $f(\boldsymbol{u}) = 1$. While the framework above is laid out for non-Newtonian effects, and indeed a computational implementation has been written, encountered challenges in verifying the accuracy of our results, primarily computational in nature, motivated the concentration on the Newtonian case for this thesis.

2.4 Shape parameterisation

We will induce mixing in our binary fluids by embedding moving stirrers of a given geometry. Even though our formalism allows for a point-by-point definition and manipulation of the stirrer geometry, we will instead choose to restrict our design space and, in view of equation (2.52), choose a set of low dimensional parameterisations that ensures that the problem remains numerically tractable. For verification purposes we will consider stirrers of elliptical cross section to illustrate the optimisation capabilities. Once this verification is complete, we will tackle more complex parameterisations. The elliptical cross sections are introduced by using the follow parameterisation

$$\begin{aligned} f &= \left[\left(\frac{(x-x_{0,i})\cos\alpha_i - (y-y_{0,i})\sin\alpha_i}{a_i}\right)^2 \right. \\ &\qquad \left. + \left(\frac{(x-x_{0,i})\sin\alpha_i + (y-y_{0,i})\cos\alpha_i}{b_i}\right)^2\right]^{\frac{1}{2}}, \end{aligned} \tag{2.77}$$

where $x_{0,i}$ and $y_{0,i}$ denote the centre of the elliptical solid i, a_i and b_i are the two perpendicular axes, respectively, and α_i is the angle of attack with respect to the horizontal coordinate direction. This parameterisation will yield a lower-dimensional version of general shape optimisation. It is possible to optimise the axes a_i and b_i independently, however, as we seek to enforce a constant cross-sectional area of our stirrers, we will enforce $a_i b_i = 1$. The above simplifications for the optimisation of the stirrer shape carries through to the optimality condition that furnishes, together with the optimisation routine, a new and improved geometry after each iteration. With the area constraint in place we only control the axis a_i and reformulate the cost functional gradient with respect to our (restricted) control variables, expressed in terms of the adjoint variable $a_i^\dagger$, as follows

$$\begin{aligned} a_i^\dagger &= \int_0^{T^F} \frac{\partial \mathcal{L}}{\partial a_i} \,\mathrm{d}t, \\ &= \int_0^{T^F} \frac{\partial \mathcal{L}}{\partial \chi_i} \frac{\partial \chi_i}{\partial f} \frac{\partial f}{\partial a_i} \,\mathrm{d}t. \end{aligned} \tag{2.78}$$

In the above expression the gradient $\dfrac{\partial \mathcal{L}}{\partial \chi_i}$ follows directly from equation (2.54). For the first of the two remaining terms we can see from equation (2.52) that

$$\frac{\partial \chi_i}{\partial f} = \begin{cases} 0, & |f(x, y, a_i, b_i)| < r_i, \\ -\dfrac{\pi}{4h} \sin\left(\dfrac{\pi(f - r_i)}{2h}\right), & r_i < |f(x, y, a_i, b_i)| < r_i + 2h, \\ 0, & \text{otherwise.} \end{cases} \tag{2.79}$$

The remaining derivative $\dfrac{\partial f}{\partial a_i}$ is conceptually straightforward but algebraically unwieldy

$$\frac{\partial f}{\partial a_i} = -\frac{(x - x_{0,i}) \cos \alpha_i - (y - y_{0,i}) \sin \alpha_i}{a_i^3 f}. \tag{2.80}$$

The above expressions can then be substituted into (2.78) to obtain a definitive expression for $a_i^\dagger$.

2.5 Choosing a specific cost

Up to this point the definition of our cost functional has been left rather general, partly to demonstrate the independence of the framework from a precise cost-functional, partly to illustrate the flexibility of the adjoint approach. However, as we are particularly interested in the mixing of binary fluids, it now becomes necessary to commit to some measure that allows us to optimise our mixing strategies. To this end, we consider a mathematical definition of mixing.

2.5.1 Defining mixing

Several papers have considered the design of effective mixing measures[50,60,64], where a study of energy norm, variance and mix-norm have been compared and evaluated. While all measures have been successful in generating efficient mixing strategies within an optimisation framework, the most effective has been found to be the mix-norm defined by

$$||(\theta)||_{MN} = \frac{1}{V_\Omega} \int_\Omega ||\nabla^k(\theta(\boldsymbol{x}, t))|| \mathrm{d}\Omega. \tag{2.81}$$

We note that $k = 0$ is equivalent to the variance of the passive scalar. The mix-norm, which is a Sobolev norm of fractional and/or negative index, was introduced in Mathew *et al.*[46] and has its mathematical origins in measure theory. The mix-norm, which requires k to be negative[50] to converge, focuses specifically on the amount of small scales in the passive scalar. There are two mechanisms which drive mixing, advection and diffusion, which are active and passive mechanisms, respectively. As we are only able to control advection, which ultimately affects diffusion, the seemingly most efficient way of optimising the diffusion process is by creating small scale and elongated filaments by exploiting advective processes. Once these filaments have been created, diffusion will be far more effective and rapid. Thus, the mix-norm appears to be a natural choice for our optimisation attempts. However, we wish to stress at this point that the choice of measure does not affect the optimisation framework, and no conceptual (but quantitative) changes would occur if, for example, the variance were chosen instead.

2.5.2 Mixing cost

The success of our optimisation routine will be dependent on our ability to maximise mixing in our passive scalar, which is equivalent to minimising our chosen norm, in this, case the mix-norm. To this end, we introduce a cost functional of the following form

$$\mathcal{J} = \frac{1}{V_\Omega} \int_\Omega ||\nabla^k \theta|| \mathrm{d}\Omega \bigg|_T + \text{ constraints}, \tag{2.82}$$

where T is the time horizon for our optimisation. We have left the constraints deliberately general, as these vary from case to case and will be defined more specifically later on. Similarly to the governing equations we need to spatially discretise (2.82). Therefore, we replace the ∇ operator by the discrete matrix A defined in (2.47) as follows

$$\mathcal{J} = \frac{[\mathsf{A}_i^k \theta]^H \mathsf{M} [\mathsf{A}_i^k \theta]}{V_\Omega} \bigg|_T + \text{ constraints}, \tag{2.83}$$

where M is the weight matrix defined previously. We note that the general constraints will not affect the adjoint fields $\boldsymbol{u}^\dagger$ and $\theta^\dagger$ but only the control parameters. Therefore, we are able to define explicitly the final time conditions from equations (2.72) for $\boldsymbol{u}^\dagger$ and $\theta^\dagger$ from the first term of the above

$$\boldsymbol{u}^\dagger(\boldsymbol{x}, T) = 0, \qquad \theta^\dagger(\boldsymbol{x}, T) = \frac{2}{V_\Omega} (\mathsf{A}_i^k)^H \mathsf{M} (\mathsf{A}_i^k \theta(\boldsymbol{x}, T)). \tag{2.84}$$

At this point we will elaborate on the control parameters we will use to optimise mixing strategies. The framework is sufficiently flexible to deal with a multitude of different control parameters, however, in the context of this thesis, we have focused on rotation, speed along a fixed path and shape. Each of the optimisation variables comes with a unique set of challenges to obtain a physically viable and realistic mixing strategy. To this end, relevant constraints will be introduced and discussed in the form of new cost functionals.

3

Numerical Framework

3.1 FLuSI and modifications

The starting point for the implementation of the penalised governing equations is the open-source software `FLuSI`[56], a Fourier pseudo-spectral code for fluid-structure interactions. Written in `FORTRAN 90`, it solves the three-dimensional, incompressible Navier-Stokes equations on an equi-spaced grid using a spectral formulation, adaptive time-stepping and a pressure-projection approach. The inclusion of solid bodies and complex computational geometries are treated by the Brinkman-type penalisation method discussed previously; sponge layers are utilised to handle open and outflow boundaries.

Originally `FLuSI` was written to study the dynamics of insect flight, focusing on the velocities and vorticities generated by the beating of the insect wing[65]. The strength of these results, the maturity of the code and the ability of `FLuSI` to deal with complicated geometries at high resolutions have been the reasons we chose this particular starting point. However, significant modifications had to be undertaken to adapt this code to our specifications. Firstly, a class of routines to more generally treat geometries, both for the computational domain and the moving stirrers, had to be added. Secondly, the temporal evolution of the passive scalar field, tracking the binary fluid, had to be added, validated and tested. Thirdly, and most importantly, an adjoint framework had to be implemented and verified.

3.1.1 Mask implementation

Due to the nature of our geometry we have two fundamental types of solids: (i) the outer wall of the cylinder which may rotate around the centre of the vessel, and (ii) the embedded stirrers, which may move through the fluid as well as rotate around their own centres. These two solids have different characteristics and behaviours as well as implementations.

We begin by parameterising the outer wall as a ring consisting of two circles of slightly different radii, i.e.,

$$\chi_{wall} = \begin{cases} 1, & r_1 < |f(x,y)| < r_2, \\ 0, & \text{else} = 0, \end{cases} \tag{3.1a}$$

where f is the parameterisation defined in equation (2.77), and r_1, r_2 represent the radii of the smaller and bigger circle, respectively. This generates an impermeable boundary that confines the binary fluid to a circular domain.

The second type of solids we consider are the embedded stirrers; these will perform the mixing of the fluids and, throughout this thesis, their paths, shape and rotation are the targets of optimisation. Therefore, it is imperative that their numerical implementation remains as flexible as possible with regards to their control parameters. To this end, a specific *solid_object* class has been created within the `FLuSI` framework.

This class is sufficiently flexible as stirrer-unique class variables can be assigned, e.g., location, speed of rotation and shape, and are easily accessible throughout the entire program. Furthermore, in future optimisation studies this class can be simply extended with a multitude of new parameters to generate adaptable and powerful objects, and thus increase the design space of mixing strategies. The flexibility of the number of stirrers is a further advantage of this approach.

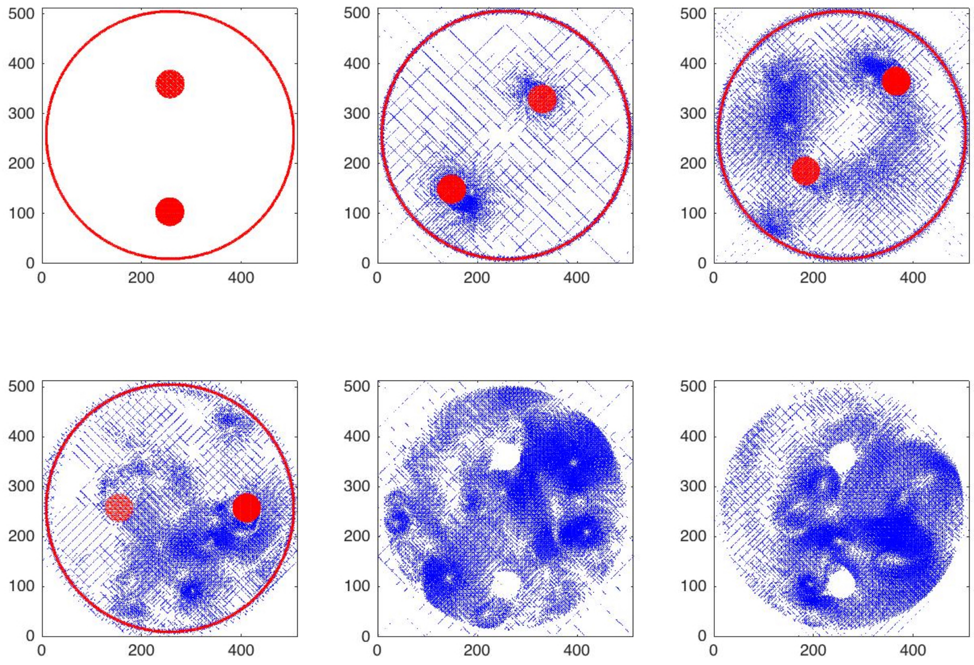


Figure 3.1: Velocity field at t = 0, 1, 5, 10, 30, 64. As expected at this Re value we observe the formation of vortices through the actions of the wall and cylinders.

A first implementation of this system can be seen in figure 3.1 where we have replicated the initial geometry shown in figure 2.1 with $Re = 1000$ and 512×512 grid points. The figure visualises the velocity field, shown in blue, at chosen time instances under the influence of stirrers and wall. The velocities of all solids, wall and stirrers, are shown in red. The wall and stirrers initially move clockwise on a circular path with a given angular velocity $u_s = \frac{2\pi}{8}$ from $t = 0$ until $t = 8$, after which they reverse and move anticlockwise with velocity $-u_s$. Finally at $t = 16$ the solid movement stops and the system eventually comes to rest.

With this initial benchmark, testing the flexible geometry given by the associated stirrer class, a first step has been taken within the larger optimisation framework. Next we turn towards the scalar field implementation and the introduction of a measurable quantity for our mixing.

3.1.2 Scalar field implementation

Even though a passive scalar implementation in `FLuSI` existed, the fact that it had been implemented using finite difference methods and ghost points did not lend itself to our uniform formulation using Fourier-based spectral methods. Therefore, we decided to implement an equivalent solver for the temporal evolution of the passive scalar equation using the pseudo-spectral approach as taken in the velocity counterpart. Again, gradient evaluation for a field $f(\boldsymbol{x})$ is accomplished using the Fourier transform given by

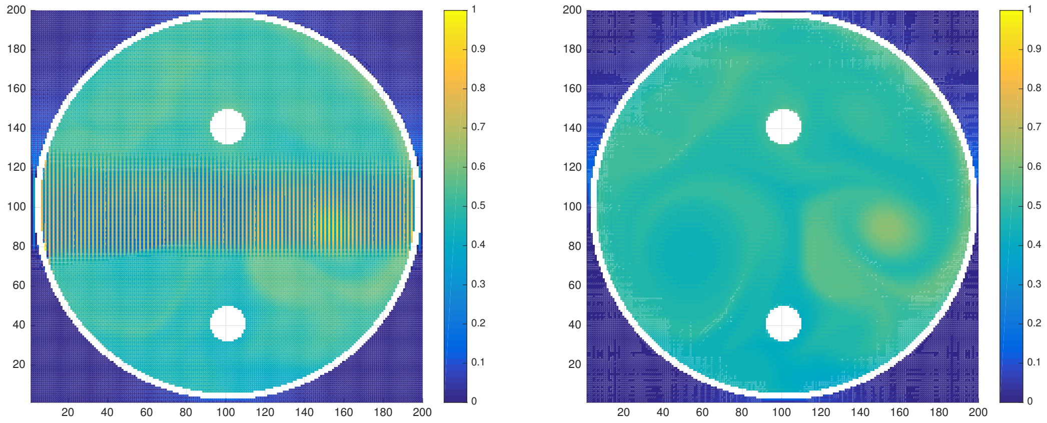
$$f'(\boldsymbol{x}) = \mathfrak{F}^{-1}\{i\boldsymbol{k} \cdot \mathfrak{F}(f(\boldsymbol{x})\}, \tag{3.2}$$

where $\boldsymbol{k} = (k_x, k_y, k_z)$ is the spatial wavenumber vector and $\mathfrak{F}$ denotes the three dimensional Fourier transform. Thus, for a given time-stepping scheme the spatial part of the right hand side of (2.8) can be rewritten as follows

$$[(1-\chi)\boldsymbol{u} + \chi_i \boldsymbol{u}_{s,i}] \cdot \mathfrak{F}^{-1}\left\{i\boldsymbol{k}\mathfrak{F}\{\theta\}\right\} + \mathfrak{F}^{-1}\left\{i\boldsymbol{k}\mathfrak{F}\left([Pe^{-1}(1-\chi) + \frac{\chi}{C_\eta}]\mathfrak{F}^{-1}\left\{i\boldsymbol{k}\mathfrak{F}\{\theta\}\right\}\right)\right\}. \tag{3.3}$$

As before, the boundary conditions are implicit in the solution, and we, therefore, retain flexibility across any number of solids embedded in the fluid.

To complete the scalar field implementation one last step is required, namely the application of dealiasing to the transported field. In simulations with extended time horizons, and/or in simulations above certain values of Re, Pe and resolution n, aliasing effects were observed in the passive scalar field, an example of which can be seen in figure 3.2a. As we utilise a pseudo-spectral approach, significant small scales in the passive scalar field are generated artificially which cause energy accumulation in the higher wavenumbers. This energy manifests itself as high-frequency numerical noise. This accumulation of energy can be remedied using dealiasing methods. `FLuSI` implements a traditional 2/3 dealiasing rule[66], where the highest third of the Fourier modes are set to zero. This approach causes the loss of a third of the spatial information and introduces a discrete cut off in the Fourier space. Nevertheless, it is effective in reducing the majority of the numerical aliasing error, but some energy remains in the highest wavenumbers and



(a) Simulation without dealiasing

(b) Dealiased simulation

Figure 3.2: Passive scalar simulation for $Re = 500, Pe = 166$ and $n = 200$. These figures were both taken at $t = 63$, and clearly numerical noise can be observed on the left.

continues to contaminate the temporal simulations. As an effective alternative, we use a high-order smoothing method introduced in Hou *et al.*[67] which involves the multiplication of the field of interest by a high-order exponential cut-off function. More specifically, we introduce a Fourier filter function ρ of the form

$$\rho(\boldsymbol{k}/N) = e^{-\alpha(\boldsymbol{k}/N)^m}, \tag{3.4a}$$

where N is the number of Fourier modes and $\alpha, m = 36$ have been chosen in accordance with the original paper. This method has been shown to be numerically stable while retaining a higher number of Fourier modes; the result of this methodology can be seen in figure 3.2b. In addition to its effectiveness, the implementation of the Fourier-filter technique is minimally invasive and thus convenient. All further simulations in this thesis utilise this dealiasing approach.

A first test of our passive scalar field implementation was performed with $Re = 1000, Pe = 1000, n = 512$ and the same stirring protocol (see figure 3.3). For the purpose of presentation the figure was additionally up-sampled to $n = 1024$ by zero-padding the higher wavenumbers (512 to 1024). A qualitative comparison was made with the original `FLuSI` approach, and satisfactory results were achieved.

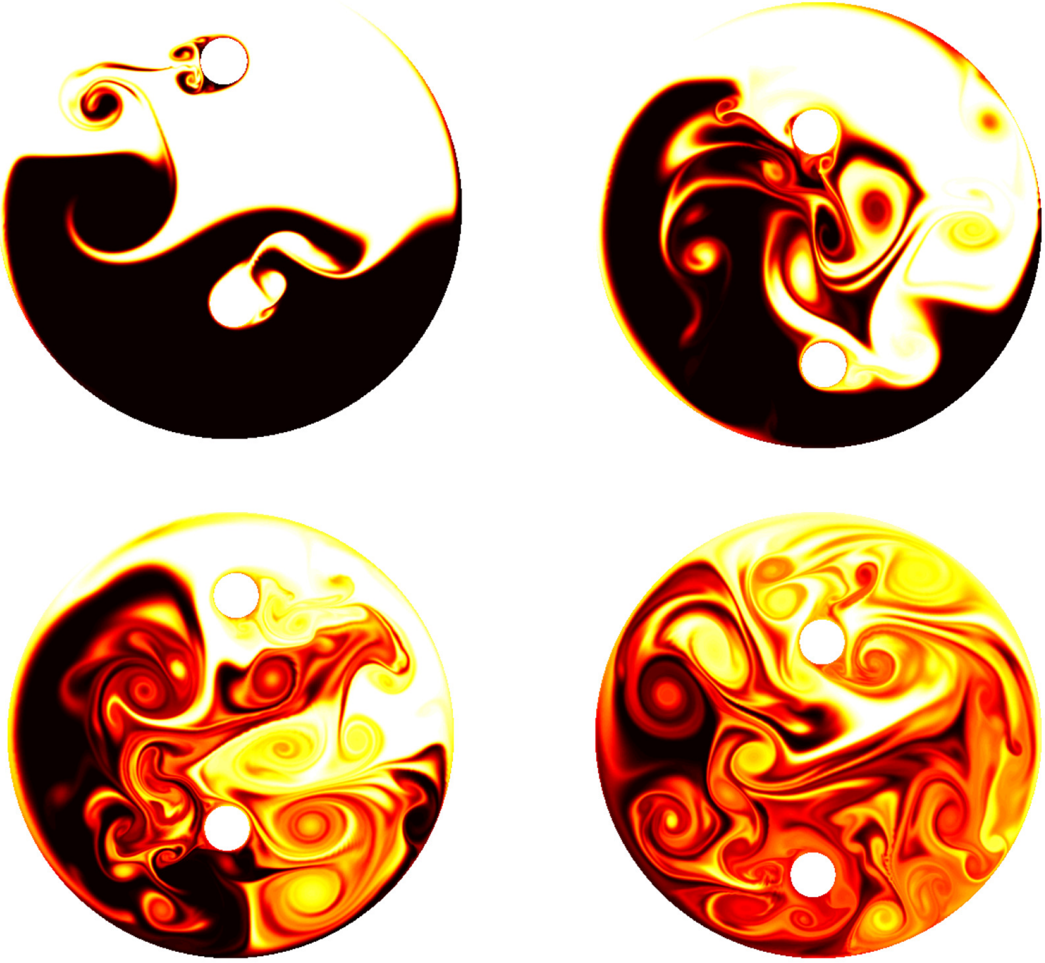


Figure 3.3: Test of scalar field implementation for $t = 8, 16, 24$ and 32.

3.2 Adjoint implementation

With the completion of the above modifications that allow `FLuSI` to accurately and efficiently simulate the mixing processes over a wide variety of geometries and strategies, we now turn our attention towards implementing the adjoint system.

3.2.1 Problem description

The necessary ingredients for the implementation of the adjoint solver consist of the following steps and mirror many features of the direct simulation. Among these features we have the generation of the mask equivalent to the forward run, the spatial discretisation of the adjoint equations (2.71), an appropriate time stepper that allows for backwards time stepping and, lastly, an efficient and accurate communication protocol between the forward state variables and the adjoint solver.

Both forward and adjoint solvers will lastly be part of an optimisation shell which accomplishes the convergence towards an optimal mixing strategy, and coordinates the communication between direct and adjoint solutions.

3.2.2 Backward-in-time integration

The choice of time-stepping algorithm does not affect the optimisation routine and was purely chosen for ease of use and efficiency. As the existing passive scalar routine in `FLuSI` was implemented with an Adams-Bashforth multi-step routine[68], we adopted the same routine for the adjoint solver. More specifically, we utilise the two-step Adams-Bashforth (AB2) with an Euler pre-step. In this case a temporal update with non-equidistant time steps[57] becomes

$$\boldsymbol{u}_{n+2} = \boldsymbol{u}_{n+1} + \beta_{11}\mathrm{RHS}_{u,n+1} + \beta_{10}\mathrm{RHS}_{u,n}, \tag{3.5}$$

where

$$\beta_{10} = \frac{1}{2}\frac{\Delta t_{n+2}}{\Delta t_{n+1}}(\Delta t_{n+2} + 2\Delta t_{n+1}), \tag{3.6}$$

$$\beta_{11} = -\frac{1}{2}\frac{\Delta t_{n+2}^2}{\Delta t_{n+1}}, \tag{3.7}$$

$$\Delta t_{n+1} = t_{n+1} - t_n. \tag{3.8}$$

For a backward-in-time integration using the AB2, the only modification consists of a change from positive to negative Δt, i.e.

$$\beta_{10} \to -\beta_{10}, \tag{3.9}$$

$$\beta_{11} \to -\beta_{11}. \tag{3.10}$$

A more detailed derivation of the above, and the treatment of the diffusive term using exponential time stepping, can be found in appendix D.

3.2.3 Technical considerations

It is important to realise that, owing to the non-linearity of the direct problem, there is an explicit dependence of the adjoint equations on the direct variables. For this reason we have to store the direct variables during the forward sweep and inject them, at the appropriate time steps, into the adjoint equations. For high-resolution cases and large time horizons we typically cannot afford to store all necessary direct variables, as memory requirements may exceed the available amount of physical storage. In this case we have to resort to checkpointing where we store relevant direct information only at specific checkpoints in time. In this manner we perform a full forward sweep of the governing equations, saving the appropriate fields at chosen checkpoints. During the backwards sweep the fields stored at these checkpoints act as initial conditions for additional simulations that reconstruct the flow fields at higher temporal resolution. Therefore, we trade memory restrictions for a (minor) increase in run-time. Throughout this thesis the checkpoints were placed in an equidistant manner, however, for more optimal placements of the checkpoints the reader is referred to the revolve package[69]. A graphical representation of the checkpointing process can be seen in figure 3.4,

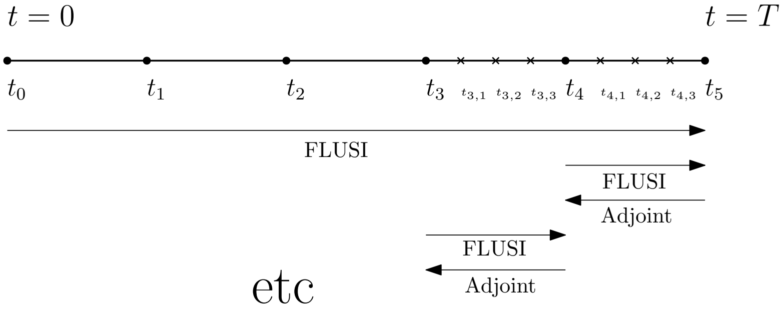


Figure 3.4: Graphical representation of the checkpointing procedure. One sparsely saved full forward sweep of `FLuSI` is completed before several sub-simulations are executed and all time steps are saved for use in the adjoint system.

including the initial forward `FLuSI` sweep, as well as the back-and-forth integration that recovers the high-fidelity fields.

3.2.4 Overview of non-linear adjoint looping

The steps above lead to the concept of non-linear adjoint looping. We begin with some initial condition for our mixing strategy. This strategy is implemented during the first solution of the non-linear governing equations over a time horizon from $t = 0$ to $t = T$. The information gathered from this sweep is then passed, via checkpointing, to the adjoint solver, which subsequently integrates backwards from $t = T$ to $t = 0$. The sensitivity information gained from this backwards run then enters the optimality conditions, which in turn are used to generate an improved mixing strategy for the next forward run. This forward and backward integration constitutes the adjoint direct loop and can be repeated until a user-defined criterion for the mixing strategy has been reached. At this point the loop iterations terminate, and the output is the user-defined optimal strategy. A visual representation of this procedure is displayed in figure 3.5.

3.3 Code structure and algorithm

The system of equations, reformulated as an iterative scheme, coupled with an optimisation strategy, completes the full algorithm for computing optimal mixing strategies.

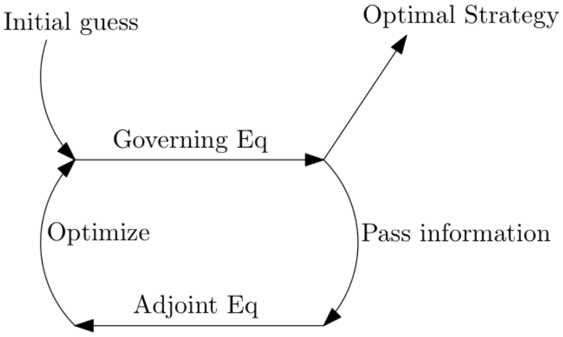


Figure 3.5: A visual representation of the non-linear adjoint loop.

The step-by-step procedure advances along the following lines, this time presented in algorithmic units.

1. We begin by running our forward simulation from $t = 0$ to $t = T$, solving our discretised and penalised governing equations (2.48) and (2.51). At this point, we have specifically chosen checkpoints, at which we save our state variables to disk. We aim to have sufficient checkpoints such that (i) the memory required to save all state variables to RAM does not exceed our resources, and (ii) efficiency of reading and writing to disk is ensured.

2. Once we have reached the endpoint of our simulation at $t = T$, we run our simulation forward from our final checkpoint, say t_n, to T. During this forward solution, we now ensure that we save the required state variables in RAM for each time step.

3. When we reach T, we start the adjoint simulation backwards in time from T to t_n. We have the relevant forward variables in RAM, and thus can feed them into the adjoint equations (2.71) at the appropriate time step.

4. Once we arrive at t_n with the adjoint simulation, we save the last state of the adjoint to RAM, making sure we have continuity in the adjoint variables across checkpoints. We then clear the memory and begin with running the forward simulation from t_{n-1} to t_n, once again saving flow fields for each time step to RAM.

5. We repeat steps 2-4, moving successively backwards in the checkpoints until we reach the initial starting time $t_0 = 0$. At this point, we evaluate the final

time conditions for the quantities we wish to optimise and supply these to our optimisation routine to generate the new (and improved) mixing strategy.

6. The direct-adjoint looping strategy is continued until a user-specified criterion is reached; at this time the simulations terminate.

3.4 Stopping criteria

A fundamental component of any optimisation attempt revolves around the user-defined criterion which terminates the optimisation routine. This stopping criterion is commonly associated with converging behaviours of the optimisation variables from update to update. However, if the design space is of a certain size, convergence can be difficult to achieve as too many different aspects affect the optimisation strategy. Therefore the termination of the optimisation is enacted even before a minimum has been reached, and as such we are performing an enhancement (or betterment) rather than an optimisation.

In the majority of our optimisation scenarios we are unable to achieve convergence in our optimisation strategies as our design space does not easily furnish such behaviour. Therefore, the criterion we set to decide when to terminate the optimisation are largely based on (i) realistic and feasible mixing strategies and (ii) the numerical stability of the algorithm. The former consideration comes into play when considering shape optimisation, as we do not wish for the solid shapes to intersect with each other. Therefore, when shapes are optimised in a direction that inevitably leads to collisions we terminate the optimisation routine before this stage. The second consideration, on the other hand, is more applicable to the velocity optimisation of the final case. The movement of the stirrers through the domain presents a significantly more difficult challenge to the adjoint algorithm, as the sensitivities produced can be quite considerable. Therefore, when the velocity profile of the stirrers reaches a critical point where the noise overwhelms any meaningful signal, the optimisation is once again terminated.

However, even though we have not reached a minimum in our mixing regimes, the enhancements we have achieved lead to a more mixed field that can lead to cost-saving efforts in the industry. A future endeavour will lie in designing the cost-functional so that a 'manual-handbrake' intervention from the user is not required and that convergence can be reached naturally.

3.5 Summary

In the sections above we have presented the numerical details of non-linear adjoint looping. The next step is to perform a verification of the adjoint implementation followed by the optimisation of user-specified control parameters of the stirrers.

4

Results 1: Shape and rotation optimisation

4.1 Optimisation specifics

As we now move into specific test cases, we will set the values for the Reynolds and Péclet numbers to $Re = 1000$ and $Pe = 1000$ for all following simulations. This parameter regime is particularly interesting as inertial effects can no longer be ignored, while turbulent fluid motion has not yet developed. The inertial features occurring in this parameter regime, described by a Reynolds number above the Stokes-flow limit, guarantee a rich and varied control space, taking advantage of advective, unsteady and diffusive processes, while the laminar aspect avoids numerical divergences of the direct-adjoint optimisation scheme due to the existence of positive Lyapunov exponents. Despite these restrictions, a great many natural and industrial mixing processes fall into our chosen parameter regime.

4.1.1 Optimisation parameters

The control parameters that are available to us in the optimisation of our system allow for an overwhelmingly large design space. While the framework is sufficiently flexible to accommodate various internal or external control parameters, we will first concentrate on time-independent variables, such as the shape, χ_i, of (multiple) stirrers via their eccentricity, a_i, and their rotational speed, ω_i. These quantities can be updated between iterations using the optimality conditions, but remain constant in time during the forward and backward simulations. This implies that expression (2.76) for $\omega_i^\dagger$ is modified by integration in time.

4.1.2 Introduction of cost functional

We next have to decide upon a suitable cost functional that also includes the constraints that still remain undefined in equations (2.83). Given that we will optimise the rotational velocity and eccentricity of the stirrers, it appears natural to impose limitations on their kinetic energy, as otherwise unrealistically high velocities might develop. To this end, we introduce a discretised cost functional of the following form

$$\mathcal{J} = \left.\frac{\theta^H \mathsf{M} \theta}{V_\Omega}\right|_{T^F} + \lambda \int_0^{T^F} \sum_i [(\boldsymbol{u}_{s,i})_j \chi_i]^H \mathsf{R}_i [(\boldsymbol{u}_{s,i})_j \chi_i] \; \mathrm{d}t, \tag{4.1}$$

where we choose R_i as a positive definite weight matrix yielding a valid norm. We point out that we have chosen the variance as our mixing measure for two purposes. Firstly, we wish to focus more on the verification of the framework than the final mixing result. Therefore, an intuitive and simple mixing measure appears advantageous. Secondly, we wish to demonstrate that the effectiveness of the optimisation framework is independent of the chosen measures.

We must emphasise that the choice of constraint may not be strictly industrially applicable either. When considering rotating stirrers, the power of the driving motor is a quantity that can, and should, be included in the cost-functional. The extra cost associated with moving a high-drag solid through the fluid should not be underestimated, and can consequently nullify any gains made in the more homogeneous final mixture. We note, however, that we wish, in this thesis, to consider the framework as a proof-of-concept and thus neglect constraining the more realistic power. Furthermore, the time dependency that would enter the cost-functional with power as a constraint complicates the optimisation problem significantly and thus, in line with previous studies that optimised similar systems and scenarios[4,59], turn to penalising the L^2-norm of the solid velocity of the stirrers. While the aforementioned studies do not include the movement of solid stirrers as mixing mechanisms, the ease of use and intuitive nature of this norm lends itself naturally to this problem. Furthermore, this quantity is comparable to the kinetic energy of the stirrers, and thus, we aim to thus retain physicality and realism. From this point onward when we constrain this quantity we will refer to this as the energy penalisation of the stirrers. We stress that future endeavours that wish to apply this optimisation framework to industrially interesting cases will need to include the power consumption as a constraint.

4.2 Validation of gradient direction and optimisation results

Results presented in this section have been reported in Eggl *et al.*[70].

4.2.1 A simple gradient check

Before embarking on various test cases for the optimisation of mixing strategies, we perform a consistency check of our adjoint framework. To this end, we consider a representative scalar control variable q, together with its adjoint equivalent $q^\dagger$. The optimality condition

$$\frac{\delta \mathcal{L}}{\delta q} = 0 \tag{4.2}$$

then establishes a link between the adjoint control variable $q^\dagger$ and the cost functional gradient $\delta \mathcal{J}/\delta q$. We have

$$\operatorname{sgn}(q^\dagger) = \operatorname{sgn}\left(-\frac{\delta \mathcal{J}}{\delta q}\right), \tag{4.3}$$

	$a+\epsilon$	$a-\epsilon$
$\omega+\epsilon$	$+$	$+$
$\omega-\epsilon$	$-$	$-$

	$\epsilon a^\dagger$	$-\epsilon a^\dagger$
$\epsilon\omega^\dagger$	$+$	$+$
$-\epsilon\omega^\dagger$	$-$	$-$

Table 4.1: Consistency check based on the sign distribution of the control variable gradient: (left) based on a finite-difference approximation of the forward problem, (right) based on the adjoint system.

which we will use to test our direct-adjoint system. More specifically, we evaluate the right-hand side by a finite-difference approximation according to

$$\frac{\delta \mathcal{J}}{\delta q} \approx \frac{\mathcal{J}(q+\epsilon)-\mathcal{J}(q)}{\epsilon}, \tag{4.4}$$

for a small value of ϵ. This expression is based on the forward problem only and is then contrasted to the value $q^\dagger$ which stems from the adjoint system. While we will not be able to make a quantitative comparison between the two expressions, we can match the sign-distributions of the various gradients and thus map out the gradient landscape. In other words, we evaluate the consistency relation

$$\operatorname{sgn}\left(q^\dagger \cdot \epsilon\right) = \operatorname{sgn}(\mathcal{J}(q) - \mathcal{J}(q+\epsilon)). \tag{4.5}$$

The results of this check are listed in table 4.1, where we show that the sign combinations across all scenarios (using the axis a and the rotational speed ω as control variables in our case) match accordingly. This test verifies the correct directionality of the adjoint-based gradient which, in turn, will ensure an improved mixing strategy from iteration to iteration.

4.2.2 Definition of test configurations

To further validate the direct-adjoint optimisation framework for enhancing mixing, we choose a suite of test problems that progressively challenge the computational procedure but still comply with our intuition for an optimal solution. The purpose of this exercise is less in treating a physically interesting case, but rather in demonstrating the efficiency and effectiveness of the adjoint-based optimisation scheme.

In each chosen scenario all involved stirrers begin as circular cylinders with unit radius and an initial rotational speed of $\frac{2\pi}{8}$ (either in the clockwise or anti-clockwise direction). The bounding vessel has a radius $R = 10$. The various configurations, ranging from a stationary rotating cylinder to five rotating cylinders, to a moving and rotating cylinder, are sketched in figure 4.1. The time horizon for the optimisation is $T = 32$; and the penalisation term C_η has been set to $C_\eta = 0.001$, – a value chosen in accordance with similar scenarios as present in `FLuSI`. The passive scalar θ is initially stratified with $\theta = 1$ in the upper half of the cylindrical domain Ω and $\theta = 0$ in the lower half. In all cases, we will present the results for a weakly and highly penalised optimisation setting (choosing the tuning parameter λ, arising from equation (4.5)).

4.2.3 Case 1: one stationary rotating stirrer

We commence by considering the case of a single, rotating (initially circular) stirrer located in the centre of a circular vessel. We optimise the rotational speed as well as the stirrer shape which we assume generally elliptical. By lengthening or shortening the stirrer's axes and increasing the speed at which the stirrer spins, we seek to enhance the mixing efficiency (measured by the variance of the passive scalar) over a given time interval. We place limitations on the shape of the stirrer by holding constant its cross-sectional area and on the speed of the rotation by capping the maximum kinetic energy, by limiting the L^2-norm of $\boldsymbol{u}_s$, injected into the mixture. We note that the shape of the stirrer directly affects the energy transferred into the fluid, as highly eccentric shapes require a larger input effort, but simultaneously may yield improved mixing.

This case serves as a first benchmark for the direct-adjoint optimisation framework; in particular, we wish to gauge the convergence behaviour, probe the influence of the penalisation parameter, and assess the physical fidelity of the obtained solution.

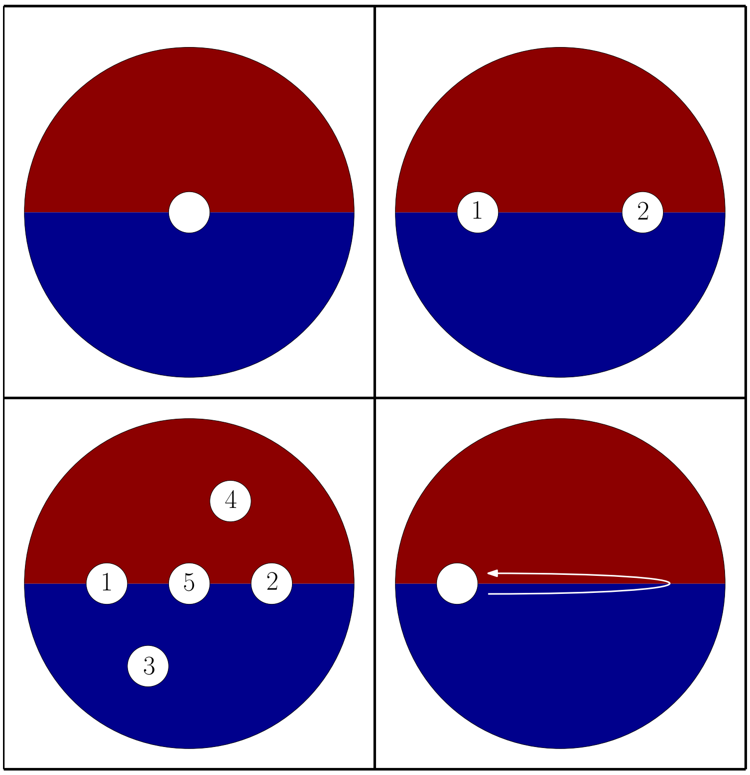


Figure 4.1: Sketch of the initial configurations for the four test cases. Case 1 (top left) : one centred cylinder rotating about its centre. Case 2 (top right): two cylinders on the horizontal axis, rotating in opposite directions. Case 3 (bottom left): five rotating cylinders, placed such that three cylinders are aligned along the horizontal axis, while the remaining two are vertically offset. Case 4 (bottom right): one rotating cylinder moving from left to right with the velocity of the horizontal movement dictated by a $\cos$-function.

Highly penalised system

As a proof of concept, we present the results of our optimisations with a rather high penalisation parameter of $\lambda_1 = 10^{-3}$, while retaining the same time horizon of $T = 8$. As expected, the optimisation increases the rotational velocity and the eccentricity of the stirrer. The progression in these control parameters versus the number of iterations is shown in figure 4.2a where we observe a monotonic shift towards more eccentric shapes (in red) and a gradual increase in the rotation speed ω (in blue). After six iterations, convergence is achieved, caused by the input energy penalisation.

Considering the evolution of the variance over the chosen time interval $t \in [0,\ T]$, we observe a clustering of the various iterations; it appears that an over-penalisation has allowed only marginal improvement in mixing efficiency. This matches the convergence of the shape and rotational speed variables in figure 4.2a.

From a physical point of view, the marginal mixing improvement can be attributed to the inability of the (penalised) stirrer to produce small-scale structures which could be dissipated or to induce significant advective mixing processes. Instead, after nine iterations we remain within the solid-body rotation regime, an example of which is shown in the left column of figure 4.4. The lack of small-scale structures renders diffusion ineffective, and explains the disappointing decrease in variance; the final mixing is solely due to the diffusion of an extended fluid interface created by the faster spinning (near-)cylinder.

Weakly penalised system

To induce more effective mixing, the system must be able to use advective processes to create small-scale structures and filaments which then give rise to significant mixing by diffusion. The highly penalised system of the previous section does not venture into the proper parameter regime to encourage this behaviour. For this reason, we drastically lower the penalisation value to $\lambda = 10^{-4}$.

When considering the variances that follow from each direct-adjoint iteration in figure 4.3a, we note a clustering over the first optimisations, similar to the highly penalised approach. In this range, we still remain in the solid-body regime and

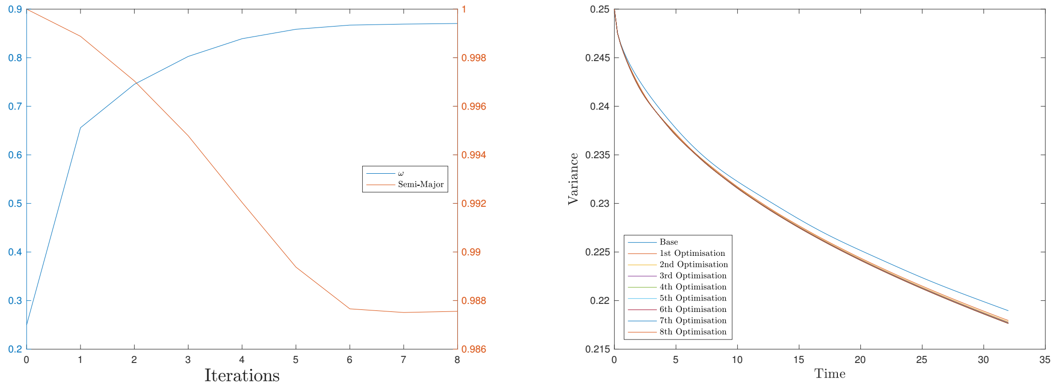


Figure 4.2: Case 1: mixing optimisation using one stationary rotating stirrer. A highly penalised optimisation setting ($\lambda = 10^{-3}$ from equation (4.1)) has been used. (a) Rotational speed ω and axis a versus the number of direct-adjoint iterations. (b) Variance of the passive scalar versus time $t \in [0,\ T]$.

thus solely rely on diffusion for our mixing. However, we note that the seventh and eighth iteration brings about a marked change and leads to a significant decrease in the variance when compared to the preceding steps (see figure 4.3b for a closer view). The associated stronger mixing is created by the availability of an advective process caused by the elliptic stirrer that is now sufficiently elongated (and spinning sufficiently fast) to shed vortices off its tips. These vortices form the sought-after small-scale structures that intensify the diffusion process by increasing the length of the interface between the two fluids. These increased small scale dynamics enhance the adjoint's ability to create further mixing, and therefore, result in the drastic decrease in variance, as can be observed in figure 4.3a.

During the optimisation, we notice a substantial increase in the adjoint variables – to a degree that requires the marked reduction of the step-size in the optimisation routine. This increase is expected, as a more efficient process that accomplished mixing is available after the stirrer has been modified to induce vortex shedding. In other words, after six iterations on the cost functional surface, we have reached the edge of the diffusion-dominated plateau and progressed towards lower variances by exploiting advection-diffusion-dominated mixing. The high penalisation parameter λ in the previous section inhibited the exploration of this regime.

The shape of the variance for the ninth iteration (see figure 4.3a) exhibits a levelling off during the later stages of the direct simulation (between $t \approx 25$ and $t = 32$). This suggests that at this point the mixing process driven by the elliptical stirrer is nearly complete, and any further mixing is primarily due to diffusion. In fact, when comparing the gradient of the variance to previous iterations, we observe a corresponding similarity and thus can conclude that advection no longer plays an important role. This conclusion is further corroborated by regarding the right-hand column of figure 4.4, which shows snapshots from the final iteration; the uniformity of the passive scalar field θ is evident towards the end of the temporal optimisation horizon. The optimisation is stopped here, as the magnitude of the rotational speed of the next update step lead to numerical instabilities and the time-step would have had to be reduced significantly at great computational cost to achieve any meaningful result.

While this simple example has supplied information about the convergence behaviour and the role of the penalisation parameter in including or excluding physical mixing strategies to accomplish optimal mixing results, we now proceed to more complex cases and further probe the direct-adjoint optimisation framework.

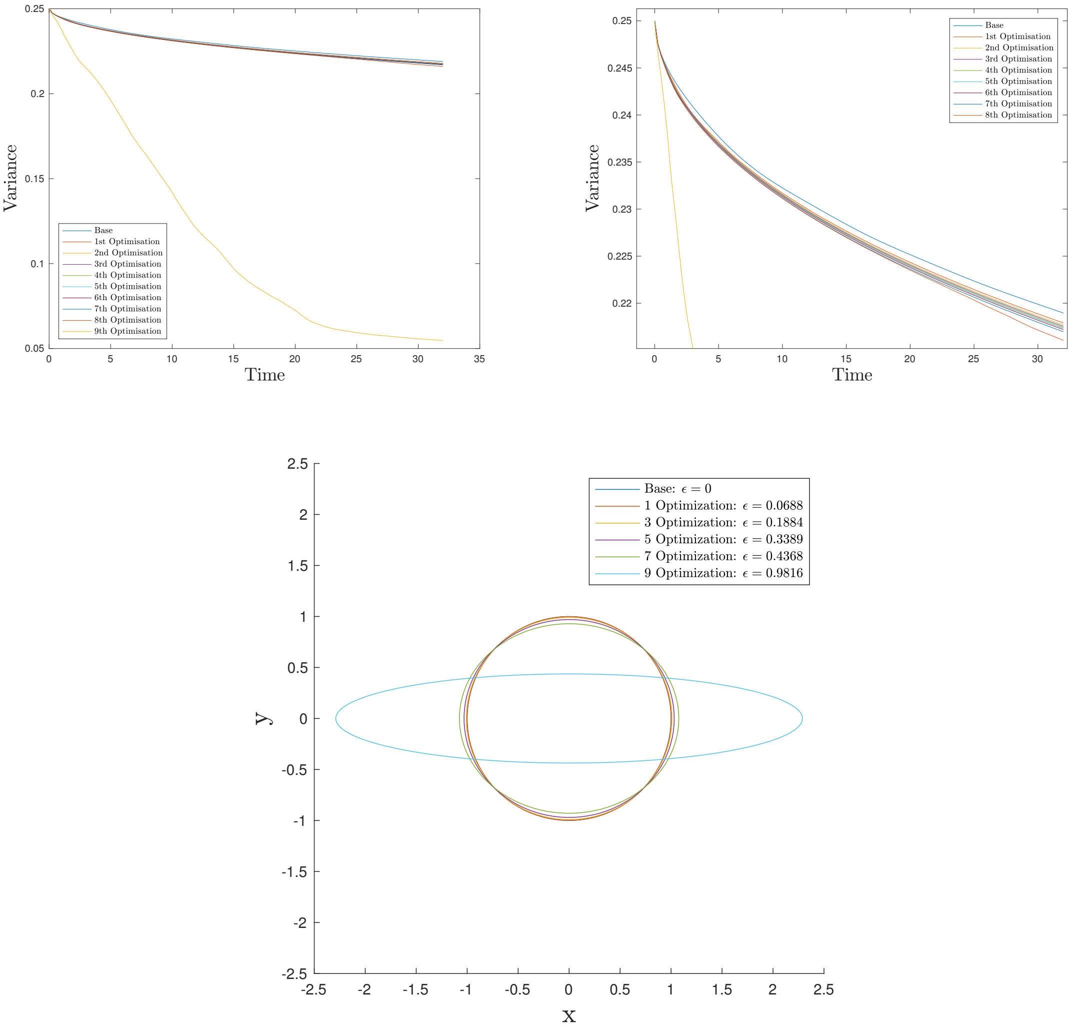


Figure 4.3: Case 1: mixing optimisation using one stationary rotating stirrer. (a) Variance, as defined in equation (2.81), of the scalar field θ versus time $t \in [0,\ T]$. (b) Zoomed-in view to illustrate the decrease in variance for the first eight iterations. (c) Contour of cylinder shapes as result of successive iterations.

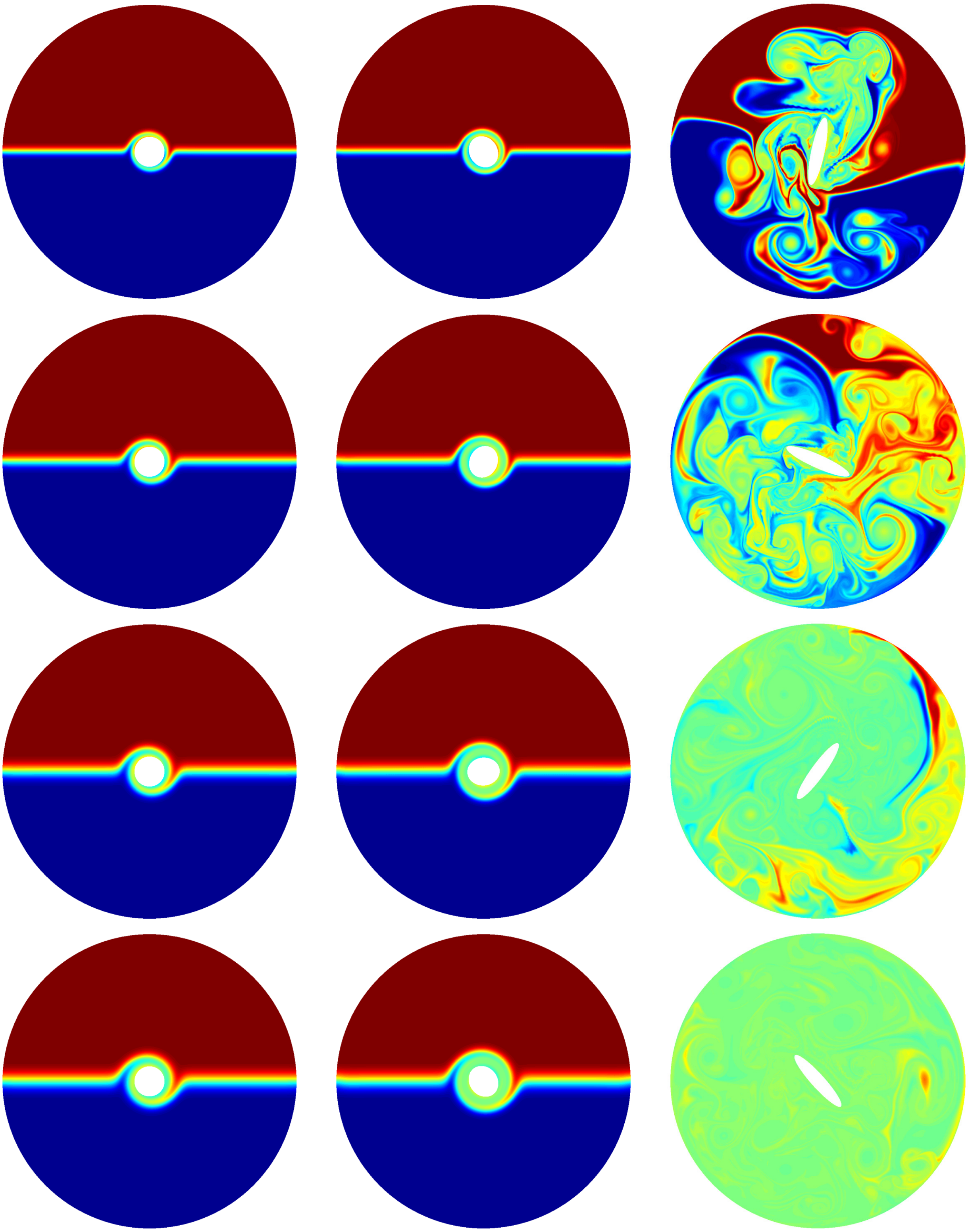


Figure 4.4: Case 1: mixing optimisation using one stationary rotating stirrer. Left column: unoptimised configuration, with snapshots at $t = 8, 16, 24, 32$ (top to bottom). Middle Column: after four direct-adjoint optimisations, with snapshots at $t = 8, 16, 24, 32$ (top to bottom). Right column: after nine direct-adjoint optimisations, with snapshots at $t = 8, 16, 24, 32$ (top to bottom). For videos of these scenarios please refer to `1Before.mp4`, `1Intermediate.mp4` and `1After.mp4` for the left, middle and right column, respectively.

4.2.4 Case 2: two stationary rotating stirrers

We complicate the geometry and optimisation scheme by introducing a second stirrer which we place along the horizontal axis of the mixing vessel; as before, the initial shape of both stirrers is taken as circular. Again, we also consider the influence of the penalisation parameter (controlling the maximum amount of kinetic energy added to the system, i.e., limiting $\int_0^{TF} \sum_i [(\boldsymbol{u}_{s,i})_j \chi_i]^H \mathsf{R}_i [(\boldsymbol{u}_{s,i})_j \chi_i] \, \mathrm{d}t$) on the convergence behaviour and the chosen physical optimisation strategy.

Highly penalised system

We observe, similar to the previous one-cylinder case, that a high energy-penalisation parameter prevents the optimal mixing strategy to explore options other than solid-body rotation and diffusion of the spinning boundary layer around the (mostly) circular stirrers. With the existence of a second stirrer, there is an additional possibility for the optimisation scheme of exiting this regime: by placing the two stirrers close to each other, we can have their respective rotating boundary layers interact and exchange sufficient adjoint (gradient) information to induce advection-dominated strategies and a corresponding drop in variance. In our case, the two stirrers appear unaware of each other; no cooperative strategy is pursued by the optimisation scheme and, as a result, little progress is made in improving mixing efficiency.

Figure 4.5a shows the evolution of the axis length (while maintaining the cross-sectional area) and the rotational speed of the first (left) stirrer as we progress through six iterations of the direct-adjoint looping. We see an increase in the rotational speed, but only an insignificant change in the stirrer's eccentricity. The control variables for the second (right) stirrer are identical to the ones shown in figure 4.5a. It is not surprising that the temporal evolution of the variance over these six iterations appears rather clustered (see figure 4.5b).

Weakly penalised system

Applying a lower value of λ to this two-stirrer configuration is expected to yield similar results as observed before: by allowing a less restrictive value for the kinetic energy of the stirrers, advective processes will become a feasible option, vortex shedding from elliptical stirrers will commence and substantially more efficient mixing will ensue.

While this behaviour is certainly prevalent (as shown in figure 4.6), when consider-

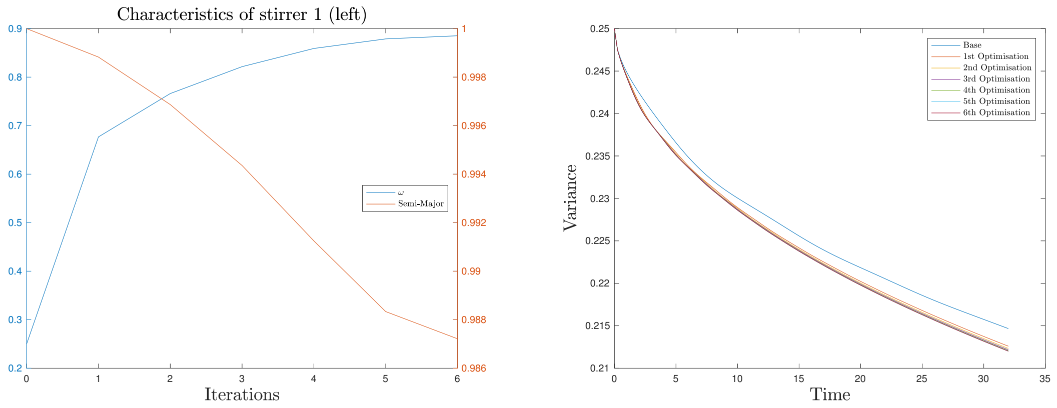


Figure 4.5: Case 2: mixing optimisation using two stationary rotating stirrers. A highly penalised optimisation setting ($\lambda = 10^{-3}$ from equation (4.1)) has been used. (a) Rotational speed ω and axis length a versus the number of direct-adjoint iterations for the first (left) cylinder. (b) Variance of the passive scalar versus time $t \in [0,\ T]$.

ing snapshots in time after the seventh iteration of the direct-adjoint optimisation we observe that, while the right stirrer has been optimised into a fast-rotating, elliptical shape (as anticipated), the left cylinder is still nearly circular in nature and appears to only mix by diffusing its rotating boundary layer (see the right column of figure 4.7). In fact, the bulk of the variance drop can be ascribed to the right stirrer.

This observation highlights an issue and shortcoming of gradient-based optimisation. In order for the left stirrer to 'engage' in the mixing process, it has to pass through a (locally) less optimal configuration. In other words, we have to first allow an increase in variance contribution from the left stirrer, before we can substantially lower the global variance by having the left stirrer contribute to the overall mixing process. Within our optimisation framework, where after every iteration we proceed along the local gradient, this required procedure is excluded. However, this is a well-known and acknowledged issue of gradient-based optimisation schemes: we are able to find local minima, but have no guarantee (or strategy) to find a global optimum.

In our case, we can improve the situation by mirroring the control variables of the second (right) stirrer onto the first (left) stirrer. In this manner, we induce sufficient gradients in the cost functional for *both* cylinders to encourage further progress in the reduction of the global variance. Hence, we achieve lower variance levels than before (see the dashed line in figure 4.6). The final shape of the stirrers for the improved strategy consists of two ellipses of marked eccentricity, with their respective semi-major axes positioned 90° to each other (the behaviour of this

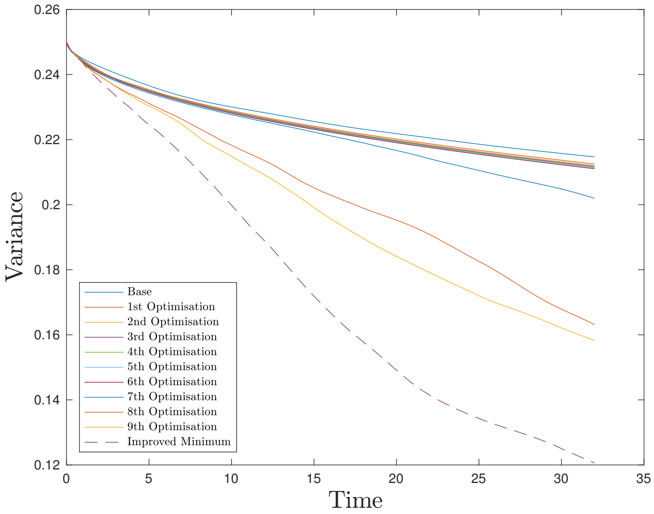


Figure 4.6: Case 2: mixing optimisation using two stationary rotating stirrers. Variance, as defined in equation (2.81), of the scalar field θ versus time $t \in [0,\ T]$. The solid lines represent a local optimum, where the left stirrer remains rather inactive. The dashed line represents an improved optimum, by mirroring the control variables onto the left stirrer before continuing the gradient-based optimisation.

improved minimum can be seen in the video `2ImprovedMin.mp4`).

For a more objective manner of reaching a global optimum, sophisticated optimisation strategies have to be employed in addition to the gradient-based framework; these strategies require the (commonly stochastic) evaluation of various parameter settings and quickly become prohibitively expensive for large-scale, PDE-based optimisations.

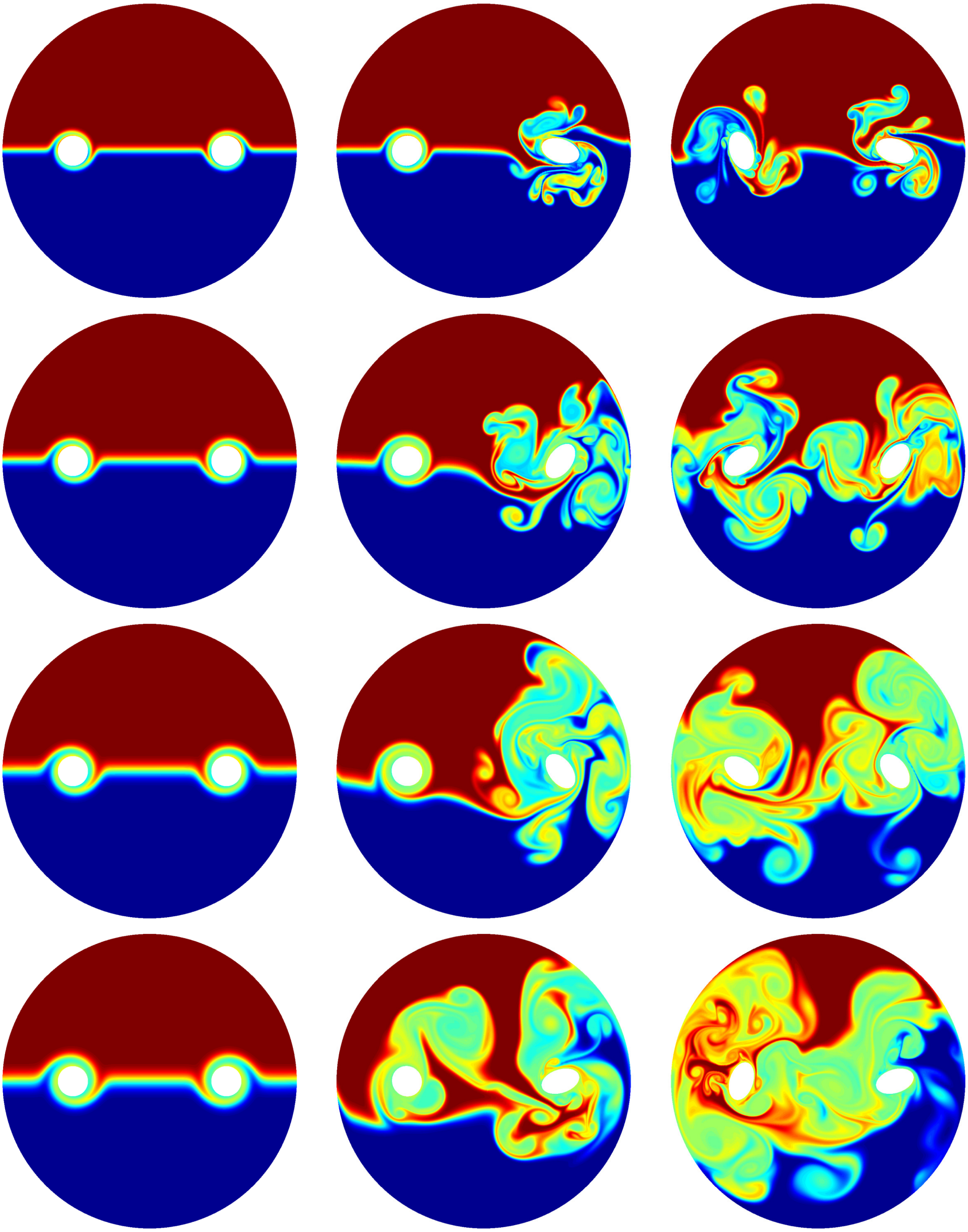


Figure 4.7: Case 2: mixing optimisation using two stationary rotating stirrers. Left column: unoptimised configuration, with snapshots at $t = 8, 16, 24, 32$ (top to bottom). Middle column: after seven direct-adjoint optimisations, with snapshots at $t = 8, 16, 24, 32$ (top to bottom). Right column: Enforced minimum by mirroring the axis length across both cylinders at $t = 8, 16, 24, 32$. For videos of these scenarios please refer to `2Before.mp4`, `2After.mp4` and `2ImprovedMin.mp4` for the left, middle and right column, respectively.

4.2.5 Case 3: five stationary rotating stirrers

Motivated by the previous configuration, we proceed by adding more stirrers and further explore the behaviour of the optimisation scheme when multiple optima and strategies compete for the best mixing efficiency.

Highly penalised system

To study the behaviour of the direct-adjoint optimisation scheme on this more complicated geometry, we first impose a high energy penalisation, as before. The results are displayed in figure 4.8, for the stirrers labelled 1, 3 and 5 (see 4.1). While the two stirrers on the horizontal axis show behaviour similar to the previous cases (i.e., a tendency towards higher rotational speeds and elliptic shapes), eccentricity remains largely unchanged, while the rotational velocity is significantly dampened in the off-set stirrer. At first sight, this may run counter to intuition that suggests that higher speeds result in improved mixing. However, it appears that – under the constraints of a limited energy budget – it is more advantageous to invest energy into the aligned stirrers rather than squandering it on the offset stirrers that are located in a rather homogeneous tracer field and thus cannot contribute to the global variance drop to any significant degree. For this reason, the optimisation scheme (more specifically, the adjoint system) directs focus on the three aligned stirrers that do make a difference.

Regarding the variances of the scalar field in figure 4.9 we conclude, as before, that we do not fully utilise diffusion *and* advection, which is required for efficient mixing. To exhibit a significant decrease in the variance, we have to explore the weak energy penalisation regime.

Weakly penalised system

Whenmoving from the high penalised to the low penalised scenario, while still limiting the total amount of kinetic energy distributed among the five stirrers, we see the emergence of advection-based mixing. The five stirrers take on elliptical shapes of varying eccentricity and tend to increase in their spin rate. This process induces a complex system of shed vortices that not only introduces small-scale features and thin filaments, but also transports information between the five stirrers. The ensuing optimisation scheme then finds an optimal collaborative mixing strategy between the five stirrers that optimises the global variance while remaining within the imposed L^2 constraint of $\boldsymbol{u}_s$.

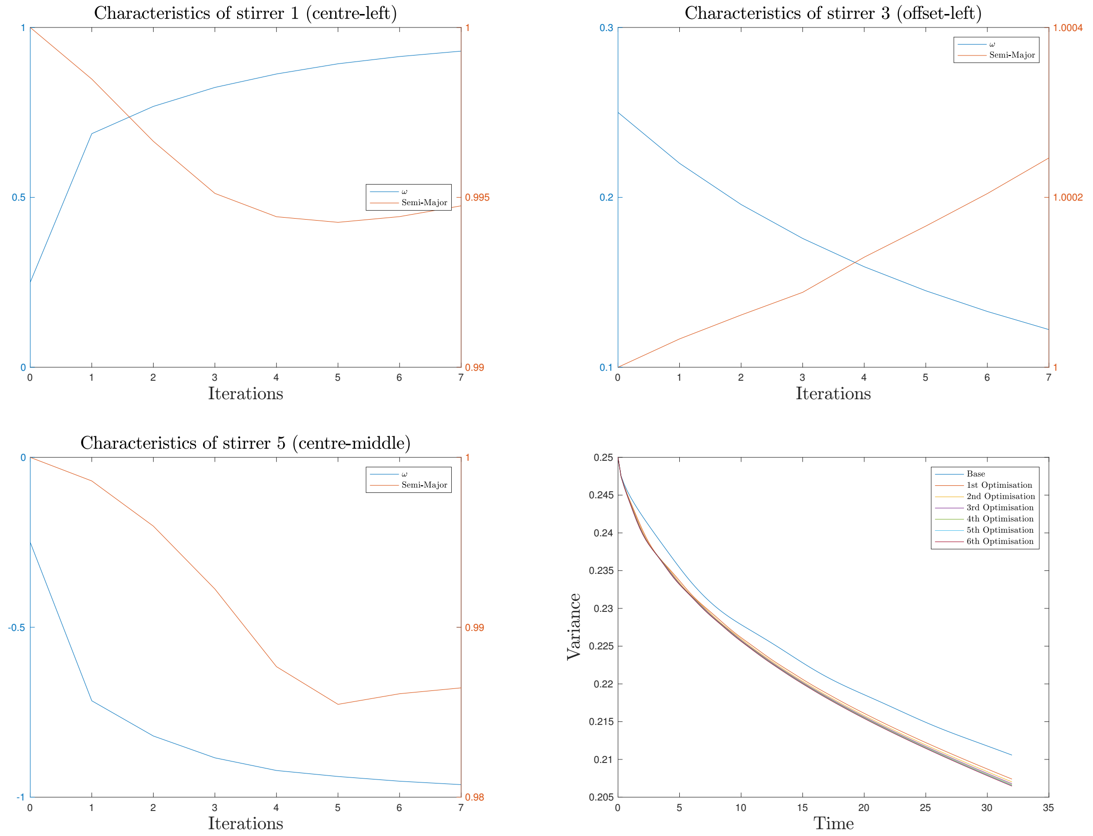


Figure 4.8: Case 3: mixing optimisation using five stationary rotating stirrers. A highly penalised optimisation setting ($\lambda = 10^{-3}$ from equation (4.1)) has been used. (a) Rotational speed ω and axis a versus the number of direct-adjoint iterations for the first (left-most) cylinder. (b) Rotational speed ω and axis a versus the number of direct-adjoint iterations for the third (bottom) cylinder. (c) Rotational speed ω and axis a versus the number of direct-adjoint iterations for the fifth (centre) cylinder. (d) Variance of the passive scalar versus time $t \in [0,\ T]$.

Throughout the initial iterations, no significant information is exchanged between the cylinders, and therefore the off-set stirrers are neglected due to the homogeneity of their surroundings. Even in the low penalisation the optimisation regards any energy used by these cylinders as wasteful. During subsequent iterations the transport of information is built up but remains negligible. In particular, considering the variances in figure 4.9, we note that the initial five iterations do not lead to significant improvements in mixedness, as we mainly stay within the previously mentioned solid-body rotation. Once vortex shedding sets in (starting at the sixth iteration, mainly with the central stirrer), however, we see a significantly larger decrease in variance. Continuing further in the optimisation, all remaining stirrers are involved in the mixing process, as a jet forms between the central cylinders which interacts with the off-set cylinders implying that energy expended on their rotation will have an effect on the mixing. In particular, the elliptical shape of the stirrers converges towards its final configuration relative to each other.

Similar to the one-cylinder case, our final optimisation reaches a point where the predominant mixing process is diffusion and the gradient of the variance is comparable to the purely diffusive limit.

Temporal snapshots of the optimised mixing (after seven iterations) can be observed in figure 4.10 in the right column. We remark that this is the final optimisation, since any further lengthening of the central stirrer (as suggested by the next adjoint step) would cause a collision with the other horizontally aligned stirrers.

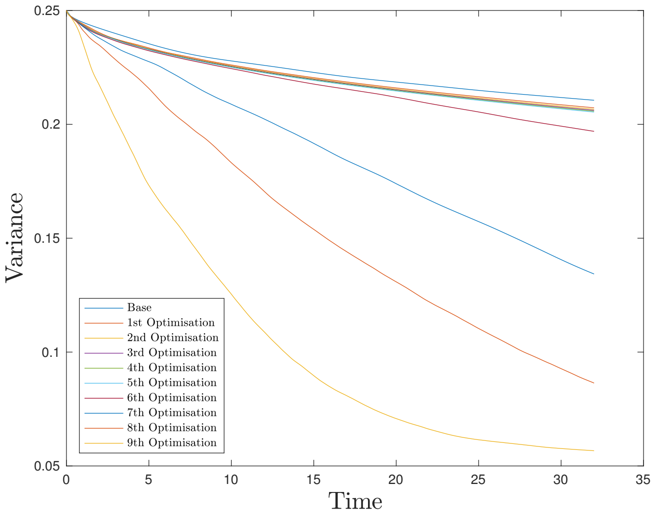


Figure 4.9: Case 3: mixing optimisation using five stationary rotating stirrers. Variance, as defined in equation (2.81), of the scalar field θ versus time $t \in [0,\ T]$.

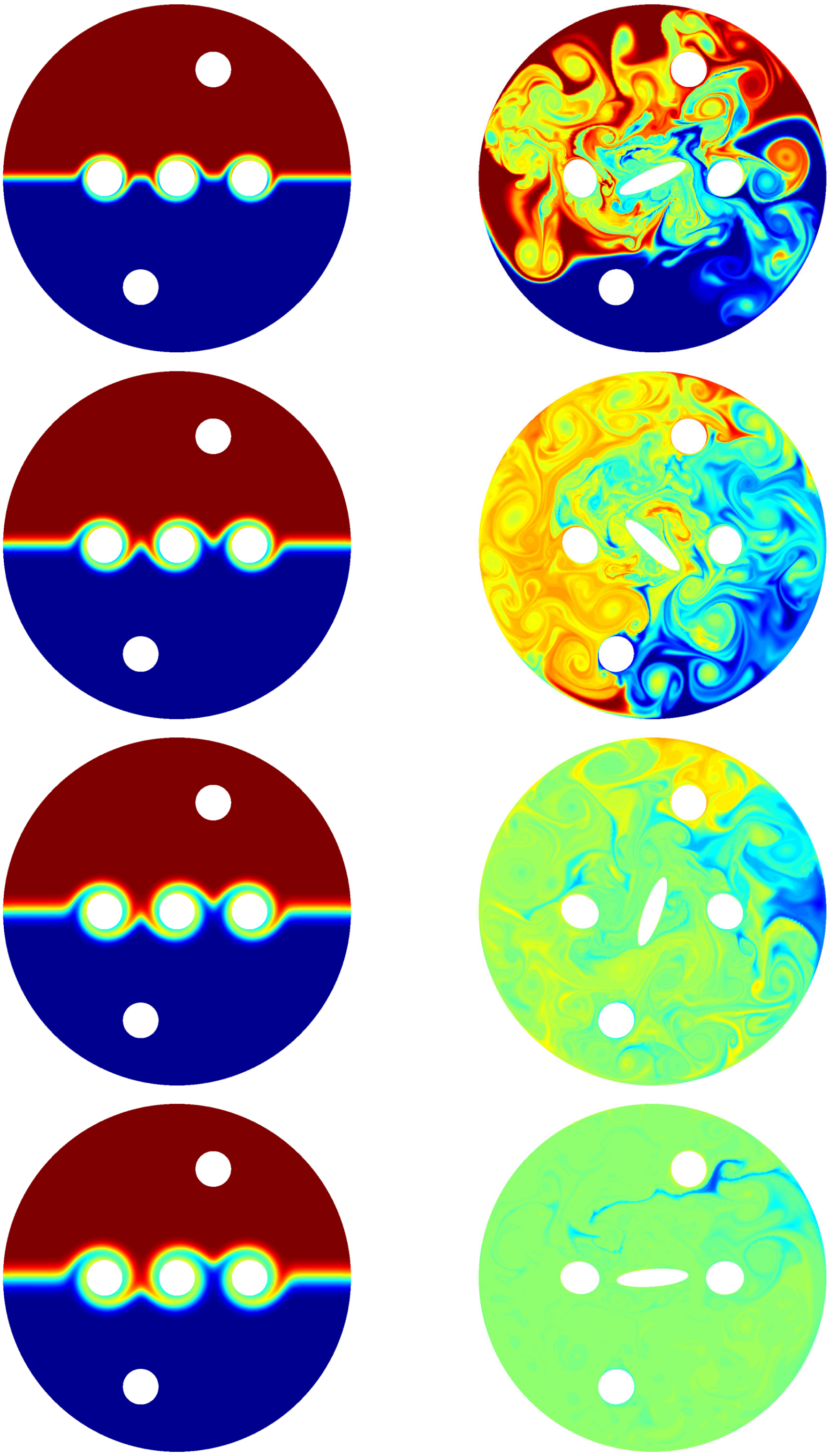


Figure 4.10: Case 3: mixing optimisation using five stationary rotating stirrers. Left column: unoptimised configuration, with snapshots at $t = 8, 16, 24, 32$ (top to bottom). Right column: after seven direct-adjoint optimisations, with snapshots at $t = 8, 16, 24, 32$ (top to bottom). For videos of these scenarios please refer to `5Before.mp4` and `5After.mp4` for the left and right column, respectively.

4.2.6 Case 4: one horizontally moving, rotating stirrer

All previous configurations relied on rotating but stationary stirrers, and thus only partially demonstrate the capabilities of the direct-adjoint method and the associated penalisation framework. In this final case, we present a scenario that considers the optimisation of the shape of a stirrer while being dragged through the binary fluid; the velocity of the stirrer is defined by a function of the form $\cos(t)$. We neglect energy penalisation (i.e., $\lambda = 0$) in this case, as little effect is expected from a pure shape optimisation, since the bulk of the energy expenditure is already contained in the back-and-forth motion of the stirrer. As mentioned previously, a case may be made for including a power constraint in this system as the change in the shape will introduce extra cost in terms of the physical operation of the motor. However, we wish here to solely study the capability of framework in optimising a moving stirrer. Therefore, we aim to employ the simplest cost-functional for this particular scenario.

We observe in figure 4.11 a pronounced decrease in the variance as the initially circular cylinder is lengthened vertically (for the position at $t = 0$). The reason for this optimal configuration is certainly linked to the fact that dragging an elliptical stirrer across the interface starting in this vertical position achieves a great deal of mixing by producing small-scale structures. An alternative, initially horizontal design would perform significantly worse. In addition to this obvious observation, we notice that starting in the vertical position at $t = 0$ allows the stirrer to take on the high-drag vertical position nine times during a full simulation cycles, while an initially horizontal ellipse would exhibit the same high-drag position only eight times per cycle. From the early stages of the cycle (e.g. $t = 8$), we create a great many filamented structures that subsequently get diffused by the flow and help in ultimately producing a homogeneous mixture. This effect continues further throughout the simulation and leads to the substantial decrease in variance, as shown in figure 4.11.

Finally, we note that the direct-adjoint system, as introduced, is ignorant of the physical restrictions by the geometry; specifically, geometrically overlapping or otherwise colliding structures are not explicitly accounted for. For this reason, we have to manually terminate the optimisation scheme in the case of such an event.

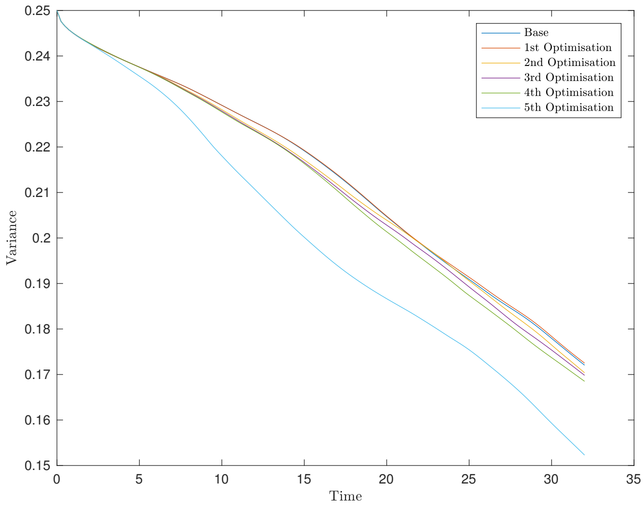


Figure 4.11: Case 4: mixing optimisation using one horizontally moving and rotating stirrer. Variance, as defined in equation (2.81), of the scalar field θ versus time $t \in [0,\ T]$.

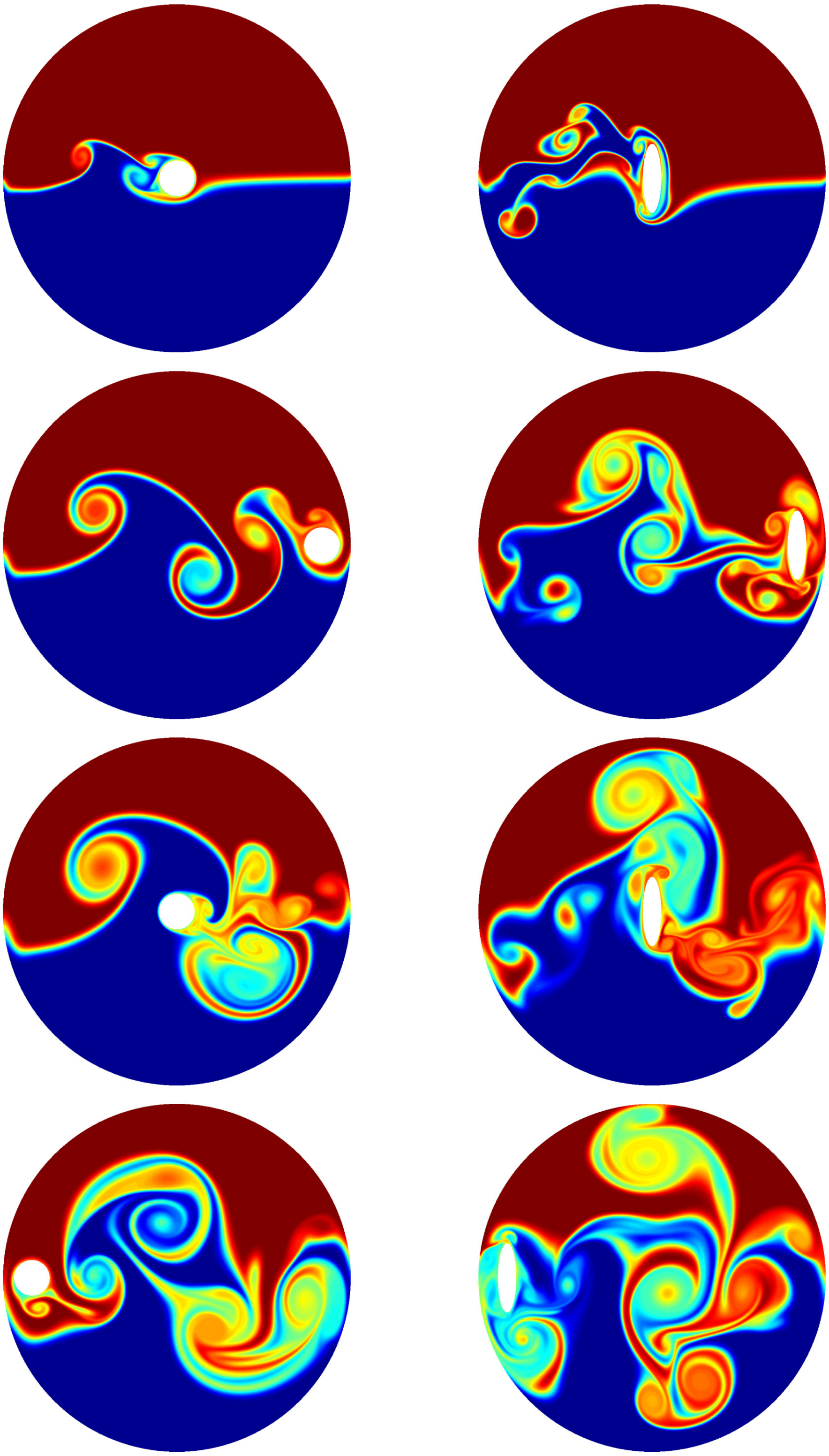


Figure 4.12: Case 4: mixing optimisation using one horizontally moving and rotating stirrer. Left column: unoptimised configuration, with snapshots at $t = 8, 16, 24, 32$ (top to bottom). Right column: after five direct-adjoint optimisations, with snapshots at $t = 8, 16, 24, 32$ (top to bottom). For videos of these scenarios please refer to `CosBefore.mp4` and `CosAfter.mp4` for the left and right column, respectively.

5

Results 2: Fourier-based shape optimisation

5.1 Optimisation specifics

5.1.1 Optimisation parameters

In the previous section both stirrer geometry and rotational velocity have been our control parameters. It has been found that the bulk of the mixing efficiency was attributed to the additional kinetic energy supplied to the system by the stirrers' increased rotation, less effect was due to a change in eccentricity. To isolate the effects of these two control parameters we will concentrate in this section on pure shape optimisation and study its effects on mixing efficiency.

5.1.2 Shape parameterisation

While the previous parameterisation of the stirrers and their morphing from circles to ellipses is more than sufficient for the validation of the computational framework, it lacks the high dimensionality of the control space to accommodate more complicated stirrer shapes. In particular with a view towards industrially applicable cases that cannot be approached by intuition alone, we need to introduce a geometric parameterisation with a far higher degree of flexibility while still retaining the main features of the previous optimisation. We are motivated by our numerical approach in the choice of an efficient parameterisation. To this end, we introduce a Fourier-based shape parameterisation which has been shown[71] to allow for an wide array of shapes of the following form

$$\begin{aligned} f_x(\alpha) &= \frac{a_0}{2} + \sum_{k=1}^{n} a_k \cos(k\alpha) - b_k \sin(k\alpha)\,, && \text{(5.1a)} \\ f_y(\alpha) &= \frac{c_0}{2} + \sum_{k=1}^{n} c_k \cos(k\alpha) - d_k \sin(k\alpha)\,, && \text{(5.1b)} \end{aligned}$$

where α is a vector representing the discretised interval $[0, 2\pi]$. We then generate a parametric curve in Cartesian coordinates according to

$$\begin{aligned} x &= f_x(\alpha), && \text{(5.2a)} \\ y &= f_y(\alpha), && \text{(5.2b)} \end{aligned}$$

or, in composite form,

$$f(a_i, b_i, c_i, d_i) = [f_x, f_y], \tag{5.3}$$

which forms the basis for the generation of the i^{th} stirrer shape. The potential complexity of the stirrer geometries introduces additional considerations. One of these is related to the fact that the mask requires a condition which specifies whether a point falls inside or outside our geometry. For the previous cases of circular or elliptical cross-sections this condition could be formulated in a trivial manner, i.e, via a distance function from the centre. As our new parameterisation allows for shapes of significant concavity, we have to introduce a more complex measure to determine the interior or exterior of our geometry. Additionally, further complications are added to this issue by allowing highly distorted geometric shapes whose barycentre lies outside the solid structure. Therefore, we introduce the winding number $w(\boldsymbol{x})$. This number is defined as

$$w(\boldsymbol{x}_k) = \sum_{i=1}^{n} \phi_{i,i-1}, \tag{5.4}$$

where $\phi_{i,j}$ denotes the angle of the arc, which is centred at the test point $\boldsymbol{x}_k$, and runs along the parametric curve from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$. As the points $\boldsymbol{x}_i, \boldsymbol{x}_j$ and the angle ϕ sweep the full perimeter of the parametric curve, the total angle travelling around the curve will be 2π if the point $\boldsymbol{x}_k$ is inside the curve or zero if it falls outside.

Combining this concept with our previous definition of the mask we obtain

$$\chi_i(\boldsymbol{x}, t) = \begin{cases} 1, & w_i(\boldsymbol{x}) = 2\pi, \\ 0, & w(\boldsymbol{x}) = 0, \end{cases} \tag{5.5}$$

We note that this formulation is discontinuous and thus would generate spurious numerical waves as our stirrers move across our discrete grid. Therefore, we introduce a smoothing layer to avoid the formation of this numerical noise, but emphasise that this has a negligible effect on the analytic framework. The problem is further complicated, because the the smoothing layer cannot be defined as a distance from the centre (as in equation (2.52)), instead we must rely on the minimum distance from our chosen point to our curve, i.e.,

$$d = \min_{\alpha} \sqrt{(x - f_x)^2 + (y - f_y)^2}. \tag{5.6}$$

The angle of α that assumes this minimum value will be defined as $\tilde{\alpha}$ and can be uniquely determined for each point (x, y) inside the shape. We then augment our mask function χ_i with d to obtain the smoothed mask

$$\chi_i(x, t) = \begin{cases} 1, & w = 2\pi \text{ and } d > h, \\ \frac{1}{2}\left(1 + \cos(\frac{\pi(h-d)}{h})\right), & w = 2\pi \text{ and } d < h, \\ 0, & w = 0. \end{cases} \tag{5.7}$$

We note that the smoothing function connects to the limit points of the discrete function but via the middle smoothing layer, represents a continuous and differentiable formulation.

A benefit of using the parameterisation above is its flexibility in the derivation and implementation of the adjoint equation. The last statement requires further explanation. Without loss of generality we will focus on optimising the Lagrangian $\mathcal{L}$ with respect to one representative coefficient a_k, but stress that the other Fourier coefficients follow the same procedure. Since $\mathcal{L}$ does not explicitly depend on a_k, but is linked implicitly through the geometric mask, we have to split the derivative with respect to a_k into its partial components as follows

$$\frac{\partial \mathcal{L}}{\partial a_k} = \int_0^T \frac{\partial \mathcal{L}}{\partial \chi_i} \frac{\partial \chi_i}{\partial d} \frac{\partial d}{\partial a_k} \, \mathrm{d}t. \tag{5.8}$$

The first term is identical to (2.74) while the second is the analogue of the term reported in Eggl *et al.*[70]. The last term is novel and will be discussed below. While equation (5.6) does include a minimisation, the derivative term in equation (5.8) is independent of this minimum and does not affect the further derivation. Given the form of the parameterisation (equation (5.1)) , the derivative is in essence isolating the wave number within the minimum distance function, i.e.,

$$\frac{\partial d}{\partial a_k} = \left.\frac{(f_x - x)\cos(k\alpha)}{d}\right|_{\tilde{\alpha}}. \tag{5.9}$$

Combining this expression with the terms above, we arrive at the full formulation of the optimality condition with respect to the chosen a_k. The composite nature of equation (5.8) allows for an efficient implementation as the first two terms are independent of our choice of a_k and can be determined ahead of the bulk of the computations. Furthermore, the final term only changes minimally, either by including a $(f_y - y)$ term or by replacing sin with a cos term, depending on which of the coefficients we choose. This means that the calculation of the optimality condition is efficient and readily scalable as the majority of the required terms can be calculated *a priori*.

One key issue that needs attention is area conservation. The optimisation routine at this point is not constrained by area and therefore can and will modify the area of the stirrers. This may encourage shapes that achieve better mixing simply by changing the size of the stirrers. To counter this tendency, we need to define and conserve the area of our parametric curves. Previously, when we considered elliptical stirrers, we conserved their area

$$A = \pi ab, \tag{5.10}$$

where a and b are the semi-major and semi-minor axis of the ellipse, by enforcing a constant value of ab. In the Fourier-based parameterisation we have to take a similar approach, albeit with more algebraic overhead. We begin as follows

$$
\begin{aligned}
A &= -\int_{\alpha=0}^{\alpha=2\pi} f_y \mathrm{d} f_x && (5.11)\\
&= -\int_0^{2\pi} f_y \frac{\partial f_x}{\partial \alpha} \mathrm{d}\alpha && (5.12)\\
&= -\int_0^{2\pi} \left[\frac{c_0}{2} + \sum_{k=1}^{n} c_k \cos(k\alpha) - d_k \sin(k\alpha)\right] \left[\sum_{l=1}^{n} l(-a_l \sin(l\alpha) - b_l \cos(l\alpha))\right] \mathrm{d}\alpha. && (5.13)
\end{aligned}
$$

Without loss of generality, we will absorb the factor of $\frac{1}{2}$ into c_0 to simplify the notation

$$
\begin{aligned}
&= \sum_{k=0}^{n} \sum_{l=1}^{n} l \left(\int_0^{2\pi} [c_k \cos(k\alpha) - d_k \sin(k\alpha)] [a_l \sin(l\alpha) + b_l \cos(l\alpha)] \mathrm{d}\alpha\right) && (5.14)\\
&= \sum_{k=0}^{n} \sum_{l=1}^{n} l \Bigg(\int_0^{2\pi} c_k b_l \cos(k\alpha) \cos(l\alpha) + c_k a_l \cos(k\alpha) \sin(l\alpha)\\
&\qquad - d_k a_l \sin(k\alpha) \sin(l\alpha) - d_k b_l \sin(k\alpha) \cos(l\alpha) \mathrm{d}\alpha\Bigg). && (5.15)
\end{aligned}
$$

We note that the trigonometric sin and cos are mutually orthogonal over the interval $(0, 2\pi)$, and that the only non-zero terms are given by $k = l$ of $\cos \times \cos$ and $\sin \times \sin$. This implies that the evaluation of the terms in equation (5.15) result in

$$
\begin{aligned}
\sum_{k=0}^{n} \sum_{l=1}^{n} \int_0^{2\pi} c_k b_l \cos(k\alpha) \cos(l\alpha) \mathrm{d}\alpha &= \sum_{k=0}^{n} \pi c_k b_k, && (5.16)\\
\sum_{k=0}^{n} \sum_{l=1}^{n} \int_0^{2\pi} c_k a_l \cos(k\alpha) \sin(l\alpha) \mathrm{d}\alpha &= 0, && (5.17)\\
\sum_{k=1}^{n} \sum_{l=1}^{n} \int_0^{2\pi} d_k b_l \sin(k\alpha) \cos(l\alpha) \mathrm{d}\alpha &= 0, && (5.18)\\
\sum_{k=1}^{n} \sum_{l=1}^{n} \int_0^{2\pi} d_k a_l \sin(k\alpha) \sin(l\alpha) \mathrm{d}\alpha &= \sum_{k=1}^{n} d_k a_k \pi. && (5.19)
\end{aligned}
$$

Using the above and substituting into equation (5.15) we obtain

$$A = \sum_{k=1}^{n} \pi k (c_k b_k - d_k a_k). \tag{5.20}$$

It is easily shown that we recover the area of circle and ellipse using this more general formula. Next, we will use this expression to normalise the coefficients to preserve an initial area. More specifically, for an initial area

$$\tilde{A} = \sum_{k=1}^{n} \pi k (\tilde{c}_k \tilde{b}_k - \tilde{d}_k \tilde{a}_k), \tag{5.21}$$

we multiply all Fourier coefficients for the new geometric shape with new area A by $\sqrt{\frac{\tilde{A}}{A}}$, i.e.,

$$\begin{aligned} \sum_{k=1}^{n} \pi k (\sqrt{\frac{\tilde{A}}{A}} c_k \sqrt{\frac{\tilde{A}}{A}} b_k - \sqrt{\frac{\tilde{A}}{A}} d_k \sqrt{\frac{\tilde{A}}{A}} a_k) &= \frac{\tilde{A}}{A} \sum_{k=1}^{n} \pi k (c_k b_k - d_k a_k) \qquad (5.22) \\ &= \frac{\tilde{A}}{A} A \qquad (5.23) \\ &= \tilde{A}, \qquad (5.24) \end{aligned}$$

to obtain an area-normalised new shape.

We now have in place the key ingredients to successfully optimise any number of stirrers of arbitrary Fourier-based shapes. This geometric parameterisation has been found to be particularly simple and elegant to optimise.

5.1.3 Shape constraints

The possibilities of the Fourier-based approach to generate complex shapes requires additional constraints that go beyond area conservation. The new constraints we have to impose arise from the desire to generate physically feasible shapes that are implementable in industrial processes.

The first constraint we impose is related to the twisting of the parametric curve. When generating geometric shapes with a multitude of Fourier coefficients, it is highly likely that the parametric curve self-intersects, an example of which can be seen in the left shape of figure 5.1. In essence, the overall optimisation circumvents

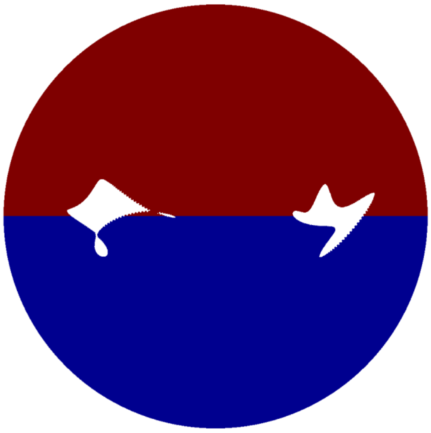


Figure 5.1: Unconstrained shape optimisation of two rotating stirrers. We can see that the left shape has generated two pinched off shapes throughout the optimisation.

the imposed choice of a prescribed number of stirrers by generating new pinched-off geometric components that then enhance mixing in later optimisation steps, albeit with the same dynamic constraints. While this option may be beneficial to the mix-norm, the geometric shape is, of course, unrealisable in a real world setting. It may seem sufficient to solely expand the distance between the points on the curve and therefore avoid such a pinching off. However, the full path along the parametric curve needs to be considered as well, as any pure thickening of the curve will inevitably leave some twisting due to our use the winding number w. What is instead needed is an untwisting strategy.

We consider two line segments $\vec{AB}$ and $\vec{CD}$ of arbitrary length and evaluate criteria for their mutual intersection. If this criterion is met, without loss of generality, we exchange B and D and mirror the parameterisation of the part of the parameter curve from A to D to generate an untwisted, and simply connected, shape. An example of such an untwisting can be seen in figure 5.2, which contains the previously shown shape that had two pinched-off areas.

After this untwisting procedure, new Fourier coefficients can be calculated and subsequently supplied to the area calculation and further optimisations.

The above untwisting would be sufficient for a continuous setting. However, on a finite grid we can arrive at pinched-off shapes even after untwisting has been performed. This is due to small geometric features that fall below the resolution of the grid despite the fact that the shape is simply connected. To avoid this complication, we introduce a displacement strategy for any points that fall below a chosen distance, so that their modified distances remain above minimum grid distance. This displacement is based on the following steps. Consider two points

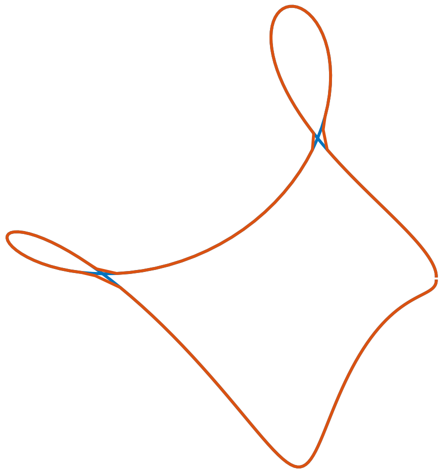


Figure 5.2: Untwisting algorithm applied to the left shape from figure 5.1. The new curve is seen in red.

which are a distance r apart. If this distance falls below the minimum distance constraint, i.e., $r < r_{min}$, we displace the two points along their respective outward pointing normals to the parametric curve. The amount of displacement, ζ, is given by a Buckingham potential[72]

$$\zeta = r_{min} \left[e^{-\frac{r}{r_{min}}} \right] . \tag{5.25}$$

A demonstration of this strategy is displayed in figure 5.3.

This strategy, in combination with the untwisting procedure, ensures a simply connected shape that is well represented on our underlying grid.

5.1.4 Introduction of the cost functional

As the entire optimisation is purely shape-based and no increase in rotational velocity is applied to the system, there is no need for the L^2 penalisation of $\boldsymbol{u}_s$ of the previous section, and thus our cost functional takes the form defined by equation (2.83). The constraints we do impose are not included in the cost functional, but are rather applied once the update from the optimisation is completed. An added change to the cost functional is concerned with the measure of mixedness: we choose the mix-norm with an exponent of $-2/3$. This particular choice of exponent was made for the following reason; previous studies using mix-norm optimisation employed either $-1/2$[60] or -1[47]. $-2/3$ represents a value that falls between the two values used in literature and thus incorporates characteristics of

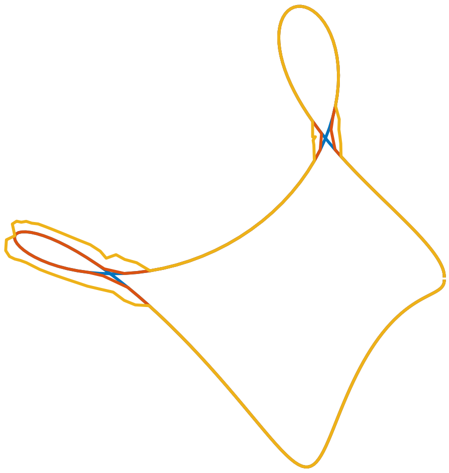


Figure 5.3: Application of thickening routine to untwisted shape.

both. As no previous studies of mix-norm optimisation had been applied to our chosen configuration, and thus no precedent had been set, we made the above choice for the exponent. The semi-discretised form of the cost functional $\mathcal{J}$ takes the following form

$$\mathcal{J} = \left. \frac{[\mathsf{A}^{-2/3}\theta]^H M [\mathsf{A}^{-2/3}\theta]}{V_\Omega} \right|_T . \tag{5.26}$$

We repeat that this type of norm encourages the formation of filamented structures in the passive scalar field, which are subsequently diffused once the optimisation window has passed.

5.2 Geometric configuration

We now turn to introducing the geometry of the test cases that demonstrate the effectiveness of shape optimisation. Earlier, we observed that the initially circular condition of the stirrers severely hampered the convergence towards an optimal design as the initial solid-body rotation did not induce vortex shedding and hence, did not provide a viable gradient direction for the optimisation. As a consequence, a substantial amount of computational effort was expended over the first few iterations, simply increasing the rotational velocity of the stirrers before shedding set in and the optimisation routine could drastically alter the shape of the stirrers. Therefore, in the absence of rotational speed as a control variable, it will be advantageous to avoid the circular initial shape and instead begin the shape optimisation with a geometry that induces vortex shedding. In this manner, the adjoint information will more quickly converge towards more optimal mixing shapes. To this end, we initialise the stirrers as four-pointed hypocycloids (astroids), examples of which can be seen in figure 5.4. This choice was primarily made, as explained above, to encourage vortex shedding from the cusps, but also, because the generated vortices do not interact with each other, but are rather confined to the respective cavity between the cusps. Therefore, we do not precondition the optimisation towards collaborative mixing strategies with interacting vortices. Instead, any change of shape in the stirrers that leads to this type of collaboration, as was observed in case three of section §4, has to be found by the optimisation algorithm of its own accord.

The arrangements and rotational speed ($\omega = \frac{2\pi}{8}$) of the star-shaped stirrers are identical to those of the previous study and allow us to make qualitative comparisons. We have also kept the values of $Re = 1000$ and $Pe = 1000$ the same. To avoid overly complicated shapes with highly corrugated boundaries, we restrict our control space to the five largest wave numbers, i.e.,

$$a_i, b_i, c_i, d_i \text{ for } i = 1, 5, \tag{5.27}$$

where these quantities are defined in equation (5.1). Finally, we limit the optimisation time to one single rotation, i.e., $T = 8$. This choice focuses the optimisation effort on short-term features rather than any residual diffusive phenomena. In effect we are forcing the optimisation to achieve efficient mixing with severe limitations in time.

5.3 Presentation of optimisation results

5.3.1 Case 1: Shape optimisation of one stirrer

We begin by studying the first case, namely a single, centrally placed stirrer. The effect on the scalar field by the action of the unoptimised stirrer and the fully optimised stirrer can be seen in figure 5.6. We observe that the shape generated by the optimisation is noticeably more asymmetrical than the initial configuration, which in turn induces the creation of more vortices and their mutual interaction. In the absence of a second stirrer, a fully collaborative mixing strategy is not possible; rather the single stirrer must achieve a self-collaborative strategy among its shed vortices. One strategy is to elongate the shape, as has been found in the previous of the same configuration: the longer the shape, the higher the velocity at the trailing edge and the more vigorous of shedding of the vortices. However,

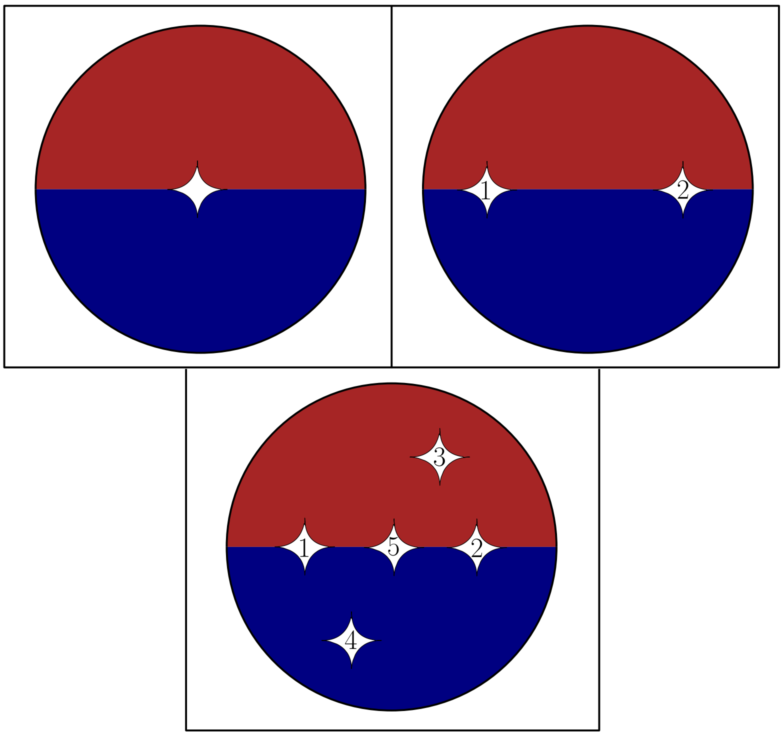


Figure 5.4: Sketch of the initial configurations for the three shape optimisation cases. Case 1 (top left): one centred stirrer rotating about its centre. Case 2 (top right): two stirrers on the horizontal axis, rotating in the clockwise directions. Case 3 (bottom centre): five rotating stirrers, placed such that three cylinders are aligned along the horizontal axis, while the remaining two are vertically offset.

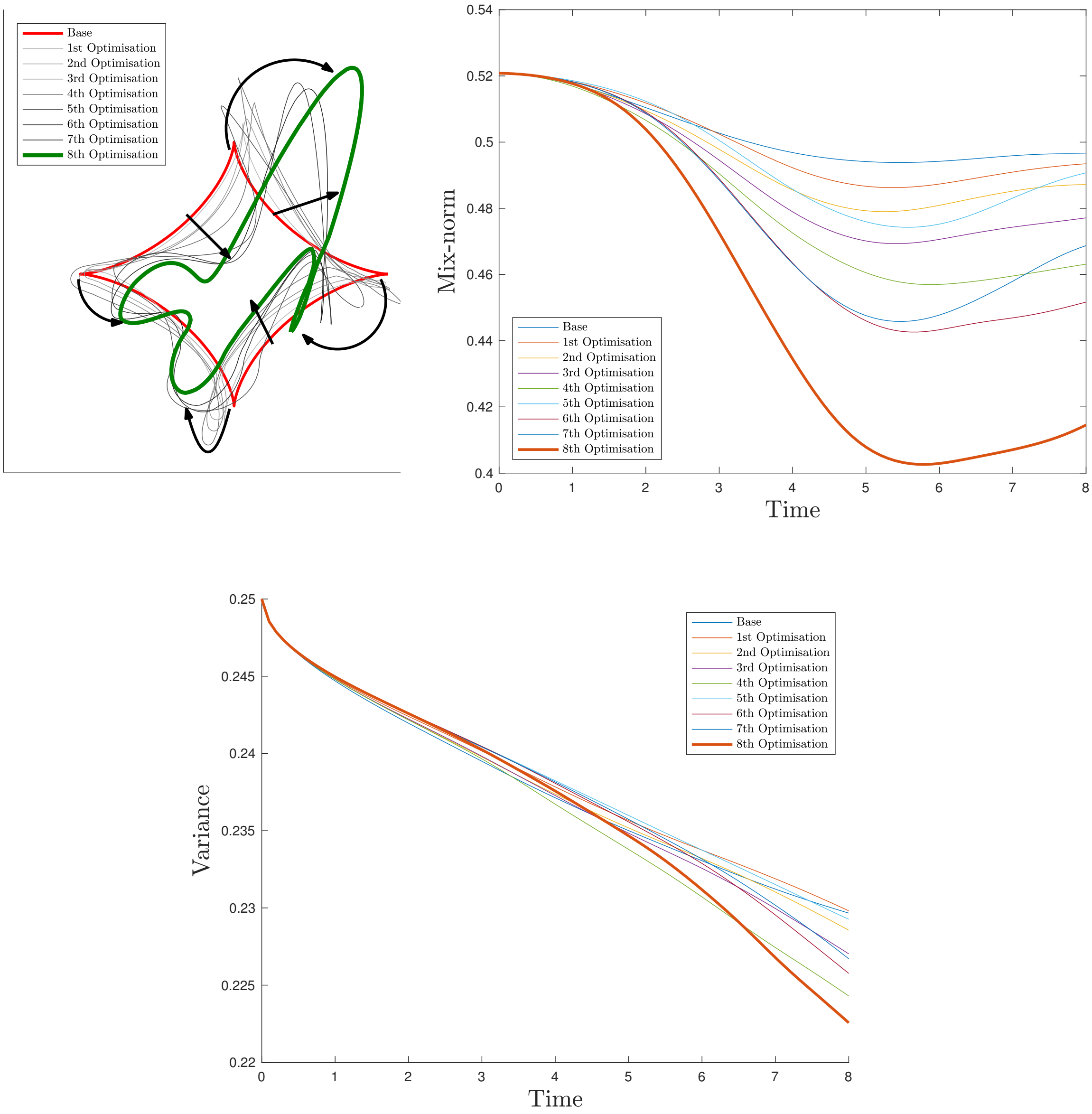


Figure 5.5: Case 1: mixing optimisation using one stationary rotating stirrer. No untwisting or thickening was applied. (a) Evolution of the shape throughout the optimisation steps. The red line is the initial configuration and green the final one. Arrows have been added in black to illustrate the changes in the shape. (b) Mix-norm of the passive scalar versus time $t \in [0,\ T]$. (c) Variance of the passive scalar versus time $t \in [0,\ T]$.

vortex shedding was not only accomplished by a simple elongation of one cusp, but also by an alignment of the three other cusps along the axis of the principal cusp. The graphical revolution of the shape evolution is shown in figure 5.5a with black arrows guiding the eye. The reader is reminded that we conserve the initial cross-sectional area of the stirrer which results not only in a lengthening, but also a thinning of the structure.

The final converged shape after our final iteration induces the formation of a trailing line vortex street which can be seen at $t = 6$ in the right column of figure 5.6. Furthermore, one of the protruding cusps near the centre of the stirrer generates an area of high circulation which draws in unmixed passive scalar and, later in the rotation, flings vorticity towards to the trailing line vortex street ($t = 6$ to $t = 8$). We urge the reader to consult the animations in the supplemental material to fully appreciate this dynamic process.

One item of particular interest in this optimisation is the capability of the direct-adjoint looping, coupled with shape optimisation, to explore competing minima beyond its current local solution. This manifests itself in figure 5.5b where the final values of the mix-norm after the fifth and seventh iteration are significantly higher, and therefore worse, than the previous iterations. Further analysis of this observation reveals that the shapes (shown in figure 5.5a) go through drastic changes in the configuration, i.e., there is a shift from an initially thought optimal configuration to a newer, more efficient, one. Throughout the initial stages of the optimisation, the first attempted modifications reinforce a particular configuration. During this reinforcement, and owing to the global nature of the Fourier-based shape representation, the optimisation scheme becomes aware of a more optimal shape at a different point in design space. In the pursuit of this new configuration the optimisation must pass through a regime of sub-optimality. Once this traversal is completed, we obtain a stirrer geometry with significantly enhanced mixing efficiency. This same convergence behaviour was not realised in case two of the previous study since the control variables for each stirrer did not communicate effectively and only an isolated control strategy could be pursued. In contrast, the Fourier-based parameterisation of the stirrer geometry furnishes a strong link between all points of the curve, and thus changes at one point are communicated effectively to the rest of the shape. In the latter case, this allows the optimisation to probe multiple local minima. However, due to the non-convexity of the problem, a convergence to a global minimum cannot be guaranteed.

After eight optimisations we terminate the direct-adjoint loop, even though better mixing efficiency could be achieved in subsequent iterations, as the resulting shapes increasingly become physically unfeasible. Up to eight optimisations, no post-processing such as thickening or untwisting was necessary and thus this optimal shape can conceivably be implemented in an industrial setting.

As a final observation, we note the rise of the mix-norm towards the end of the simulation. This may seem counter-intuitive and raise the question why the optimisation did not terminate when the mix-norm is lowest. We remind the reader that our cost-functional focuses on the end-time evaluation of the mix-norm and, therefore, any influence prior to the final time horizon does not directly enter the optimisation. The variance continues to decrease, which implies that the fluid continues to mix.

We have now demonstrated that it is possible to achieve mixing enhancement for a single stirrer, solely through the use of shape optimisation within a limited time window and without the need for increased rotational velocity. We next turn our attention to the optimisation of multiple stirrers.

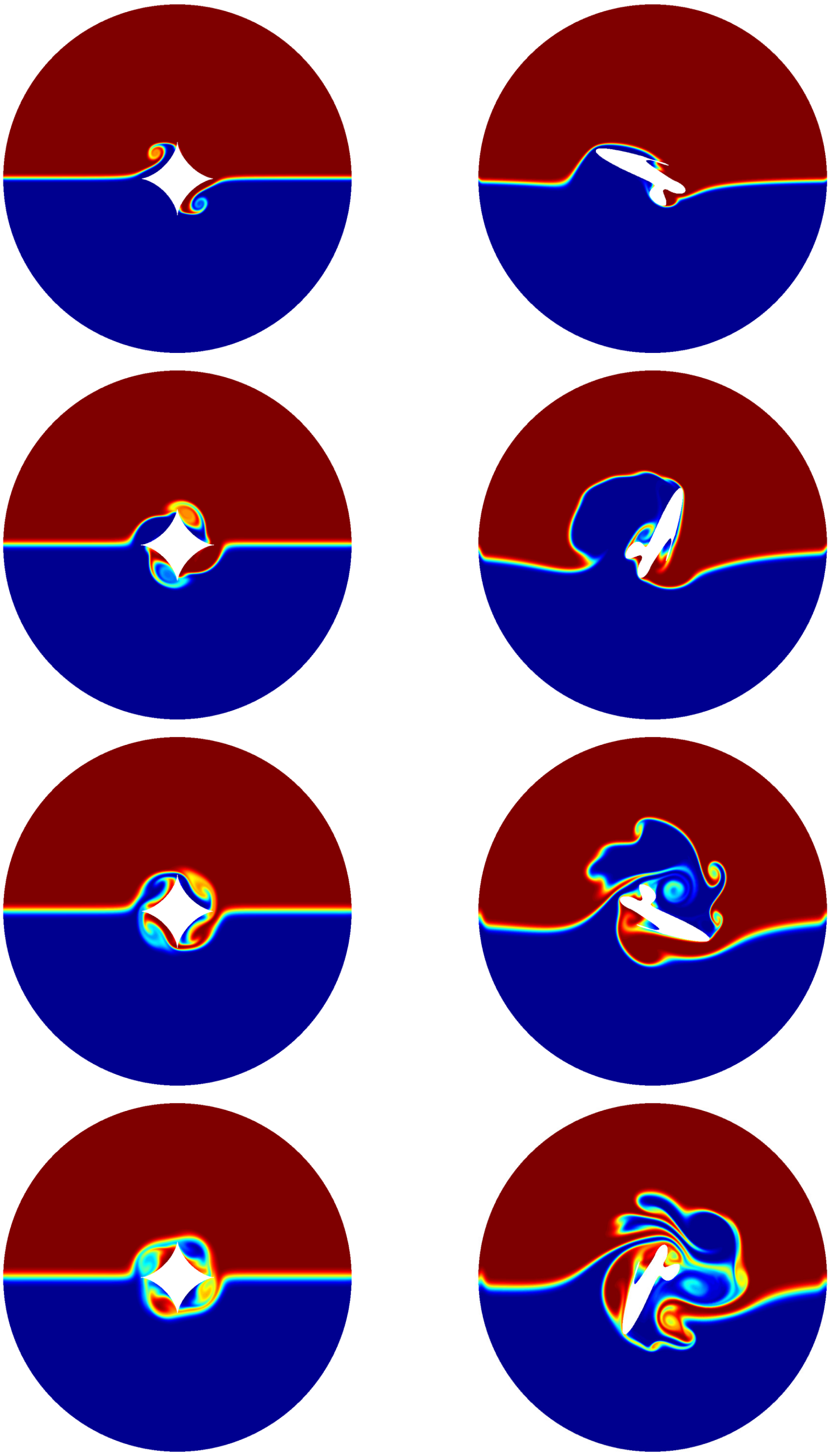


Figure 5.6: Case 1: mixing optimisation using one rotating stirrer. Left column: unoptimised configuration, with snapshots at $t = 2, 4, 6, 8$ (top to bottom). Right column: after eight direct-adjoint optimisations, with snapshots at $t = 2, 4, 6, 8$ (top to bottom). For videos of these scenarios please refer to `Shape1NoOpt.mp4` and `Shape1Opt.mp4` for the left and right column, respectively.

5.3.2 Case 2: Shape optimisation of two stirrers

We complicate the geometry and optimisation scheme by introducing a second stirrer which we also place at the interface of the fluid (see figure 5.4). Both stirrers are initially taken as a four-pointed hypocycloids and are placed sufficiently far from each other such that communication between them is weak. The interest in this configuration is two-fold: on one hand we intend to assess the proximity-effects of the outer wall, and on the other we investigate the possible rise of a collaborative mixing between the two stirrers.

As can be seen from the left column of figure 5.8, showing the unoptimised state, there is no visible interaction between the stirrers and the outer wall; the shed vortices appear unaware of any other solid in their vicinity. In contrast, the optimised results illustrate an influence of the outer wall, in particular for the left stirrer, while the right stirrer tends to concentrate its mixing efforts on the local vortex dynamics. This break in symmetry can be attributed to the identical rotational direction of the stirrers. Firstly, we observe there is an elongation in the horizontal axis for both stirrers. This echoes the behaviour of the single cylinder optimisation, where elongation has been used to generate a vortex street. However, we observe one fundamental difference: the only elongation for the single-stirrer optimisation was tilted at an angle to the interface to take advantage of a maximum plunging action, while the elongation for the left stirrer in the present case is in the horizontal direction. The principal reason for this observation is the presence of the wall, which encourages effective vortex wall collisions. The right stirrer, sensing far less influence of the wall, engages more in a plunging action and therefore develops a secondary elongation at angle to the interface similar to the one stirrer case.

In general, we observe geometric features similar to the one stirrer case, i.e., elongation in a principal direction, combined with the formation of concavities and protuberances, to generate and induce interacting vortices. While the left stirrer focuses on an exchange with the outer wall, the right stirrer follows the previous strategy of more local vortex inter-plays.

Overall we observe only a weak effort for the collaboration between the two stirrers. Minor and final changes in the right stirrer can, nevertheless, be attributed to an exchange of information occurring through the velocity field, and an example of this can be seen in form of the centre vortex at $t = 8$, which forms the beginning of a joint and collaborative mixing strategy.

We make one final remark: the constraints on the geometry (untwisting and thickening) had to be brought to bear to the left stirrer, thus ensuring a singly connected stirrer cross-section. The enforcement of these constraints compromised the optimality suggested by the adjoint system. In other words, we traded a slight loss in optimality for physical feasibility. Despite this loss, we have been successful once again (figure 5.7c).

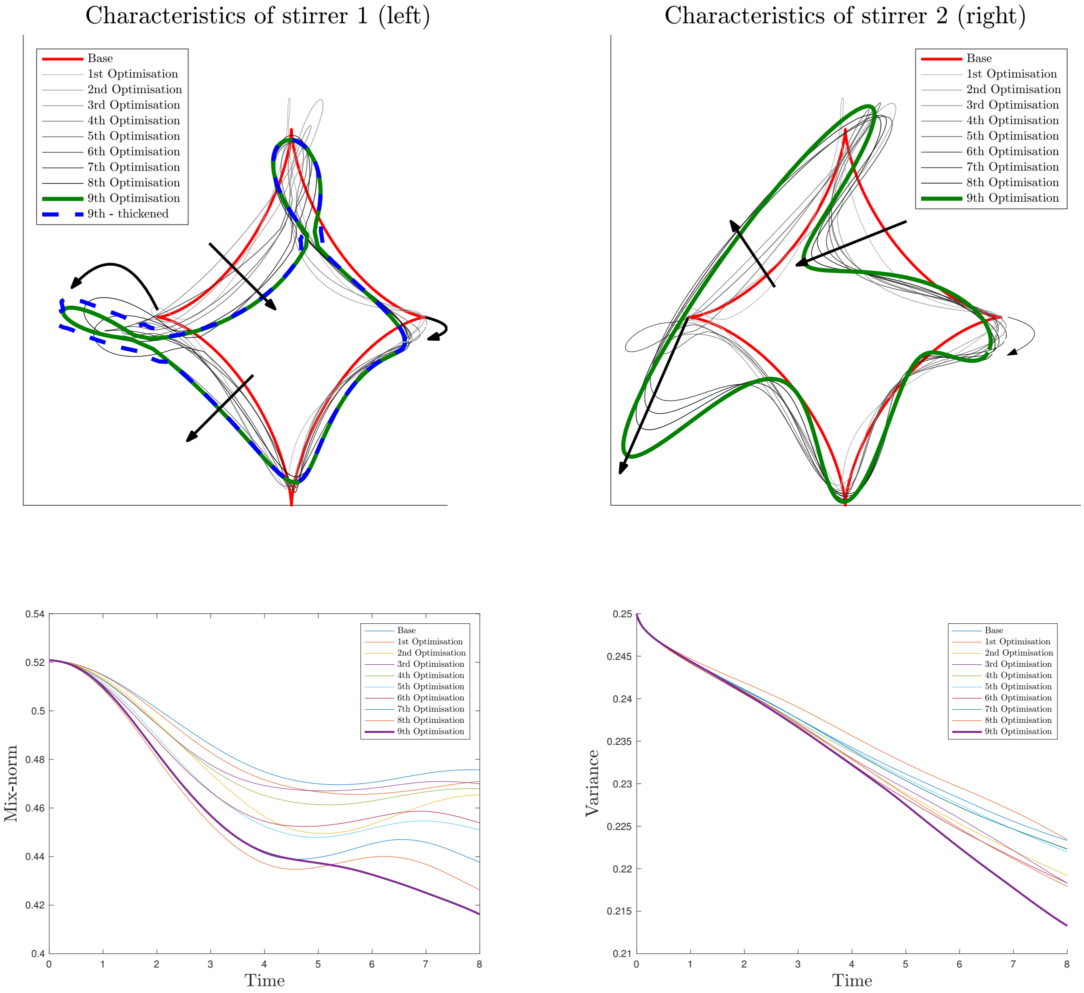


Figure 5.7: Case 2: mixing optimisation using two stationary rotating stirrers. (a) Evolution of the shape of the left stirrer throughout the optimisation steps. The red line is the initial configuration, green the final one and the blue dashed line refers to the thickening routine applied to the final optimised shape. Arrows have been added in black to illustrate the changes in the shape. (b) Evolution of the shape of the right stirrer throughout the optimisation steps. (c) Mix-norm of the passive scalar versus time $t \in [0,\ T]$. (d) Variance of the passive scalar versus time $t \in [0,\ T]$.

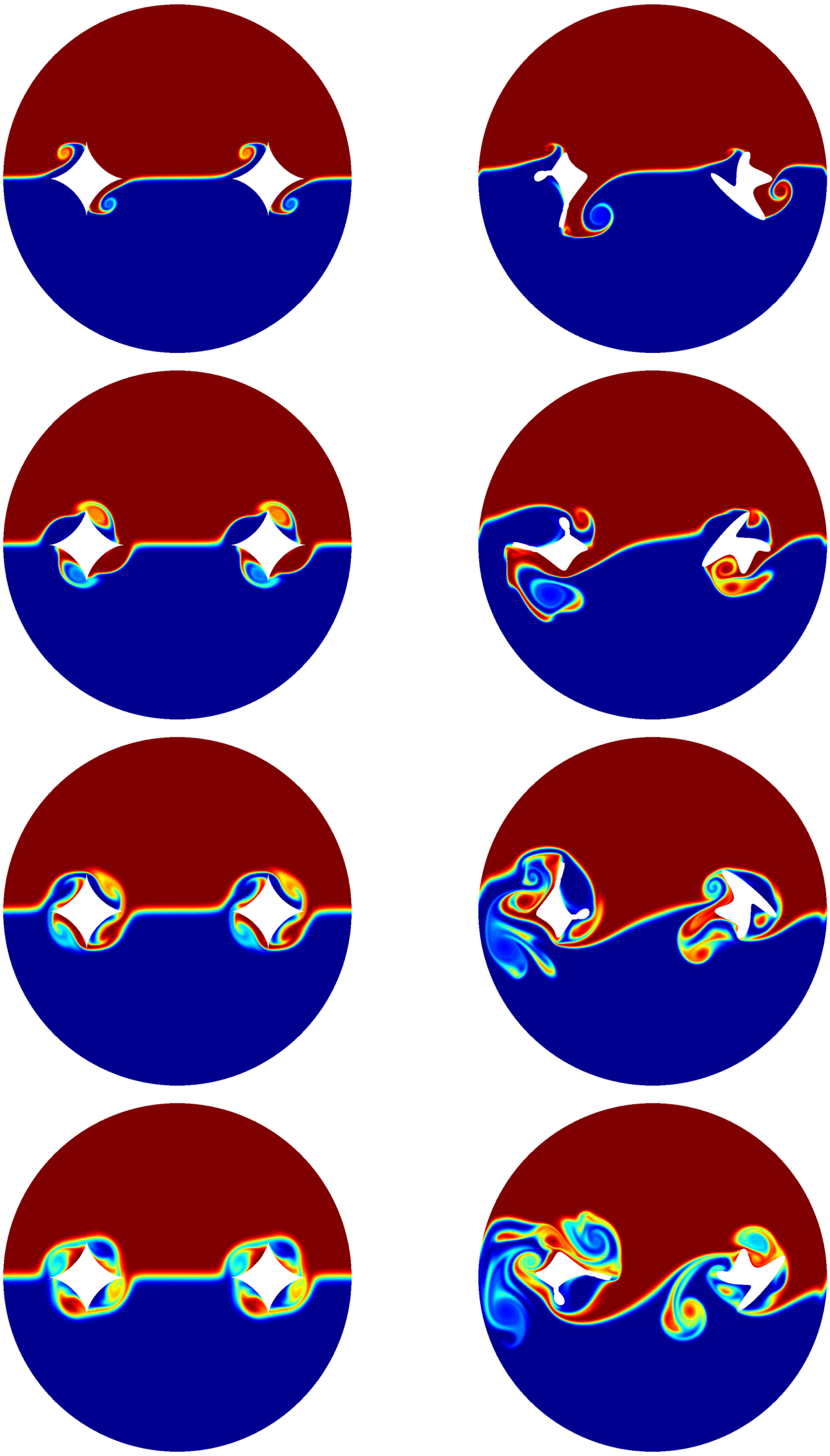


Figure 5.8: Case 2: mixing optimisation using two rotating stirrers. Left column: unoptimised configuration, with snapshots at $t = 2, 4, 6, 8$ (top to bottom). Right column: after 9 direct-adjoint optimisations, with snapshots at $t = 2, 4, 6, 8$ (top to bottom). For videos of these scenarios please refer to `Shape2NoOpt.mp4` and `Shape2Opt.mp4` for the left and right column, respectively.

5.3.3 Case 3: Shape optimisation of five stirrers

Encouraged by the slight rise of a collaborative mixing strategy for multiple stirrers, we further facilitate this process by embedding additional stirrers into the fluid. Three stirrers will be placed in the interface, with two additional stirrers at an off-set to the interface, as can be seen in figure 5.4. The closeness of the three stirrers is expected to result in a coordinated strategy between the three central stirrers, and possibly, interaction with the off-set cylinders. We recall, however, that the previously studied five cylinder configuration identified the off-set cylinders as non-critical to the overall mixing process. With no imposed energy constraints, the same optimisation strategy may not carry over to this case.

Considering the unoptimised case (see the left column of figure 5.11), we note that, despite the close proximity of the central stirrers, little to no interaction is observed between the generated vortices. After only four iterations (right column), a completely different picture arises. Stirrers one and two have been modified with changes in angles of the cusps and the introduction of areas of high concavity (see figures 5.10a,b; the black arrows illustrate the evolution of the stirrer shapes). These modifications, when combined with the changes that middle stirrer (five) has undergone, produce an interlocking gear structure. Within the narrow gaps between the central stirrers, a great many filaments in the passive scalar are produced and thoroughly mixed. Less emphasis is placed on the interaction with the outer wall, and more focus is directed towards the collaboration between the stirrers resulting in a highly distorted interface and the generation of small scale structures. Vortices generated by one stirrer collide with a neighbouring stirrer. The middle stirrer acts as a mediator between the two outer and more pointed stirrers, enabling a full and effective communication between the three central cylinders. We note that the interlocking geometry resulting in the gear-like motion of the rotating stirrers was produced by the adjoint optimisation without explicitly enforcing it.

As before, the outer off-set cylinders undergo less change, but nevertheless contribute in a minor fashion to the overall mixing process. In contrast to the previous case, where the energy considerations (L^2-norm of $\boldsymbol{u}_s$) played an important role to shut down the outer cylinders, in the present case (with no energy constraints) they eventually participate in decreasing the mix-norm. Specifically the upper off-set stirrer has been altered, so that two of its cusps have been extended to reach the mixed fluid. This can be observed at $t = 8$ in figure 5.11, where minute filaments of fluid are transported into the upper half of the disk.

One of the more remarkable aspects of this result consists of the fact that it was achieved in only four iterations. The proximity of the stirrers, and the information

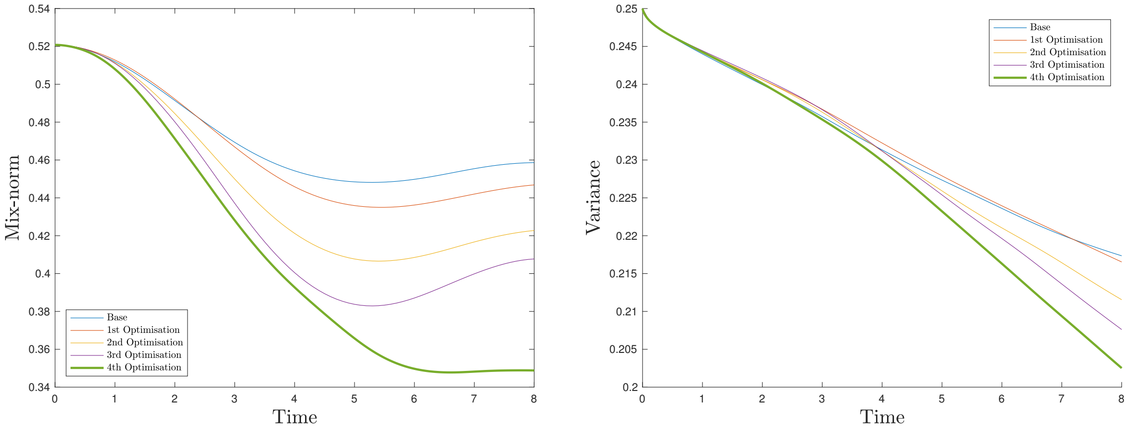


Figure 5.9: Case 3: mixing optimisation using five stationary rotating stirrers. (a) Mix-norm of the passive scalar versus time $t \in [0,\ T]$. (b) Variance of the passive scalar versus time $t \in [0,\ T]$.

they communicate to each other, is a powerful incentive for the optimisation algorithm and ensures rapid convergence to a collaborative strategy. We emphasise once again that the initial rotational velocity has not been increased, and solely shape optimisation has been performed. In view of these restrictions, and despite a rather limited time horizon, a significant decrease in the mix-norm has been achieved.

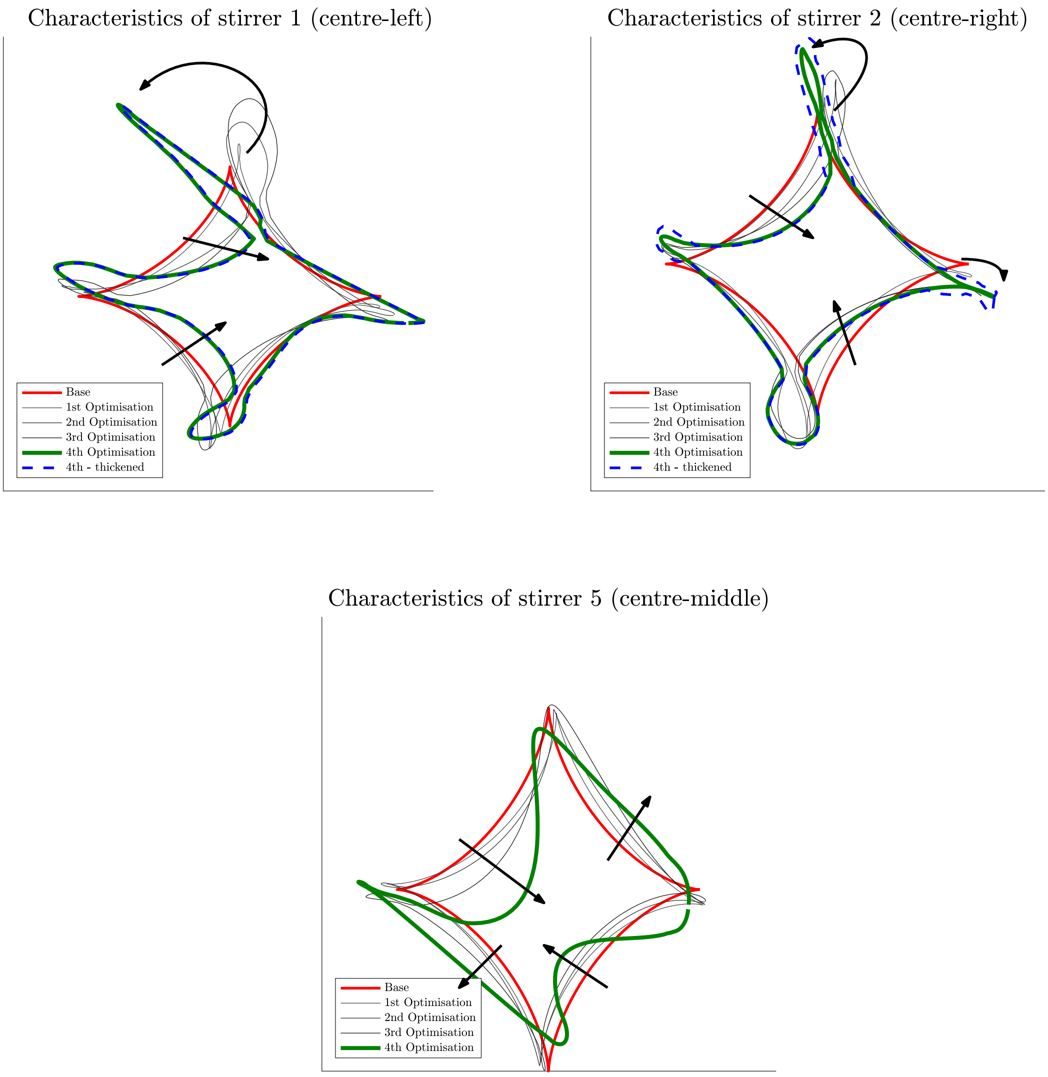


Figure 5.10: Case 3: mixing optimisation using five stationary rotating stirrers. (a) Evolution of the shape of stirrer 1 throughout the optimisation steps. The red line is the initial configuration, green the final one and the blue dashed line refers to the thickening routine applied to the final optimised shape. Arrows have been added in black to illustrate the changes in the shape. (b) Evolution of the shape of stirrer 2 throughout the optimisation steps. (c) Evolution of the shape of stirrer 5 throughout the optimisation steps.

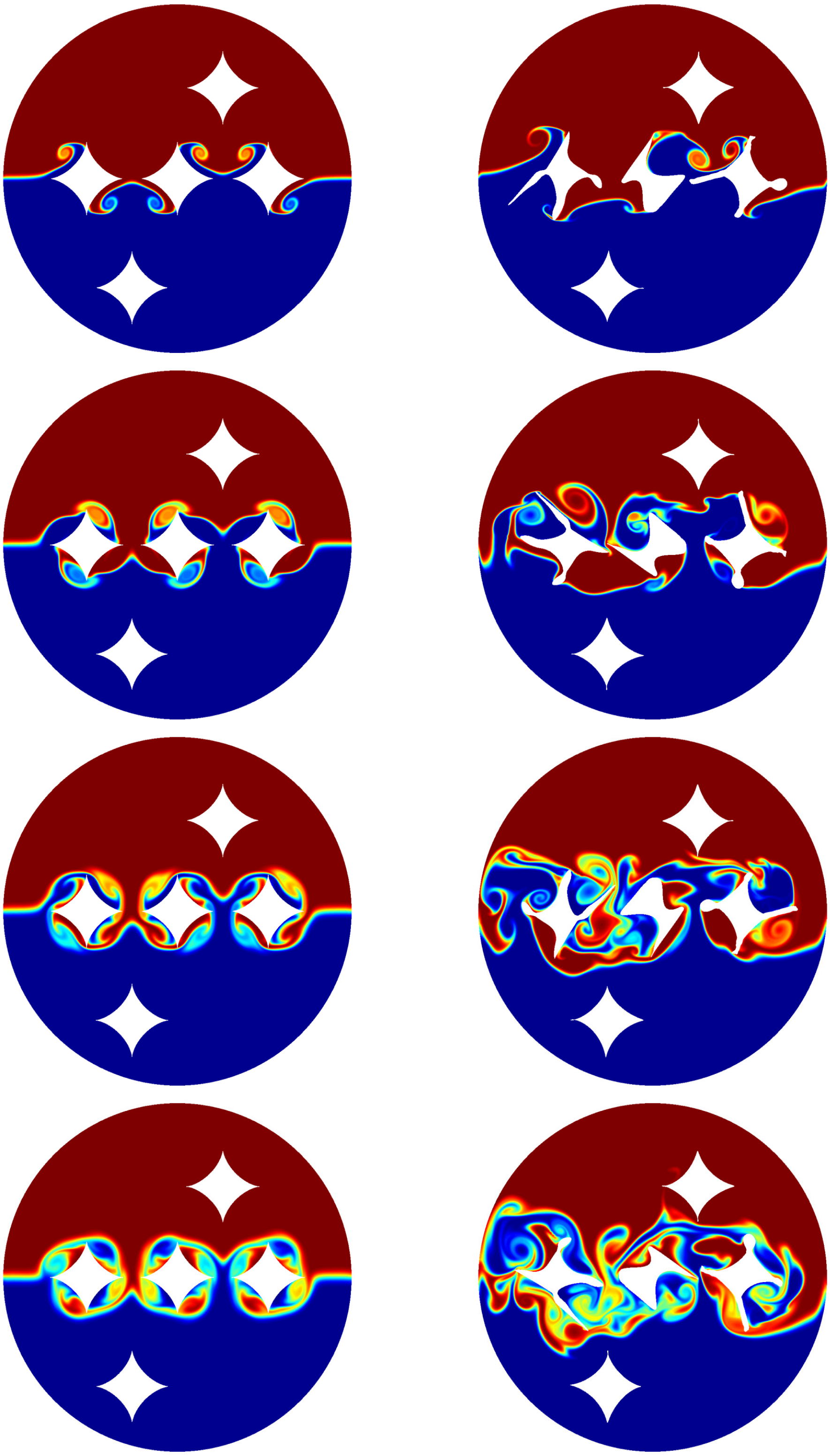


Figure 5.11: Case 3: mixing optimisation using five rotating stirrers. Left column: unoptimised configuration, with snapshots at $t = 2, 4, 6, 8$ (top to bottom). Right column: after four direct-adjoint optimisations, with snapshots at $t = 2, 4, 6, 8$ (top to bottom). For videos of these scenarios please refer to `Shape5NoOpt.mp4` and `Shape5Opt.mp4` for the left and right column, respectively.

6

Results 3: Optimisation of stirrer velocity along a path

6.1 Set-up of the problem

After two optimisation configurations where only time independent control parameters were considered, we now turn our attention to the more complicated case of time dependent control variables. This additional complexity represents the third, and final, case that demonstrates the effectiveness and efficiency of the optimisation framework in light of this considerable challenge. To this end, we will attempt to optimise the velocity of two stirrers along prescribed circular paths.

6.1.1 Path parameterisation and optimality condition

We follow the set-up shown in figure 2.1 with two circular cylinders of radius $\|a_{1,2}\| = 1$, moving on two concentric circular paths of radius $r_1 = 3.5$ and $r_2 = 1.5$ and embedded in a circular (stationary) vessel of radius $R = 10$. The Reynolds number and Péclet number are chosen as $Re = Pe = 1000$.

We note that the earlier definition of the stirrer velocities (2.9) does not include the circular path parameterisation. We, therefore, need to extend this definition by including extra terms that encode the directionality along the path. We define

$$l_1(\phi) = \sin(\phi), \tag{6.1a}$$
$$l_2(\phi) = \cos(\phi), \tag{6.1b}$$

where ϕ is the angle swept along the path of the circle. Using the above, the new parameterisation of the solid velocity of ith stirrer becomes

$$(\boldsymbol{u}_{s,i})_1 = \boldsymbol{r}_i(\boldsymbol{x})\omega_{C_i}(t)l_1(\varphi_i(t)) + \omega_i\boldsymbol{h}_1(\phi), \tag{6.2a}$$
$$(\boldsymbol{u}_{s,i})_2 = \boldsymbol{r}_i(\boldsymbol{x})\omega_{C_i}(t)l_2(\varphi_i(t)) + \omega_i\boldsymbol{h}_2(\phi), \tag{6.2b}$$

where $\boldsymbol{h}$ is defined in equation (2.10) and φ is the integral of the rotational speed along the path, i.e.,

$$\varphi_i(t) = \int_0^t \omega_{C_i}(s)\mathrm{d}s. \tag{6.3}$$

As we have altered the nature of the velocity optimisation, we need to modify optimality condition (2.75) accordingly. The derivation of this new condition does not differ greatly from analogous derivations presented in appendix C, and hence we will only present the result here. We have acquired an extra term which projects the previous vector quantity onto a directional vector, resulting in an overall scalar expression. We obtain

$$\omega_{C_i}^{\dagger} = r_i \left[l_j(\varphi(t)) + \frac{\omega_{C_i}}{\dot{\omega}_{C_i}}\frac{\partial l_j}{\partial \varphi} \right] \chi_i^H \left(\frac{\Pi_j^{\dagger}}{C_\eta} - (\theta^{\dagger} \circ [\mathsf{A}_j \theta]) \right) + \left(\frac{\partial \mathcal{J}}{\partial \omega_{C_i}} \right)^H . \tag{6.4}$$

Due to the time dependent nature of our optimisation, the time horizons over which (i) the control strategy is applied, and (ii) over which gradient information is gathered, become important factors. The former time interval determines the window given to the stirrers to be active mixers; after this window has passed, the motion of the stirrers will stop, and only the remaining inertia of the fluid and diffusion will contribute to further mixing. The latter time interval determines the amount of information extracted from the evolution process that is used to compute an enhanced stirring protocol (applied over the former time window). The control horizon may be chosen shorter than the information (predictive) horizon: in this case, a time-compressed strategy will be employed that accounts for, and optimises over, a more expansive time window. In our case, we will juxtapose a short-term strategy with $T_{\text{control}} = 1$ and a longer-time strategy with $T_{\text{control}} = 8$ and assess

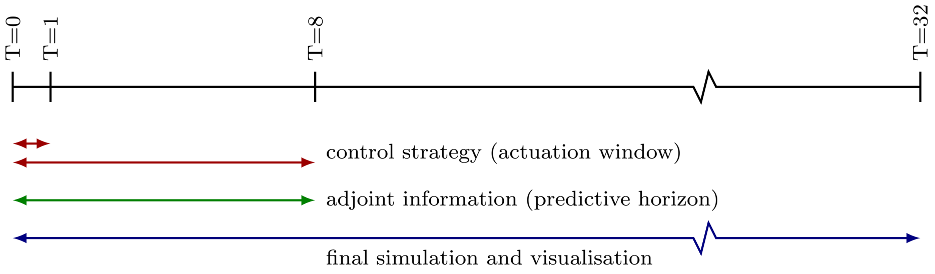

T=0
T=1
T=8
T=32
control strategy (actuation window)
adjoint information (predictive horizon)
final simulation and visualisation

Figure 6.1: Sketch of time horizons for the optimisation problem. A control strategy (red) is applied over two control horizons, $T = 1$ (for short-time control) and $T = 8$ (for long-time control). The gradient information about the flow development (green), encoded in the adjoint variables, is gathered over a predictive horizon of $T = 8$. The final simulation, based on the optimised strategy, is performed over $T = 32$ non-dimensional time units (blue).

the optimised strategies in either case. Both protocols, however, have access to information over a temporal interval of $T_{\text{info}} = 8$. Finally, the simulations have been continued over $T_{\text{sim}} = 32$ to track the further development of the instigated mixing processes; rest inertia and diffusion will remain the only mechanisms during this stage. The summary of this choice of parameters is sketched in figure 6.1.

6.1.2 Cost functional and constraints

Taking the success of the previous results into account, we elect to use the mix-norm as our mixing measure again. We can then state the optimisation problem as finding a time-dependent velocity protocol, $\omega_{C_i}(t)$, for each of the two stirrers such that the mix-norm of the passive scalar is minimised over a prescribed time horizon. This minimum has to be achieved while satisfying the governing equations and respecting constraints and bounds on the stirrer velocities. Previously, we considered either simple time independent or energy independent optimisation parameters. Therefore, no additional considerations on the constraints needed to be made. However, the test cases we consider in this section are of an arrangement that can, and will, lead to numerically as well as industrially unfeasible results.

Therefore, the optimisation, mathematically, becomes

$$\min \left\{ \int_0^T \|\theta\|_{\text{mix}} \, dt \right\} \tag{6.5a}$$

$$\text{subject to equations (2.7)} \tag{6.5b}$$

$$\text{and} \quad \int_0^T \sum_k \|\mathbf{u}_{s,k}\|^2 \, dt \leq E_0 \qquad \text{(Energy penalisation)} \tag{6.5c}$$

$$\text{and} \quad \mathbf{u}_{s,\text{lower}} \leq \mathbf{u}_{s,k} \leq \mathbf{u}_{s,\text{upper}} \qquad k = 1, 2. \tag{6.5d}$$

$$\text{and} \quad \boldsymbol{a}_{s,\text{lower}} \leq \boldsymbol{a}_{s,k} \leq \boldsymbol{a}_{s,\text{upper}} \qquad k = 1, 2. \tag{6.5e}$$

The constraints on the stirrer strategy are threefold: the first constraint limits the L^2-norm of $\boldsymbol{u}_s$, i.e., the kinetic energy of the velocity along the stirrers' paths, expended over the time horizon T to a maximum value E_0, the second and third impose upper and lower bounds directly on the stirrer velocities and accelerations, respectively. All restrictions could conceivably stem from mechanical limitations of the mixing apparatus. We will be considering the constraints on a case by case basis to study the effects they have on the optimisation result.

Additional constraints that need to be enforced are incorporated into the gradient-based optimisation routine. This is accomplished by projections and thresholding. In this case, the gradient – computed from the adjoint equations and the optimality condition, without imposed constraints – is projected and properly curtailed to comply with energy constraints and velocity bounds.

6.2 Presentation of optimisation results

Before proceeding to the various optimisation studies, it is instructive to reflect upon possible mixing mechanisms, given the setup in figure 2.1. The most obvious strategy for mixing a binary fluid consists of a plunging motion, where the cylinders push through the initial interface, distort it and drag fluid one into regions occupied by fluid two, and vice versa (6.2*a–c*). This type of strategy is nearly exclusively implemented in industrial mixers. Despite its omnipresence in applications, alternative strategies are often equally or more effective, foremost among them vortex shedding due to unsteady and abrupt motion of the stirrers, informally denoted as the vortex cannon strategy (6.2*d-f*). In this case, the stirrer generates a sequence of start-up and stopping vortices by rapid oscillations or abrupt directional changes along the circular paths. The shed vortices then act as effective autonomous mixers that, once they reach the initial or distorted interface, further deform the passive scalar field and locally (and globally) reduce the mix-norm. In this manner, a single stirrer can clone 'fluid stirrers' (shed vortices) and thus multiply its mixing effectiveness. In a further possible strategy, vortices can be generated in the fluid that collide with each other and thus generate filaments, which are then subject to more rapid diffusion and homogenisation (6.2*g*-*i*). Of course, this vortex collision strategy is strongly dependent on the initial condition of the passive scalar – and for this reason, less transferable to a general, realistic mixing strategy –, nonetheless, within our computational framework, it is a viable and pervasive strategy utilised by our direct-adjoint algorithm. A far more transferable mixing strategy is the collision of vortical structures with the outer wall (6.2*j*-*l*) whereby a large fluid element is broken up into smaller elements which further interact with other vortices and are subject to increased diffusion due to the breakdown in scales. Finally, the embedded physical stirrers can themselves interplay with the vortical structures they generate, acting as obstructions in the path of vortices (6.2*m*-*o*). A collision between a stirrer and a vortex will split the vortex and yield smaller scales, hence contributing to a decrease in the mix-norm. This final strategy will continue to cause a moderate breakdown in scales, even after the control window has closed and no more stirring motion is allowed.

Given these five fundamental strategies, illustrated in figure 6.2 with samples from our simulations, the direct-adjoint looping algorithm will select from and combine these options into a coherent strategy, given the chosen parameters and user-specified constraints.

6.2.1 Overview of test cases

We will consider six cases, grouped into three examples. Each example consists of a short-time strategy with a rather limited control horizon of $T_{\text{control}} = 1$ and a long-time strategy with a more generous horizon of $T_{\text{control}} = 8$. These two T_{control}-settings will impose noticeable constraints on the choice of strategies, the interplay of dynamic processes and the feasibility of the final protocol.

The three examples further distinguish themselves by the number of external constraints: starting with pure energy constraints, via energy and velocity constraints, to energy, velocity and acceleration constraints. Along this course of action, algorithmic requisites and physical requirements will be encountered and discussed.

We emphasise here the lack of a stopping criterion based on convergence to an optimal strategy. This is due to the nature of the optimisation; as the velocity can be changed at every time-step, we find ourselves in an exceedingly large design space on the order of at least 10^6 design variables. This abundance of dimensions does not easily furnish finding a local minimum, and therefore our stopping criteria

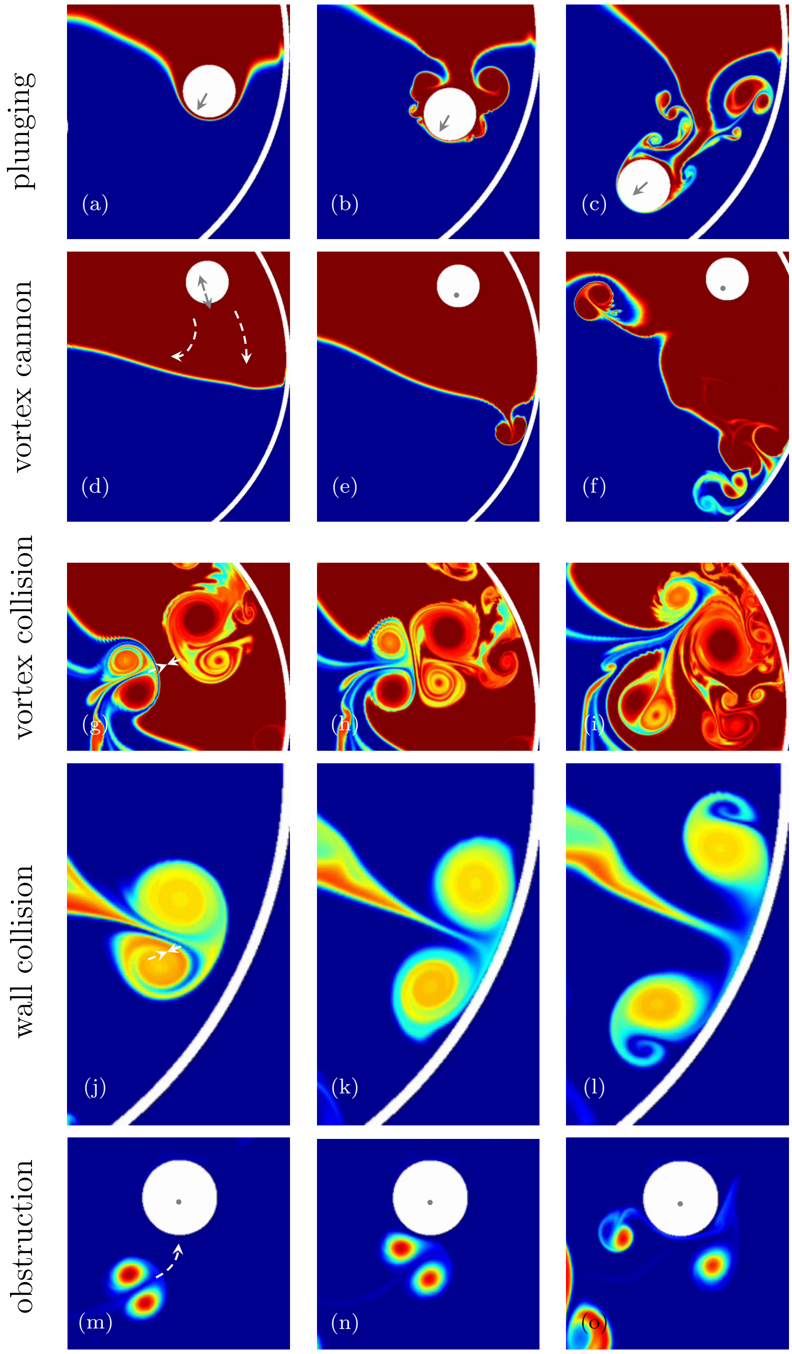


Figure 6.2: Various mixing strategies, from snapshots of the simulations. (a,b,c) Plunging of the stirrer through the interface, (d,e,f) casting of start-stop vortices towards the interface (vortex cannon), (g,h,i) collision of vortices, (j,k,l) collision with the vessel wall, and (m,n,o) breakup of vortical structures by stationary stirrers (obstruction).

is based on the industrial viability of the resulting strategy. Furthermore, the movement of the stirrers produces a large set of small scale structures, which in turn are strongly detected by the adjoint algorithm. This added sensitivity can result in non-negligible noise in the update step, in particular in the longer time horizon cases. Therefore, in some of the cases where $T_{\mathrm{control}} = 8$, the optimisation is stopped prematurely as we are no longer able to ascertain the optimal update step any longer. However, in all cases we nonetheless enhance mixing efficiency.

Since mixing enhancement based on complex stirring strategies is a highly dynamic process – based on a rapid sequence of abundant vortical features – a set of static snapshots cannot do justice to the intricacies of an optimised mixing protocol. For this reason, we once again urge the reader to turn to the animations in the supplemental material.

6.2.2 Cases 1 and 2: optimisation under energy constraints

The first two cases follow a common procedure whereby the cost functional (mix-norm of the passive scalar) of the constrained optimisation is minimised, subject to a penalisation of the control energy that accomplishes this minimum. Since the stirrers' kinetic energy is a measure of effort that goes into the mixing process, we add a corresponding term to the pure mix-norm cost functional. As a consequence, the kinetic energy expended by the stirrers is bounded to a user-specified value.

Figure 6.3 displays the results of our optimisation, visualised by iso-contours of the passive scalar θ at selected time instances. The control horizon is $T_{\mathrm{control}} = 1$. We observed that the optimisation does not utilise the 'plunging' option, as the stirrers remain nearly at their initial position. Instead, the entire energy available to the stirrers is used up in a rapid start-and-stop motion which initially causes multiple small-scale shed vortices that distort the plane interface, collide into each other and the stirrers, and merge into larger-scale vortex structures which eventually achieve good mixing. It is important to stress that for the calculation of this short-time mixing strategy, information about the full dynamics up to $T = 8$ has been incorporated into the optimisation. In other words, the consequences of the limited stirring protocol up to $T = 8$ are known to the optimisation, and adjustments to the control strategy can be made that affect the vortex dynamics beyond its active control window. The evolution of the passive scalar between $T = 8$ and $T = 32$, however, is neither designed nor recognised by the optimisation algorithm; it simply plays out according to the action taken during the control and optimisation windows. We show this further evolution here to illustrate the nature of the mix-norm, our chosen mixing measure. The optimisation with regards to this measure produces the small scales that the diffusion can subsequently take

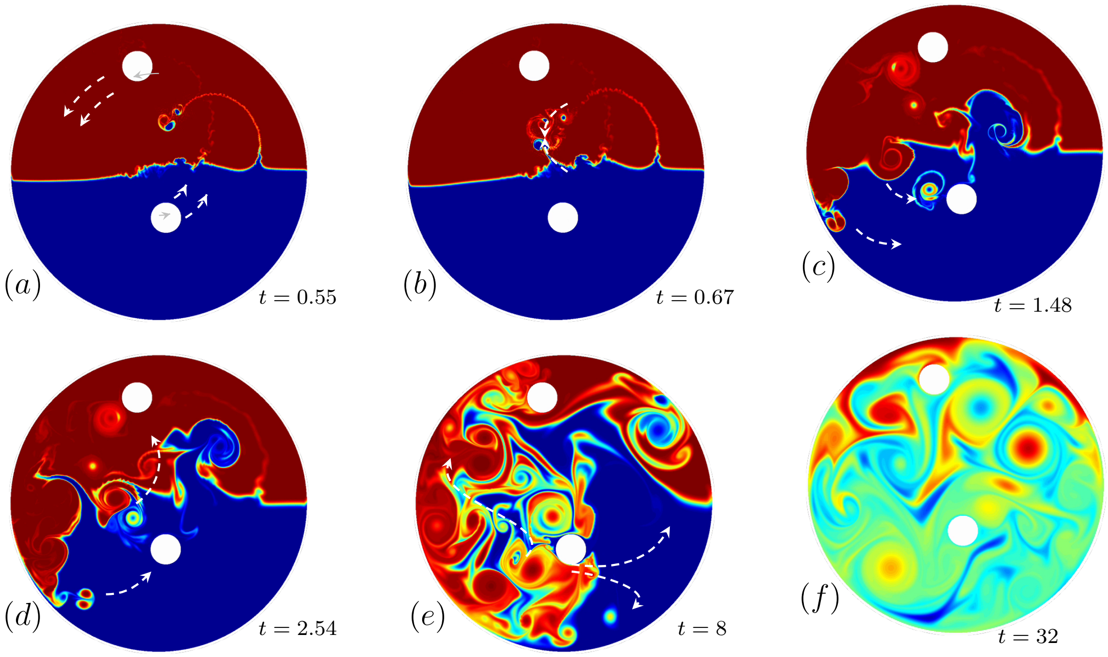


Figure 6.3: Mixing optimisation based on only energy constraints for the stirrers. The time horizon for applying control is $T_{\text{control}} = 1$. Shown are iso-contours of the passive scalar at selected instances. The optimisation algorithm includes information over a time window of $T = 8$. For the video of this scenario please refer to `EnergyPen1.mp4`.

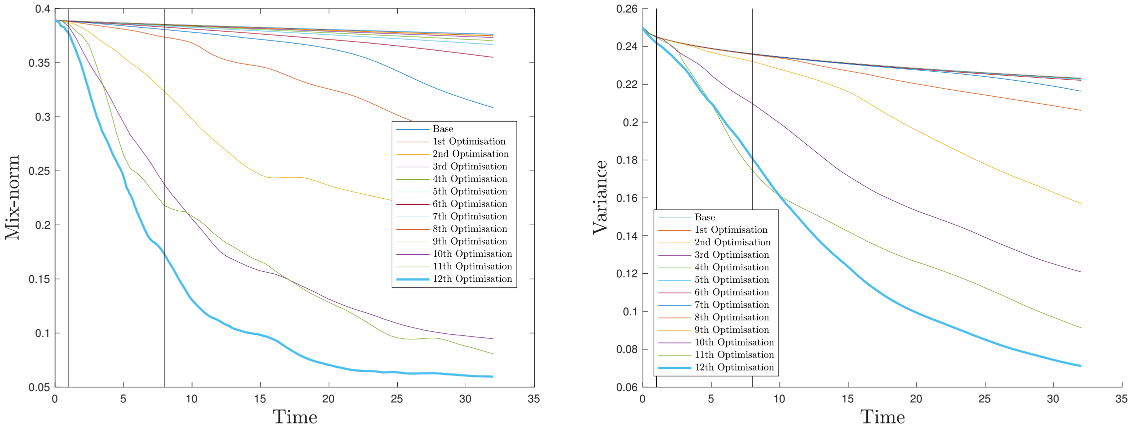


Figure 6.4: Case 1: Energy penalised mixing optimisation under lesser control horizon $T_{\text{control}} = 1$. The black lines signify the control and information time horizons. (a) Mix-norm of the passive scalar versus extended time horizon $t \in [0,\ T = 32]$. (b)Variance of the passive scalar versus extended time horizon $t \in [0,\ T = 32]$.

advantage of in this ‘cool-off’ window.

We conclude that the absence of any plunging option implies the suboptimality of this strategy in achieving an enhanced mixing process. It is thus not pursued as a viable option by the direct-adjoint optimisation technique. A reason for this lies in the gradient-based approach. Once a velocity spike has formed in the mixing strategy, any decrease of this spike would lead to less optimal results and is thus discarded. The further increase in the spike reinforces the vortex cannon strategy. Once this strategy has been identified, the optimisation concentrates on coordinating the path of the vortices such that they interact with each other and the stirrers. The latter part of the stirring strategy includes vortex collisions (see figure 6.3*b*), obstruction by the stirrers (see figure 6.3*c,d*) and collision with the outer wall (see figure 6.3*e*) to yield a well-mixed state at the end of the simulation horizon (figure 6.3*f*). We note from figure 6.4 that a similar clustering of initial mix-norms and variances can be observed as those in section 4. However, once a spike in the velocity has formed, a rapid decrease can be seen in both the variance and the mix-norm. A point of interest lies in the variance graph of figure (6.4. Here we see that the variance of the penultimate optimisation is lower than that of the final one. As we optimised with regard to the mix-norm, the variance value is not of interest at therefore is higher. However, we note that a lower mix-norm value at $T = 8$, nonetheless leads to a lower variance at $T = 32$, and demonstrates the more optimal capabilities of the mix-norm as an optimisation measure.

Increasing the control horizon from $T_{\text{control}} = 1$ to $T_{\text{control}} = 8$ leads to similar

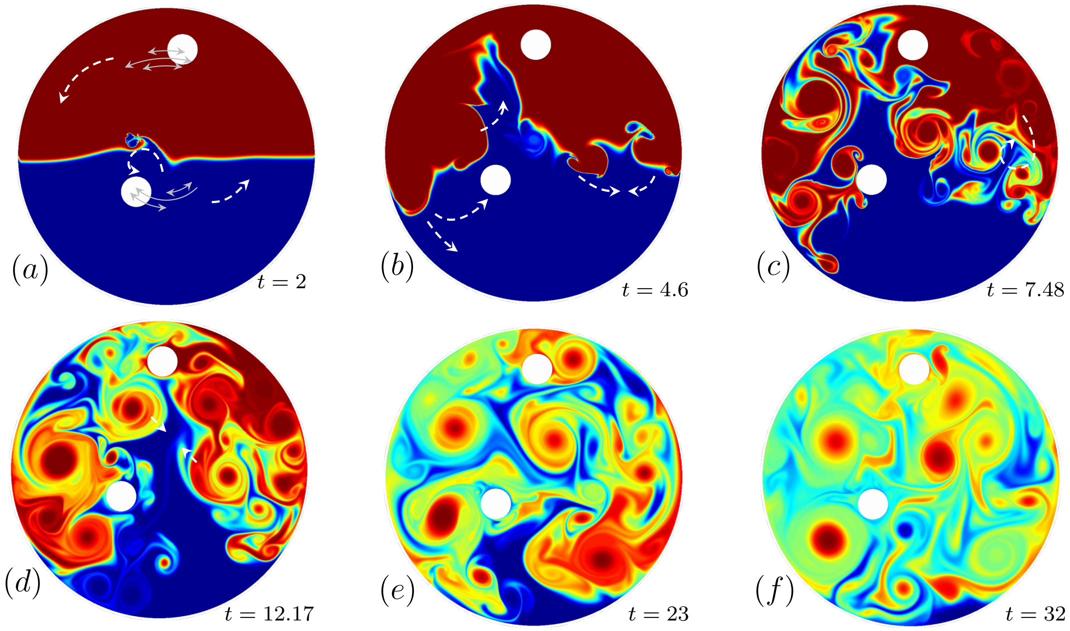


Figure 6.5: Same as figure 6.3, but with an extended control window of $T_{\text{control}} = 8$. For the video of this scenario please refer to `EnergyPen8.mp4`.

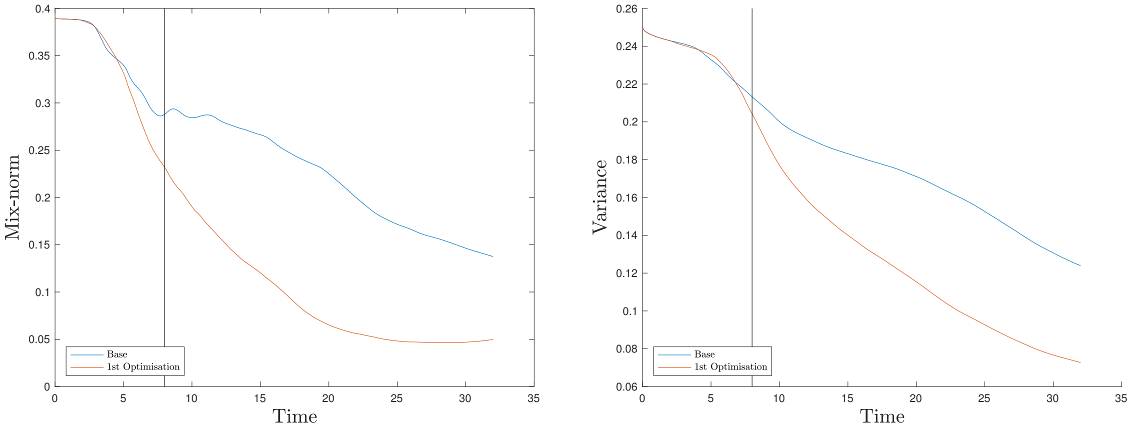


Figure 6.6: Case 2: Same as figure 6.4, but with an extended control window of $T_{\text{control}} = 8$.

conclusions, even though the stirring action by the cylinders is less abrupt and jarring, as multiple velocity spikes of less magnitude are observed. Still, the bulk of the mixing action is achieved by shedding start-and-stop vortices which collide with themselves, secondary vortices and the wall to produce a mixed state in the end. Again, the absence of plunging is noteworthy. This is even more remarkable, as the increased control time horizon would certainly allow the stirrers to approach and reach the interface; yet, they remain close to their initial position. In the interpretation of these results, it may be tempting to conclude that a different initial placement of the cylinders – closer to the initial interface – would have resulted in strategies that included plunging. However, a simulation of the same case (not shown), with the two cylinders starting immersed in the initial interface, came to the same conclusion: while, by design, there is a small amount of plunging in this case, the vast majority of the mix-norm reduction has been accomplished by the shedding of start-and-stop vortices by a vigorous oscillatory motion of either cylinder and a subsequent collision of the generated vortices. The utilisation of the stirrers' energy to shed small 'vortical stirrers' is a better strategy than the distortion of the interface by simply moving through it with the stirrers. From figure 6.6 we stress that only one optimisation was required to achieve this level of mixing efficiency. However, any further optimisations were not possible due to the noise that permeated the high-velocity spikes of the updated strategy.

In both cases, the strategy found by the direct-adjoint optimisation technique will yield increasingly larger velocities, as long as the integrated energy is constant. Eventually, the energy expenditure becomes more and more localised in time, with the stirrers barely moving. This optimisation route is a logical consequence of our current setup. It is closely connected to the semi-norm problem[73]: the mix-norm only contains the passive scalar θ, but does not account for the other dynamic variable, the velocities, in the optimisation. The energy of the stirrers is not sufficient to arrive at realistic stirring protocols that could be implemented in an experimental or industrial setting. To ensure the applicability of our stirrers to real-life settings, more constraints are required.

6.2.3 Cases 3 and 4: optimisation under energy and velocity constraints

Following the findings of the previous section, in the next example we limit the velocity of the stirrers to avoid excessive values of $\mathbf{u}_{s,i}$. This capping of the velocity is implemented by a projection of the raw cost-functional gradient onto control strategies that satisfy the given constraints (see Foures *et al.*[60]). The resulting limit on the stirrer velocities provides a longer time window (up to the control horizon

T_{control}) over which the specified energy can be expended. As a consequence, an extended and smoother movement of the stirrers is expected.

For the shorter control horizon $T_{\text{control}} = 1$, figure 6.7 shows the outcome of our optimisation. To keep the results comparable with the longer time interval optimisation, we ensure that integration of the stirrer velocities over the time window is the same. The top stirrer starts by an oscillatory motion, creating start-and-stop vortices. The capping of its velocity, however, keeps the shed vortices within bounds; nonetheless, the optimality of the 'vortex cannon' strategy can still be exploited. Both stirrers then move closer to the (already distorted) interface. But rather than plunging through it, they abruptly stop short of it and let the overtaking stop-vortices carry out the distortion of the interface and the subsequent mixing. Again, the optimisation algorithm selects the mixing by shed vortices over the plunging of the stirrers through the interface. The remaining mixing process is characterised by vortex collisions (see figure 6.7*a,d*), collisions with the wall (see figure 6.7*c*) and stirrer obstruction (see figure 6.7*c,d*).

Extending the horizon T_{control} over which the control is applied results in a change of strategy (see figure 6.9). The top stirrer now plunges through the interface – but not before stopping and starting on its circular path towards it. This uneven motion creates more vortical structures in the stirrer's wake that add to the sole plunging action of the stirrer itself. The result is a far more distorted interface (and consequently a lower mix-norm) than would be generated by a simple traversal. At the end of the motion, a back-and-forth motion is performed to generate, within the given chosen energy and velocity constraints, additional shed vortices that further interact with the interface and other vortical elements. The second stirrer does not follow the strategy of the first. It engages in an oscillatory motion along its circular path and generates, as before, the resulting start-and-stop vortices that distort the interface and interact with the other vortices inside the container. Again, obstruction by the cylinders (see figure 6.9*b,d*) and vortex and wall collisions (see figure 6.9*c,d*) contribute to the continued mixing.

In both cases, a gentler stirring strategy is observed. However, the problem of converging towards a realistic mixing protocol has not been solved completely. While we explicitly avoid highly localised action of the stirrers with excessive velocities, we now tend towards favouring strategies with excessive acceleration. In other words, within our efforts to limit the total expended energy of the stirrers while capping their velocities, the optimisation algorithm tends towards strategies that are characterised by large accelerations (high velocity gradients). This should not come as a surprise as the strength of shed vortices from the stirrers' unsteady motion is proportional to their acceleration. Our imposed constraints do account for energy and velocities, but not velocity gradients, of the stirrers. As a consequence,

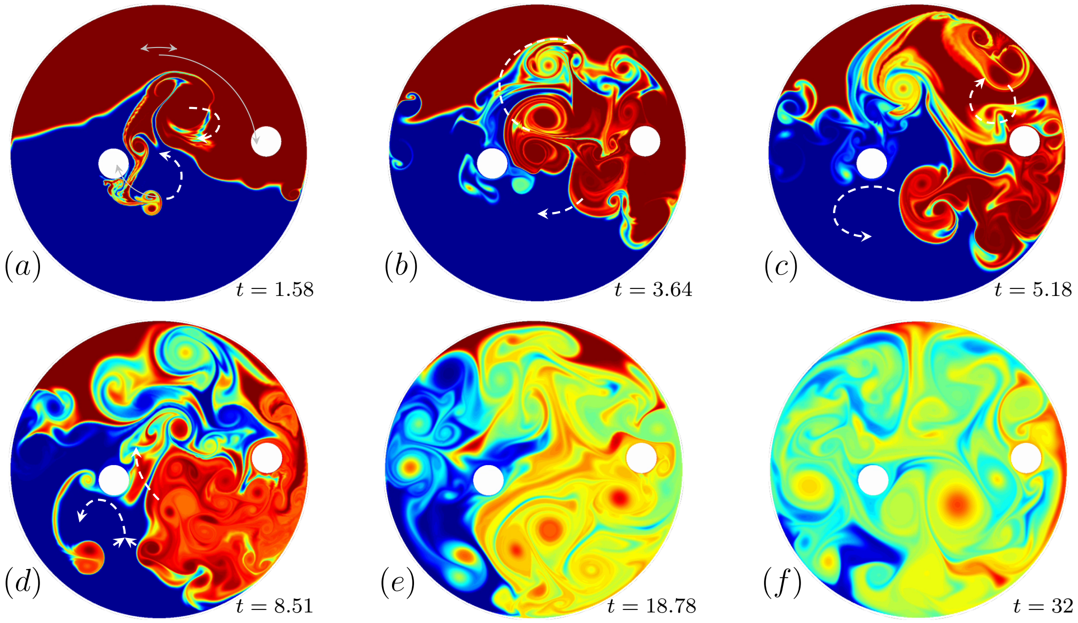


Figure 6.7: Mixing optimisation based on energy and velocity constraints for the stirrers. The time horizon for applying control is $T_{\text{control}} = 1$. Shown are iso-contours of the passive scalar at selected instances. The optimisation algorithm includes information over a time window of $T = 8$. For the video of this scenario please refer to `VelocityPen1.mp4`.

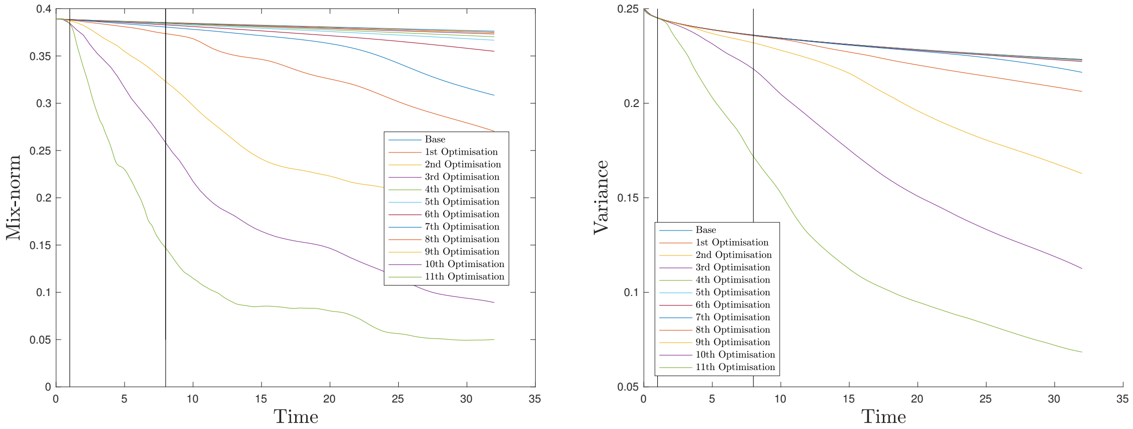


Figure 6.8: Case 3: Energy and velocity penalised mixing optimisation under lesser control horizon $T_{\text{control}} = 1$. The black lines signify the control and information time horizons. (a) Mix-norm of the passive scalar versus extended time horizon $t \in [0,\ T = 32]$. (b)Variance of the passive scalar versus extended time horizon $t \in [0,\ T = 32]$.

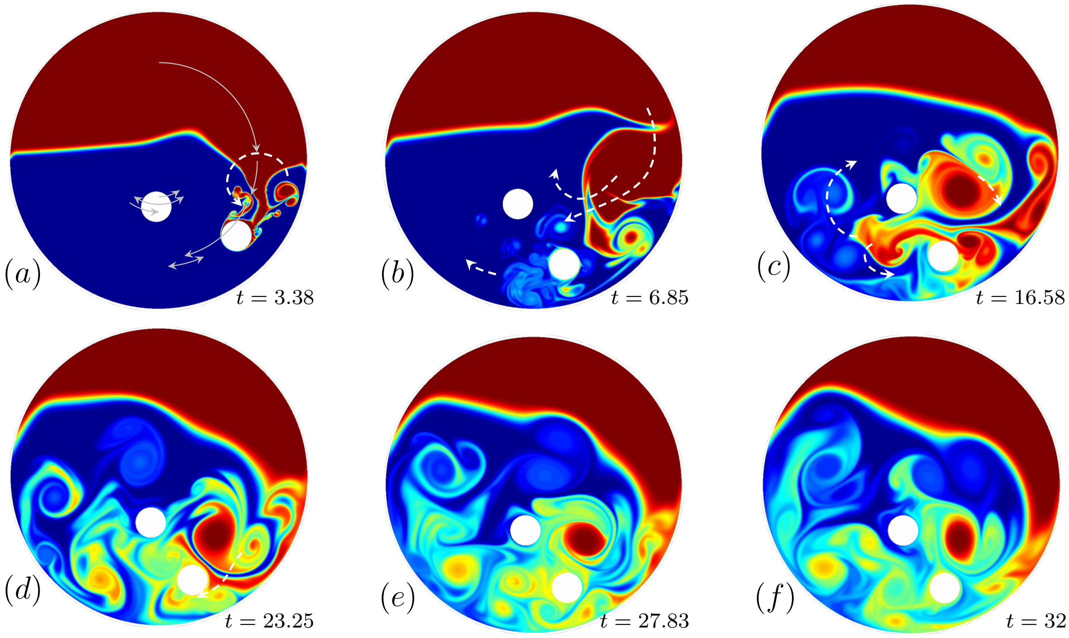


Figure 6.9: Same as figure 6.7, but with an extended control window of $T_{\text{control}} = 8$. For the video of this scenario please refer to `VelocityPen8.mp4`.

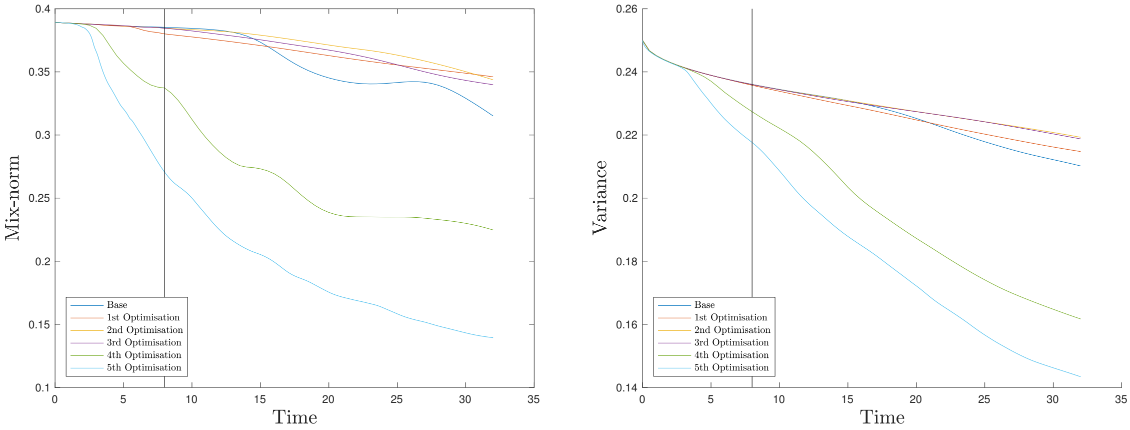


Figure 6.10: Case 4: Same as figure 6.8, but with an extended control window of $T_{\text{control}} = 8$.

we can seed our binary mixture with vortical elements of nearly unlimited strength. Again, this divergence is related to the above-mentioned semi-norm problem: the velocity field of the binary mixture is not accounted for in the mix-norm, and thus the optimisation scheme can achieve high-energy fluid states by highly accelerating stirrers (even though the stirrers' energy and velocities are capped). To limit the velocity of the fluid, we have to limit the acceleration of the stirrers. Again, additional constraints are necessary.

6.2.4 Cases 5 and 6: optimisation under energy, velocity and acceleration constraints

For accomplishing enhanced mixing in binary fluids, the direct-adjoint optimisation technique makes heavy use of an acceleration-based strategy: shed vortices generated by the abrupt motion of the stirring cylinders are injected into both fluids, and their interactions with the interface, themselves and the container wall yield a low mix-norm. A limit on this acceleration will result in a limit on the velocities in either fluid component and thus provide the necessary restriction for a successful semi-norm optimisation. To this end, we augment our optimisation scheme by additional terms accounting for the stirrers' acceleration. This type of penalisation is common in deblurring of images where strong gradients are detected and encouraged[74]. In our case, additional projections are used to enforce the acceleration constraints.

For the short-term control with $T_{\text{control}} = 1$ (see figure 6.11), the optimal strategy now includes a plunging of the first cylinder through the interface, while the second cylinder continues in a straight manner towards the interface but stops short of it. The wake vortices of the first cylinder, as well as the (weaker) start-and-stop vortices of both cylinders, are responsible for the bulk of the mixing. It can be observed that the positioning of the second cylinder places it in a highly sensitive location, i.e., where it can 'nudge' one of the start-up vortices of stirrer one onto a path where it collides with a secondary start-up vortex and subsequently generates a significant amount of small structures (6.2 g-j). Therefore we note, as before, complex vortex collisions (see figure 6.11*c*), stirrer obstruction (see figure 6.11*c,d*) and wall interactions (see figure 6.11*d*) contribute greatly to the breakdown of scales, the generation of filaments (see figure 6.11*e*) and the eventual mixing of the binary fluid (see figure 6.11*f*).

A longer control horizon of $T_{\text{control}} = 8$ yields a more varied stirring protocol. The first cylinder makes a farther excursion, plunging through the interface (not without stopping to generate additional shed vortices close to the interface) before stopping close to the interface and shedding two stop vortices. The second cylinder

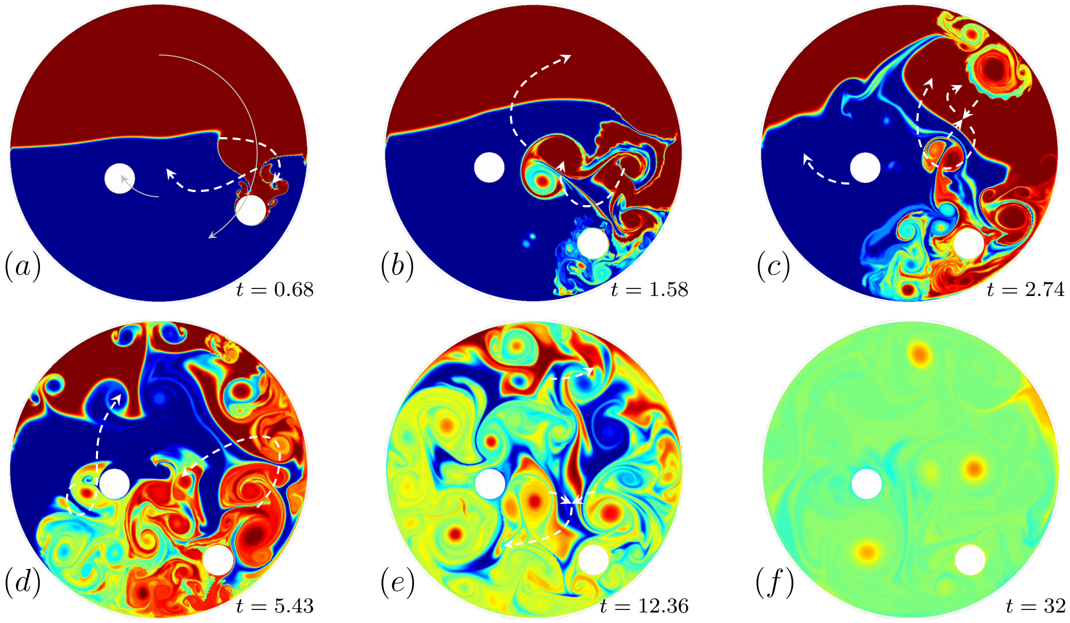


Figure 6.11: Mixing optimisation based on energy, velocity and acceleration constraints for the stirrers. The time horizon for applying control is $T_{\text{control}} = 1$. Shown are iso-contours of the passive scalar at selected instances. The optimisation algorithm includes information over a time window of $T = 8$. For the video of this scenario please refer to `AccelerationPen1.mp4`.

first approaches the interface, ejects a stop vortex before reversing and stopping short of the interface with another stop vortex. The generated structures interact with themselves and the wall to break down the binary fluid into a homogeneous mixture, although of less homogeneity (larger mix-norm) than for the short-term strategy. This lower homogeneity can be attributed to the more restrictive velocity range (which is due to the constraint of equivalent energy budget , i.e., L^2-norm of the velocity of the centre of the stirrers, across both time horizons) than the larger time-horizon strategy. This lower velocity maximum, combined with the limitation on acceleration, inhibits the same amount of vorticity shedding than the lower time-horizon case. Therefore, mixing is not as efficient.

At no point during either optimisation has the energy, velocity or acceleration of the stirrers exceeded the specified limits. As a consequence, these latter strategies are amenable to implementation in an experimental or industrial setting.

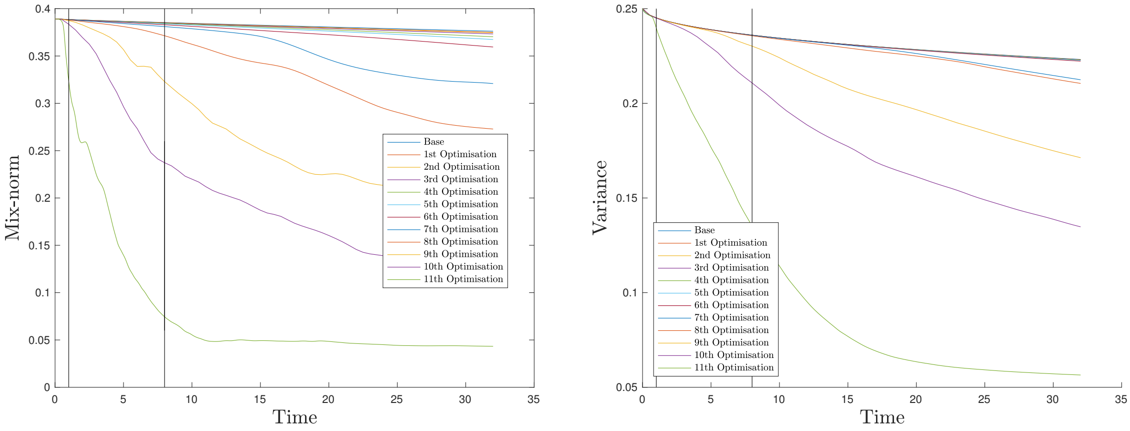


Figure 6.12: Case 5: Energy, velocity and acceleration penalised mixing optimisation under lesser control horizon $T_{\text{control}} = 1$. The black lines signify the control and information time horizons. (a) Mix-norm of the passive scalar versus extended time horizon $t \in [0,\ T = 32]$. (b)Variance of the passive scalar versus extended time horizon $t \in [0,\ T = 32]$.

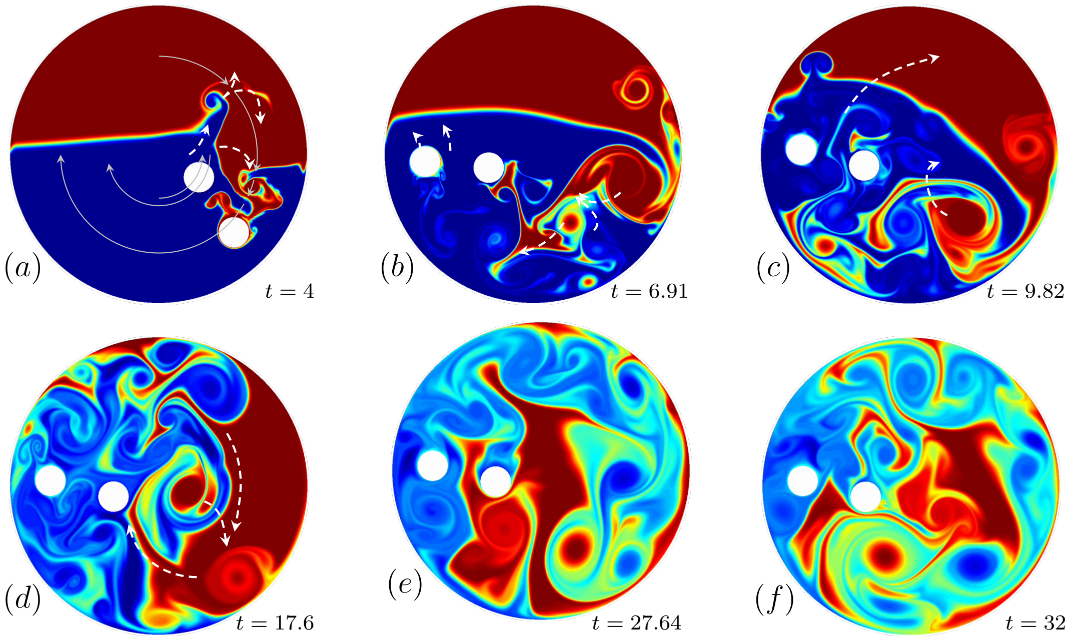


Figure 6.13: Same as figure 6.11, but with an extended control window of $T_{\text{control}} = 8$. For the video of this scenario please refer to `AccelerationPen8.mp4`.

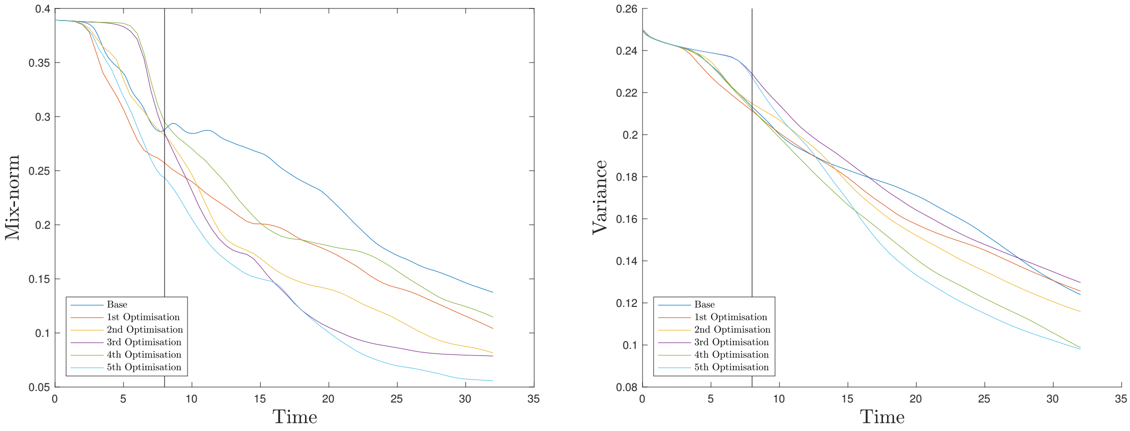


Figure 6.14: Case 6: Same as figure 6.12, but with an extended control window of $T_{\text{control}} = 8$.

7
Conclusion

In this thesis, we have presented a theoretical and computational framework that allows for the design of complicated stirrer geometries, as well as stirring strategies, to optimise the mixing of binary Newtonian fluids. This framework has subsequently been demonstrated on three test cases of increasing complexity.

The theoretical section presented a detailed study and derivation of a dual system for the governing Navier-Stokes equations, augmented to accommodate boundary conditions for our complex mixing geometry. Initially, we took the continuous approach and discussed the implementation into a computational framework. Considering this direct-dual system without the underlying fluid-structure algorithm allows for different discretisation approaches to be applied. As a representative example, we combined this framework with a Brinkman-penalisation style technique to illustrate how boundary condition terms would enter this specific system. The final result, however, has been found unsuitable for our needs. Therefore, we turned our attention to a different approach, namely deriving the adjoint system for the Brinkman-penalised semi-discretised Navier-Stokes equations. The resulting adjoint equations, which naturally inherit the boundary conditions of the direct system, allowed for a flexible and convenient computational implementation that is able to deal with a wide variety of mixing configuration. This full derivation has been performed without specifying a unique cost-functional, thus retaining a high degree of generality and demonstrating the independence of our direct-dual system from user-specified constraints and/or mixing measures.

Following on from the analytic, we then presented the computational framework, which employs Brinkman-penalisation to embed the stirrer elements into the surrounding fluid and to track their interaction with it. The resulting PDE-constrained optimisation problem is then recast into a direct-adjoint (or direct-dual) formulation, which is subsequently solved iteratively, employing a checkpointing technique due the linkage of the direct and adjoint problem. Particular attention has been paid to the explicit formulation of path and shape gradients based on the penalised formulation. These gradients are then used in an optimisation scheme to enhance the mixing efficiency (in our case, initially variance and later, a mix-norm), while observing user-specified control-energy bounds.

In terms of the numerical implementation, the framework was constructed in such a way that it allows for the efficient optimisation of mixing strategies of binary fluids by any number of moving stirrers within a circular container. Additionally flexibility has been realised by including a solid stirrer class (defined as a *structure* in `FORTRAN 90`) which is easily extendable beyond the cases considered in this thesis. This class aids future endeavours in optimising across a wide variety of control parameters in design space. We showcased this flexibility by considering four separate control parameters of increasing complexity across three test cases; the computational framework succeeded in enhancing mixing efficiency for all cases.

The three test cases covered different aspects of the control space: §4 validated the effectiveness of the presented approach in the form of a feasibility study with simple shape and rotation optimisation, §5 demonstrated successful mixing enhancement by stirrers of more exotic shapes, and §6 ventured into time dependent control by optimising the velocity along given stirrer paths. In all cases across the three scenarios a significant improvement in mixing efficiency could be accomplished, and the optimisation algorithm showed pleasing robustness in finding more optimal solutions. Particularly in the latter two sections (§5 and §6), several interesting conclusions pertaining to optimal mixing strategies could be made. In both cases vortices and their mutual interactions dominated the mixing mechanism. In contrast to the restricted paths of the stirrers, these fluid elements are able to move freely in the field and therefore can affect areas of the fluid that are inaccessible to the stirrers but that are highly beneficial to mixing enhancement. In the case of shape optimisation, this effect could be observed by the convergence of the geometries towards protuberances and the associated concavities for each of the stirrers. These two geometric components generated areas of high fluid recirculation while the stirrers were rotating and lead to the generation and shedding of vortices. The vortices then interacted and collided with previously and subsequently generated vortices, further distorting the interface and creating small scales for their subsequent diffusion. A similar process has been found for the velocity optimisation, §6, where the 'vortex cannon' and vortex collisions exemplified this vortex-based mixing strategy. A further similarity between the two scenarios lies in the collaborative mixing strategies when multiple stirrers are optimised. The placement of stirrers in locations of high sensitivity to act as obstacles for impinging vortices can be juxtaposed to the interlocking gear design of the optimised shapes of §5. In this latter case the cusp of a rotating stirrer is placed directly in the path of a vortex being shed by another solid. This co-operative strategy requires little extra kinetic energy in the form of additional velocity, but can yield a substantial increase in mixing efficiency. Further mixing studies may take advantage of a multi-stirrer configuration and exploit this collaboration of stirrers to further enhance mixing.

A final observation relates directly to the energy, velocity and acceleration constraints that were required to arrive at realistic mixing strategies. Each sequential constraint resulted in a different mixing strategy (of different degrees of optimality). This observation implies that a trade-off needs to be made between the most optimal strategy and industrial realisability. Yet, within the limitations of industrial constraints significant mixing enhancement has been found in all cases, and the flexibility of the computational framework allows for further improvements as cost functionals can be designed and implemented that focus on other mixing characteristics.

Despite these promising results, many challenges remain. The most important one is linked to the application of gradient-based optimisation to a non-convex problem. As a result, only a local minimum can be guaranteed by our algorithm. In isolated and more obvious cases, user input (see case two of §4) is able to nudge the convergence towards a more optimal strategy. However, in configurations or design spaces which defy intuition, this nudging is exceedingly difficult as a clear path towards a better optimum is not transparent. In these cases, 'nudging' could be accomplished more objectively by coupling our algorithm to an annealing-type stochastic algorithm[75]. The excessive computational cost associated with these methods excludes their usability in our case. Nonetheless, local optima should not be discarded as even small efficiency increases equate to substantial gains in efficiency and profits.

A further challenge arises from the application of the Fourier shape parameterisation. The thickening and untwisting procedures we imposed were embedded via a projection approach after the optimisation step has taken place. While this approach achieved a significant increase in mixing efficiency, the resulting shapes, observing the imposed constraints, were not strictly optimal. Mathematically these constraints should be incorporated into the cost functional; the details of this operation, however, remain an open challenge.

While keeping these challenges in mind, the investigations presented in this thesis point towards several areas where this optimisation framework can excel and achieve enhanced mixing efficiency.

In this study, we considered the optimisation of stirrer topology and path velocity as separate optimisation goals. A natural extension consists in a cooperative combination of these two parameters, i.e., the simultaneous optimisation of stirrer geometry and their path velocity along a given trajectory. Exploring this combined design space will also give insight into the underlying mechanisms that drive mixing, e.g., flapping, plunging, heaving or rotation, without much additional computational cost.

Extending the Fourier representation of the shape, the equivalent mathematical representation based on parametric curves can also be applied to optimise the trajectory of the stirrers. Throughout this thesis, we prescribed the stirrers' paths as circular curves. The Fourier-based parameterisation for the stirrers' paths opens new areas of the design. For example the option of have Lissajous curves[76] or figure eight curves for each stirrer could yield pertinent and interesting results, with potentially novel and non-intuitive mixing strategies. This approach has the added benefit that for path optimisation, the constraints of untwisting and thickening of the geometry encountered earlier can be safely ignored since any continuous, and even self-intersecting, trajectory is a viable mixing strategy. However, care must be taken to avoid collision among stirrers and/or the wall. Currently, there is a lack of these types of studies in the pertinent literature.

Another possible future area of study could focus on the the effects of the wall. In §6, we noted that the wall played a significant role in vortex collisions, and consequently, the strategic placement of corrugations at locations where vortices impacted may lead to an additional increase in mixing effectiveness. The Fourier parameterisation lends itself naturally to this approach, and considering non-circular and corrugated bounding walls could be a worthwhile effort to explore. Moreover, the shape of the wall is not the only characteristic of interest. As can be seen from figure 3.3, the numerical implementation is capable of simulating a moving wall. Therefore, the time dependent rotation of the wall, akin to a shaking of the container, is a further control quantity that could enhance mixing efficiency.

Aside from the geometry, we can gain further insight into mixing optimisation by considering parameter regimes that differ from the one studied in this thesis. This includes two fundamental directions of exploration, both related to the material properties of the fluid. The first direction still considers the fluids as Newtonian, but changes both Reynolds and Péclet numbers to simulate different industrial fluids. The second direction studies the two fluids as non-Newtonian with an algebraic expression linking the state variable to the strain tensor. This covers a majority of fluids in industrial mixing processes.

All the above extensions follow the two-dimensional nature of the test cases studied in this thesis and ignore three-dimensional effects. The computational framework is highly parallelised and scales well to many cores on parallel computer architectures; scalable performance for large-scale optimisations of binary mixing problems is thus expected. The implementation of the governing equations already allows for three-dimensional flow fields. However, non-negligible modifications are required to describe three-dimensional stirrer geometries and container walls.

In view of this exceedingly wide scope of possibilities, this thesis has laid the foundation for their exploration. Continued research in this direction holds great promise for significant and robust improvement in mixing strategies.

References

[1] G. I. Taylor, The formation of emulsions in definable fields of flow, Proceedings of the Royal Society of London. Series A, containing papers of a mathematical and physical character 146 (858) (1934) 501–523.

[2] R. R. Long, A theory of mixing in a stably stratified fluid, Journal of Fluid Mechanics 84 (1) (1978) 113124. doi:10.1017/S0022112078000063.

[3] W. Peltier, C. Caulfield, Mixing efficiency in stratified shear flows, Annual review of fluid mechanics 35 (1) (2003) 135–167.

[4] F. Marcotte, C. Caulfield, Optimal mixing in two-dimensional stratified plane Poiseuille flow at finite Péclet and Richardson numbers, Journal of Fluid Mechanics 853 (2018) 359385. doi:10.1017/jfm.2018.565.

[5] R. Spencer, R. Wiley, The mixing of very viscous liquids, Journal of Colloid Science 6 (2) (1951) 133–145.

[6] W.-L. Chien, H. Rising, J. Ottino, Laminar mixing and chaotic mixing in several cavity flows, Journal of Fluid Mechanics 170 (1986) 355–377.

[7] J. Ottino, C. Leong, H. Rising, P. Swanson, Morphological structures produced by mixing in chaotic flows, Nature 333 (6172) (1988) 419.

[8] W. Mohr, R. Saxton, C. Jepson, Mixing in laminar-flow systems, Industrial & Engineering Chemistry 49 (11) (1957) 1855–1856.

[9] S. Corrsin, Simple theory of an idealized turbulent mixer, AIChE Journal 3 (3) (1957) 329–330.

[10] S. Corrsin, M. Karweit, Fluid line growth in grid-generated isotropic turbulence, Journal of Fluid Mechanics 39 (1) (1969) 87–96.

[11] G. K. Batchelor, The effect of homogeneous turbulence on material lines and surfaces, Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 213 (1114) (1952) 349–366.

[12] P. E. Dimotakis, H. J. Catrakis, Turbulence, fractals, and mixing, in: Mixing, Springer, 1999, pp. 59–143.

[13] P. E. Dimotakis, The mixing transition in turbulent flows, Journal of Fluid Mechanics 409 (2000) 69–98.

[14] O. M. Aamo, M. Krstic, Flow control by feedback: stabilization and mixing, Springer Science & Business Media, 2013.

[15] M. A. Jalali, A. Khoshnood, M.-R. Alam, Microswimmer-induced chaotic mixing, Journal of Fluid Mechanics 779 (2015) 669–683.

[16] P. Linden, Mixing in stratified fluids, Geophysical & Astrophysical Fluid Dynamics 13 (1) (1979) 3–23.

[17] W. Tang, C. Caulfield, R. Kerswell, A prediction for the optimal stratification for turbulent mixing, Journal of Fluid Mechanics 634 (2009) 487–497.

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

[19] G. Tryggvason, H. Aref, Numerical experiments on Hele-Shaw flow with a sharp interface, Journal of Fluid Mechanics 136 (1983) 1–30.

[20] G. Mathew, I. Mezić, L. Petzold, A multiscale measure for mixing, Physica D: Nonlinear Phenomena 211 (1–2) (2005) 23–46.

[21] V. Uhl, Mixing V1: Theory And Practice, Elsevier, 2012.

[22] E. L. Paul, V. A. Atiemo-Obeng, S. M. Kresta, Handbook of industrial mixing: science and practice, John Wiley & Sons, 2004.

[23] J. M. Ottino, The kinematics of mixing: stretching, chaos, and transport, Vol. 3, Cambridge university press, 1989.

[24] F. Lekien, C. Coulliette, A. Mariano, E. Ryan, L. Shay, G. Haller, J. Marsden, Pollution release tied to invariant manifolds: A case study for the coast of Florida, Physica D: Nonlinear Phenomena 210 (1-2) (2005) 1–20.

[25] G. Hunt, P. Linden, The fluid mechanics of natural ventilation – displacement ventilation by buoyancy-driven flows assisted by wind, Buildg. and Environm. 34 (6) (1999) 707–720.

[26] P. Linden, The fluid mechanics of natural ventilation, Annu. Rev. Fluid Mech. 31 (1) (1999) 201–238.

[27] A. Annaswamy, A. Ghoniem, Active control in combustion systems, IEEE Contr. Sys. 15 (6) (1995) 49–63.

[28] V. Hessel, H. Löwe, F. Schönfeld, Micromixers – a review on passive and active mixing principles, in: Chemical Engineering Sciences, Vol. 60, 2005, pp. 2479–2501.

[29] N.-T. Nguyen, Z. Wu, Micromixers: a review, J. Micromech. Microeng. 15 (2) (2005) R1–R16.

[30] H. Aref, Stirring by chaotic advection, Journal of Fluid Mechanics 143 (1) (1984) 1–21.

[31] C. Mattiussi, The finite volume, finite element, and finite difference methods as numerical methods for physical field problems, Adv. Imag. and Elec. Phys. 113 (2000) 1–146.

[32] M. Glowinski, T. Pan, R. Wells Jr., X. Zhou, Wavelet and finite element solutions for the Neumann problem using fictitious domains, Journal of Computational Physics. 126 (1) (1996) 40–51.

[33] E. Arquis, J. Caltagirone, Sur les conditions hydrodynamiques au voisinage d'une interface milieu fluide-milieu poreux: applicationa la convection naturelle, C.R. Acad. Sci. Paris II 299 (1984) 1–4.

[34] P. Angot, C.-H. Bruneau, P. Fabrie, A penalization method to take into account obstacles in incompressible viscous flows, Num. Math. 81 (4) (1999) 497–520.

[35] Q. Liu, O. Vasilyev, A Brinkman penalization method for compressible flows in complex geometries, Journal of Computational Physics. 227 (2) (2007) 946–966.

[36] O. Boiron, G. Chiavassa, R. Donat, A high-resolution penalization method for large Mach number flows in the presence of obstacles, Comp. & Fluids 38 (3) (2009) 703–714.

[37] N. Kevlahan, J.-M. Ghidaglia, Computation of turbulent flow past an array of cylinders using a spectral method with Brinkman penalization, Eur. J. Mech. B/Fluids 20 (3) (2001) 333–350.

[38] F. Chantalat, C.-H. Bruneau, C. Galusinski, A. Iollo, Level-set, penalization and Cartesian meshes: A paradigm for inverse problems and optimal design, Journal of Computational Physics. 228 (17) (2009) 6291–6315.

[39] C.-H. Bruneau, F. Chantalat, A. Iollo, B. Jordi, I. Mortazavi, Modelling and shape optimization of an actuator, Struct. Multidisc. Optim. 48 (6) (2013) 1143–1151.

[40] M. M. Hejlesen, P. Koumoutsakos, A. Leonard, J. H. Walther, Iterative brinkman penalization for remeshed vortex methods, Journal of Computational Physics 280 (2015) 547–562.

[41] X.-A. Han, Y. Ma, X. Huang, The cubic trigonometric Bézier curve with two shape parameters, Applied Mathematics Letters 22 (2) (2009) 226–231.

[42] C. De Boor, K. Höllig, M. Sabin, High accuracy geometric Hermite interpolation, Computer Aided Geometric Design 4 (4) (1987) 269–278.

[43] C. Lin, R. Chellappa, Classification of partial 2-D shapes using Fourier descriptors, IEEE Transactions on Pattern Analysis and Machine Intelligence (1987) 686–690.

[44] M. Juniper, Optimization with nonlinear adjoint looping, in: Int. Workshop on Nonnormal and Nonlinear Effects in Aero- and Thermoacoustics, Munich, Germany, 2010.

[45] D. D'Alessandro, M. Dahleh, I. Mezic, Control of mixing in fluid flow: a maximum entropy approach, IEEE Trans. Autom. Contr. 44 (10) (1999) 1852–1863.

[46] G. Mathew, I. Mezic, S. Grivopoulos, U. Vaidya, L. Petzold, Optimal control of mixing in Stokes fluid flows, Journal of Fluid Mechanics 580 (2007) 261–281.

[47] Z. Lin, J.-L. Thiffeault, C. Doering, Optimal stirring strategies for passive scalar mixing, Journal of Fluid Mechanics 675 (2011) 465–476.

[48] L. Vermach, C. Caulfield, Optimal mixing in three-dimensional plane Poiseuille flow at high Péclet number, Journal of Fluid Mechanics 850 (2018) 875–923.

[49] J. Ottino, Mixing, chaotic advection, and turbulence, Annu. Rev. Fluid Mech. 22 (1) (1990) 207–254.

[50] J.-L. Thiffeault, Using multiscale norms to quantify mixing and transport, Nonlinearity 25 (2) (2012) R1.

[51] J. Alexandersen, N. Aage, C. S. Andreasen, O. Sigmund, Topology optimisation for natural convection problems, International Journal for Numerical Methods in Fluids 76 (10) (2014) 699–721.

[52] G. Pingen, A. Evgrafov, K. Maute, Topology optimization of flow domains using the lattice boltzmann method, Structural and Multidisciplinary Optimization 34 (6) (2007) 507–524.

[53] C. Saglietti, P. Schlatter, A. Monokrousos, D. S. Henningson, Adjoint optimization of natural convection problems: differentially heated cavity, Theoretical and Computational Fluid Dynamics 31 (5-6) (2017) 537–553.

[54] C. Saglietti, P. Schlatter, E. Wadbro, M. Berggren, D. S. Henningson, Topology optimization of heat sinks in a square differentially heated cavity, International Journal of Heat and Fluid Flow 74 (2018) 36–52.

[55] R. Hemrajani, G. Tatterson, Mechanically Stirred Vessels, John Wiley & Sons, Inc., 2004, pp. 345–390.

[56] T. Engels, D. Kolomenskiy, K. Schneider, J. Sesterhenn, FluSI: A novel parallel simulation tool for flapping insect flight using a Fourier method with volume penalization, SIAM J. Sci. Comp. (2015) 1–21.

[57] K. Schneider, Numerical simulation of the transient flow behaviour in chemical reactors using a penalisation method, Comp. & Fluids 34 (10) (2005) 1223–1238.

[58] B. Kadoch, D. Kolomenskiy, P. Angot, K. Schneider, A volume penalization method for incompressible flows and scalar advection-diffusion with moving obstacles, Journal of Computational Physics. 231 (12) (2012) 4365–4383.

[59] D. Foures, C. Caulfield, P. Schmid, Localization of flow structures using ∞-norm optimization, Journal of Fluid Mechanics 729 (2013) 672–701.

[60] D. Foures, C. Caulfield, P. Schmid, Optimal mixing in two-dimensional plane Poiseuille flow at finite Péclet number, Journal of Fluid Mechanics 748 (2014) 241–277.

[61] M. Nemec, M. J. Aftosmis, Adjoint sensitivity computations for an embedded-boundary cartesian mesh method and CAD geometry, in: H. Deconinck, E. Dick (Eds.), Computational Fluid Dynamics 2006, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 531–536.

[62] R. Horn, C. Johnson, Matrix Analysis, 2nd Edition, Cambridge University Press, New York, NY, USA, 2012.

[63] D. Kolomenskiy, K. Schneider, A Fourier spectral method for the Navier-Stokes equations with volume penalization for moving solid obstacles, Journal of Computational Physics. 228 (16) (2009) 5687–5709.

[64] W. Liu, Mixing enhancement by optimal flow advection, SIAM Journal on Control and Optimization 47 (2) (2008) 624–638.

[65] D. Kolomenskiy, M. Maeda, T. Engels, H. Liu, K. Schneider, J.-C. Nave, Aerodynamic ground effect in fruitfly sized insect takeoff, PLOS ONE 11 (3) (2016) 1–21. `doi:10.1371/journal.pone.0152072`.

[66] S. A. Orszag, On the elimination of aliasing in finite-difference schemes by filtering high-wavenumber components, Journal of the Atmospheric sciences 28 (6) (1971) 1074–1074.

[67] T. Hou, R. Li, Computing nearly singular solutions using pseudo-spectral methods, Journal of Computational Physics. 226 (2007) 379–397.

[68] E. Hairer, S. Nørsett, G. Wanner, Solving Ordinary Differential Equations I Nonstiff problems, 2nd Edition, Springer, 2000.

[69] A. Griewank, A. Walther, Algorithm 799: Revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation, ACM Trans. Math. Softw. 26 (2000) 19–45.

[70] M. Eggl, P. Schmid, A gradient-based framework for maximizing mixing in binary fluids, Journal of Computational Physics 368 (2018) 131 – 153.

[71] T. R. Crimmins, A complete set of Fourier descriptors for two-dimensional shapes, IEEE Transactions on Systems, Man, and Cybernetics 12 (6) (1982) 848–855. doi:10.1109/TSMC.1982.4308918.

[72] R. A. Buckingham, The classical equation of state of gaseous helium, neon and argon, Proceedings of the Royal Society of London Series A 168 (1938) 264–283. doi:10.1098/rspa.1938.0173.

[73] D. Foures, C. Caulfield, P. Schmid, Variational framework for flow optimization using seminorm constraints, Physical Review E 86 (2) (2012) 026306.

[74] A. Beck, M. Teboulle, Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems, IEEE transactions on image processing 18 (11) (2009) 2419–2434.

[75] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.

[76] H. M. Cundy, A. P. Rollett, Mathematical models, Clarendon Press Oxford, 1961.

A

Arithmetic of Hadamard product

For the sake of clarity and completeness, we will give a brief summary of rules and relations involving the Hadamard product[62] which has been used in the formulation of the governing equations and the derivation of the adjoint equations, and has produced a compact formalism and notation. In particular, we will consider the steps involved in the transfer of operators acting on the direct flow variables (such as velocities, pressure or passive scalar) onto corresponding operators acting on their adjoint equivalents. While this transfer is rather straightforward in the matrix-product case, care has to be exercised when the operator involves Hadamard products.

The Hadamard product, denoted by $\circ$, of two vectors $\mathbf{a}$ and $\mathbf{b}$ of identical size is defined as

$$\mathbf{c} = \mathbf{a} \circ \mathbf{b} = \mathbf{b} \circ \mathbf{a} \qquad \text{with} \qquad c_i = a_i b_i, \tag{A.1}$$

where we stress that we do not use the Einstein summation convention here. It is defined as the element-wise product of two vectors and results in a vector $\mathbf{c}$ equal in size to $\mathbf{a}$ or $\mathbf{b}$.

It will be convenient to re-express the Hadamard product of two vectors in terms of a standard matrix product. To this end, we introduce the notation $\mathbf{a}^{(D)}$ to indicate a diagonal matrix with the elements of $\mathbf{a}$ along its main diagonal. With this notation we can restate the Hadamard product as

$$\mathbf{c} = \mathbf{a} \circ \mathbf{b} = \mathbf{a}^{(D)}\mathbf{b} = \mathbf{a}\mathbf{b}^{(D)}. \tag{A.2}$$

In the derivation of the adjoint equations, we are faced with terms of the form

$$\mathcal{I} = \mathbf{a}^H \left(\mathbf{b} \circ [\mathsf{M}\mathbf{c}]\right). \tag{A.3}$$

Expressions of this type constitute an inner product $\mathcal{I}$ and arise from the nonlinear terms of the governing equations related to convective transport, but terms linked to the mask function can also yield the above example. They appear in the augmented Lagrangian formulation of the optimisation problem. Due to our spatial discretisation, $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$ are column vectors of size $n \times 1$ with n as the total number of degrees of freedom, and M is an $n \times n$ matrix. During the derivation of the adjoint equations, the vector $\mathbf{c}$ may represent a first variation of a flow variable, while the vector $\mathbf{a}$ stands for the adjoint variable (see the main text for details). We seek to isolate this first variation (the vector $\mathbf{c}$) by transferring all operators acting on it onto the adjoint variable represented by $\mathbf{a}$ while preserving the inner product. Using the alternative formulation of the Hadamard product based on diagonal matrices, we obtain

$$\begin{aligned}
\mathbf{a}^H \left(\mathbf{b} \circ [\mathsf{M}\mathbf{c}]\right) &= \mathbf{a}^H \left(\mathbf{b}^{(D)}\mathsf{M}\mathbf{c}\right) && \text{(A.4a)}\\
&= \mathbf{a}^H \left(\left(\mathsf{M}^H\mathbf{b}^{(D)}\right)^H \mathbf{c}\right) && \text{(A.4b)}\\
&= \left(\mathsf{M}^H\mathbf{b}^{(D)}\mathbf{a}\right)^H \mathbf{c} && \text{(A.4c)}\\
&= \left(\mathsf{M}^H \left[\mathbf{b} \circ \mathbf{a}\right]\right)^H \mathbf{c}. && \text{(A.4d)}
\end{aligned}$$

Using this simple rule, we are able to efficiently manipulate most expressions in our adjoint derivations. We note that in the case of M being an identity matrix, our relation simply reduces to

$$\mathbf{a}^H (\mathbf{b} \circ \mathbf{c}) = (\mathbf{b} \circ \mathbf{a})^H \mathbf{c} = \left(\mathbf{a}^H \circ \mathbf{b}^H\right) \mathbf{c}. \tag{A.5}$$

B

Full adjoint derivation of non-penalised governing equations

In this appendix we present the detailed method behind the $\theta^\dagger$ derivation of the continuous adjoint of section 2.3.3, which arises from taking the first variation of the augmented Lagrangian $\mathcal{L}$ with respect to θ. This derivation begins as follows

$$\left\langle \frac{\partial \mathcal{L}}{\partial \theta}, \delta\theta \right\rangle = \left\langle \frac{\partial \mathcal{J}}{\partial \theta}, \delta\theta \right\rangle - \langle \theta^\dagger, \partial_t \delta\theta + \boldsymbol{u} \cdot \nabla \delta\theta - Pe^{-1} \nabla^2 \delta\theta \rangle. \tag{B.1}$$

In line with the previous derivation of $\boldsymbol{u}^\dagger$, we switch from vector to component notation and using the Einstein summation convention

$$= \left\langle \frac{\partial \mathcal{J}}{\partial \theta}, \delta\theta \right\rangle - \langle \theta^\dagger, \partial_t \delta\theta + u_i \partial_i \delta\theta - Pe^{-1} \partial_i \partial_i \delta\theta \rangle \tag{B.2}$$

$$= \alpha((-1)^\beta (\partial_i \partial_i)^{-\beta} \theta, \delta\theta)_{t=T} - \int_0^T \int_\Omega \theta^\dagger (\partial_t \delta\theta + u_i \partial_i \delta\theta - Pe^{-1} \partial_i \partial_i \delta\theta) \mathrm{d}\Omega \, \mathrm{d}t. \tag{B.3}$$

Next we apply integration by parts over the spatial and temporal integrals to isolate the $\delta\theta$ terms

$$\begin{aligned}
&= \alpha((-1)^{\beta}(\partial_i\partial_i)^{-\beta}\theta, \delta\theta)_{t=T} - (\theta^{\dagger}, \delta\theta)_{t=0}^{t=T} + \langle \partial_t\theta^{\dagger}, \delta\theta\rangle - [\theta^{\dagger}u_i n_i^R, \delta\theta]_{\delta R} - [\theta^{\dagger}u_i n_i^{r_1}, \delta\theta]_{\delta r_1} \\
&\quad - [\theta^{\dagger}u_i n_i^{r_2}, \delta\theta]_{\delta r_2} + \langle \partial_i(\theta^{\dagger}u_i), \delta\theta\rangle + [Pe^{-1}\theta^{\dagger}n_i^R, \partial_i\delta\theta]_{\delta R} + [Pe^{-1}\theta^{\dagger}n_i^{r_1}, \partial_i\delta\theta]_{\delta r_1} \\
&\quad + [Pe^{-1}\theta^{\dagger}n_i^{r_2}, \partial_i\delta\theta]_{\delta r_2} - [Pe^{-1}\partial_i\theta^{\dagger}n_i^R, \delta\theta]_{\delta R} - [Pe^{-1}\partial_i\theta^{\dagger}n_i^{r_1}, \delta\theta]_{\delta r_1} \\
&\quad - [Pe^{-1}\partial_i\theta^{\dagger}n_i^{r_2}, \delta\theta]_{\delta r_2} + \langle Pe^{-1}\partial_i\partial_i\theta^{\dagger}, \delta\theta\rangle. \qquad \text{(B.4)}
\end{aligned}$$

Collecting all the like terms so that they fall under the same inner product, we are left with

$$\begin{aligned}
&= -(\theta^{\dagger} - \alpha(-1)^{\beta}(\partial_i\partial_i)^{-\beta}\theta, \delta\theta)_{t=T} - (\theta^{\dagger}, \delta\theta)_{t=0} + \langle \partial_t\theta^{\dagger} + u_i\partial_i\theta^{\dagger} + Pe^{-1}\partial_i\partial_i\theta^{\dagger}, \delta\theta\rangle \\
&\quad - [\theta^{\dagger}u_i n_i^R + Pe^{-1}\partial_i\theta^{\dagger}n_i^R, \delta\theta]_{\delta R} - [\theta^{\dagger}u_i n_i^{r_1} + Pe^{-1}\partial_i\theta^{\dagger}n_i^{r_1}, \delta\theta]_{\delta r_1} \\
&\quad - [\theta^{\dagger}u_i n_i^{r_2} + Pe^{-1}\partial_i\theta^{\dagger}n_i^{r_2}, \delta\theta]_{\delta r_2} + [Pe^{-1}\theta^{\dagger}n_i^R, \partial_i\delta\theta]_{\delta R} + [Pe^{-1}\theta^{\dagger}n_i^{r_1}, \partial_i\delta\theta]_{\delta r_1} \\
&\quad + [Pe^{-1}\theta^{\dagger}n_i^{r_2}, \partial_i\delta\theta]_{\delta r_2}. \qquad \text{(B.5)}
\end{aligned}$$

The second term of equation (B.5) gives us the expressions which defines the temporal evolution of the adjoint scalar field

$$\partial_t\theta^{\dagger} + u_i\partial_i\theta^{\dagger} + Pe^{-1}\partial_i\partial_i\theta^{\dagger} = 0, \qquad \text{(B.6)}$$

with boundary conditions at R, r_1 and r_2 as follows

$$\nabla\theta^{\dagger}\cdot\boldsymbol{n} = \theta^{\dagger} = 0. \qquad \text{(B.7)}$$

We note that these boundary conditions mirror the no-outflow condition of the forward variables. The temporal conditions for $\theta^{\dagger}$ are seen to be

$$\theta^{\dagger}(\boldsymbol{x}, T) = -\left.\frac{\partial\mathcal{J}}{\partial\theta}\right|_T, \qquad \text{(B.8)}$$

$$\theta^{\dagger}(\boldsymbol{x}, 0) = 0. \qquad \text{(B.9)}$$

This result above, combined with the results obtained in section (2.3.3) lead to the full continuous analytic adjoint system.

C

Full adjoint derivation of penalised governing equations

This appendix aims to demonstrate the method for deriving the continuous adjoint in full. The mathematical background and general overview of this approach, as well as the derivation of one of the terms, are discussed in §2.3.4. To begin, we will explicitly derive the $(\star\star)$ and $(\star\star\star)$ terms from equation (2.54), followed by the summation of these terms into one expression. Once the derivation for $\boldsymbol{u}^\dagger$ is completed, we turn to obtaining the governing equations of the other adjoint terms to close the adjoint system.

Starting with $\boldsymbol{u}^\dagger$, we begin with $(\star\star)$, and in line with the derivation in main text, we aim to isolate δu. As a reminder to the reader, the expression we consider is

$$\begin{aligned}\int_0^T \left(\frac{\partial(\star\star)}{\partial \boldsymbol{u}_i}\right)\delta\boldsymbol{u}_i \,\mathrm{d}t = \int_0^T \frac{\partial}{\partial \boldsymbol{u}_i}\Bigg\{ p^{\dagger,H}\Bigg[&[\mathsf{A}_j\mathsf{A}_j]p + \mathsf{A}_k(\boldsymbol{u}_j \circ [\mathsf{A}_j\boldsymbol{u}_k]) \\ &+ \mathsf{A}_k\left[\frac{\chi}{C_n}\circ\boldsymbol{u}_k + \frac{\chi_i}{C_n}\circ(\boldsymbol{u}_{s,i})_k\right] \\ &- Re^{-1}\mathsf{A}_k[\mathsf{A}_i[F_{ij}(\boldsymbol{u})\circ(\mathsf{A}_j\boldsymbol{u}_k)]]\Bigg]\Bigg\}\delta\boldsymbol{u}_i \,\mathrm{d}t, \qquad \text{(C.1)}\end{aligned}$$

i.e., the first variation of the pressure equation with respect to $\boldsymbol{u}^\dagger$. Applying the derivative to the terms in the integral we arrive at

$$\begin{aligned}
= \int_0^T & p^{\dagger,H}\mathsf{A}_k(\delta\boldsymbol{u}_i \circ [\mathsf{A}_i\boldsymbol{u}_k]) + p^{\dagger,H}\mathsf{A}_i(\boldsymbol{u}_j \circ [\mathsf{A}_j\delta\boldsymbol{u}_i]) + p^{\dagger,H}\mathsf{A}_i\left[\frac{\chi}{C_\eta} \circ (\delta\boldsymbol{u}_i)\right] \\
& - p^{\dagger,H} Re^{-1}\mathsf{A}_i[\mathsf{A}_k[F_{kj}(\boldsymbol{u}) \circ (\mathsf{A}_j\delta\boldsymbol{u}_i)]] \\
& \quad - p^{\dagger,H} Re^{-1}\mathsf{A}_k\left[A_l\left[\left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial\boldsymbol{u}_i}\delta\boldsymbol{u}_i\right) \circ (\mathsf{A}_j\boldsymbol{u}_k)\right]\right]\ \mathrm{d}t.
\end{aligned} \tag{C.2}$$

Using the properties of the Hadamard product from appendix A and collecting like terms we obtain

$$\begin{aligned}
= \int_0^T & \left(\mathsf{A}_k^H p^\dagger \circ [\mathsf{A}_i\boldsymbol{u}_k]\right)^H \delta\boldsymbol{u}_i + (\mathsf{A}_j^H[\boldsymbol{u}_j \circ \mathsf{A}_i^H p^\dagger])^H\delta\boldsymbol{u}_i + \left[\frac{\chi}{C_\eta} \circ \mathsf{A}_i^H p^\dagger\right]^H \delta\boldsymbol{u}_i \\
& - Re^{-1}\left(\mathsf{A}_j^H(F_{kj}(\boldsymbol{u}) \circ \mathsf{A}_k^H\mathsf{A}_i^H p^\dagger)\right)^H \delta\boldsymbol{u}_i \\
& \quad - Re^{-1}\left(\left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial\boldsymbol{u}_i}\right)^H A_l^T\mathsf{A}_k^H p^\dagger \circ (\mathsf{A}_j\boldsymbol{u}_k)\right)^H \delta\boldsymbol{u}_i\ \mathrm{d}t
\end{aligned} \tag{C.3}$$

$$\begin{aligned}
= \int_0^T & \Bigg[\mathsf{A}_k^H p^\dagger \circ [\mathsf{A}_i\boldsymbol{u}_k] + \mathsf{A}_j^H[\boldsymbol{u}_j \circ \mathsf{A}_i^H p^\dagger] + \frac{\chi}{C_\eta} \circ \mathsf{A}_i^H p^\dagger - Re^{-1}\mathsf{A}_j^H(F_{kj}(\boldsymbol{u}) \circ \mathsf{A}_k^H\mathsf{A}_i^H p^\dagger) \\
& \quad - Re^{-1}\left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial\boldsymbol{u}_i}\right)^H A_l^T\mathsf{A}_k^H p^\dagger \circ (\mathsf{A}_j\boldsymbol{u}_k)\Bigg]^H \delta\boldsymbol{u}_i\ \mathrm{d}t,
\end{aligned} \tag{C.4}$$

leaving us with the full expression for $(\star\star)$. Turning to $(\star\star\star)$ and applying the same methodology, we obtain the following

$$\begin{aligned}
\int_0^T \left(\frac{\partial(\star\star\star)}{\partial\boldsymbol{u}_i}\right)\delta\boldsymbol{u}_i\ \mathrm{d}t &= \int_0^T \frac{\partial}{\partial\boldsymbol{u}_i}\Bigg\{\theta^{\dagger,H}\Bigg[\partial_t\theta + (\mathbf{1}-\chi) \circ \boldsymbol{u}_j \circ [\mathsf{A}_j\theta] + \chi_i \circ (\boldsymbol{u}_{s,i})_j \circ [\mathsf{A}_j\theta] \\
& \qquad - \mathsf{A}_i([Pe^{-1}(\mathbf{1}-\chi) + \kappa\chi] \circ \mathsf{A}_i\theta)\Bigg]\Bigg\}\delta\boldsymbol{u}_i\ \mathrm{d}t && \text{(C.5)} \\
&= \int_0^T \theta^{\dagger,H}\Big[(\mathbf{1}-\chi) \circ \delta\boldsymbol{u}_i \circ [\mathsf{A}_i\theta]\Big]\ \mathrm{d}t && \text{(C.6)} \\
&= \int_0^T \theta^{\dagger,H}\Big[(\mathbf{1}-\chi) \circ [\mathsf{A}_i\theta] \circ \delta\boldsymbol{u}_i\Big]\ \mathrm{d}t && \text{(C.7)} \\
&= \int_0^T \left[(\mathbf{1}-\chi) \circ \theta^\dagger \circ [\mathsf{A}_i\theta]\right]^H \delta\boldsymbol{u}_i\ \mathrm{d}t. && \text{(C.8)}
\end{aligned}$$

This term, as well as ($\star$) (from equation (2.60) and ($\star\star$) are then substituted into (2.56), and we arrive at the following full expression

$$\begin{aligned}\int_0^T \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}_i}\right) \delta \boldsymbol{u}_i \ \mathrm{d}t = \int_0^T \left(\frac{\partial \mathcal{J}}{\partial \boldsymbol{u}_i}\right) \delta \boldsymbol{u}_i \ \mathrm{d}t - [\boldsymbol{u}_i^\dagger \delta \boldsymbol{u}]_0^H - \int_0^T \Big[& -(\partial_t \boldsymbol{u}_i^\dagger) + (\boldsymbol{u}_k^\dagger \circ [\mathsf{A}_i \boldsymbol{u}_k]) \\ & + \mathsf{A}_j^H [\boldsymbol{u}_j \circ \boldsymbol{u}_i^\dagger] + \frac{\chi}{C_n} \circ (\boldsymbol{u}_i^\dagger) - Re^{-1} \mathsf{A}_j^H [F_{kj}(\boldsymbol{u}) \circ \mathsf{A}_k^H \boldsymbol{u}_i^\dagger] \\ & - Re^{-1} \left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial \boldsymbol{u}_i}\right)^H (A_l^T \boldsymbol{u}_k^\dagger \circ (\mathsf{A}_j \boldsymbol{u}_k))\Big]^H \delta \boldsymbol{u}_i \ \mathrm{d}t \\ & - \int_0^T \Big[\mathsf{A}_k^H p^\dagger \circ [\mathsf{A}_i \boldsymbol{u}_k] + \mathsf{A}_j^H [\boldsymbol{u}_j \circ \mathsf{A}_i^H p^\dagger] + \frac{\chi}{C_\eta} \circ \mathsf{A}_i^H p^\dagger \\ & - Re^{-1} \mathsf{A}_j^H (F_{kj}(\boldsymbol{u}) \circ \mathsf{A}_k^H \mathsf{A}_i^H p^\dagger) \\ & - Re^{-1} \left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial \boldsymbol{u}_i}\right)^H A_l^T \mathsf{A}_k^H p^\dagger \circ (\mathsf{A}_j \boldsymbol{u}_k)\Big]^H \delta \boldsymbol{u}_i \ \mathrm{d}t \\ & - \int_0^T \big[(\mathbf{1} - \chi) \circ \theta^\dagger \circ [\mathsf{A}_i \theta]\big]^H \delta \boldsymbol{u}_i \ \mathrm{d}t. \end{aligned} \tag{C.9}$$

We proceed by collecting all quantities so that they fall under their respective time domains. With further simplification, we obtain

$$\begin{aligned} = -[\boldsymbol{u}_i^\dagger \delta \boldsymbol{u}]_0^H - \int_0^T \Bigg[& - \left(\frac{\partial \mathcal{J}}{\partial \boldsymbol{u}_i}\right)^H - (\partial_t \boldsymbol{u}_i^\dagger) + (\boldsymbol{u}_k^\dagger \circ [\mathsf{A}_i \boldsymbol{u}_k]) + \mathsf{A}_j^H [\boldsymbol{u}_j \circ \boldsymbol{u}^\dagger] \\ & + \frac{\chi}{C_n} \circ (\boldsymbol{u}_i^\dagger) - Re^{-1} \mathsf{A}_j^H [F_{kj}(\boldsymbol{u}) \circ \mathsf{A}_k^H \boldsymbol{u}_i^\dagger] - Re^{-1} \left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial \boldsymbol{u}_i}\right)^H (A_l^T \boldsymbol{u}_k^\dagger \circ (\mathsf{A}_j \boldsymbol{u}_k)) \\ & + \mathsf{A}_k^H p^\dagger \circ [\mathsf{A}_i \boldsymbol{u}_k] + [\boldsymbol{u}_j \circ \mathsf{A}_j]^H \mathsf{A}_i^H p^\dagger + \frac{\chi}{C_\eta} \circ \mathsf{A}_i^H p^\dagger - Re^{-1} \mathsf{A}_j^H (F_{kj}(\boldsymbol{u}) \circ \mathsf{A}_k^H \mathsf{A}_i^H p^\dagger) \\ & - Re^{-1} \left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial \boldsymbol{u}_i}\right)^H A_l^T \mathsf{A}_k^H p^\dagger \circ (\mathsf{A}_j \boldsymbol{u}_k) + (\mathbf{1} - \chi) \circ \theta^\dagger \circ [\mathsf{A}_i \theta]\Bigg]^H \delta \boldsymbol{u}_i \ \mathrm{d}t. \end{aligned} \tag{C.10}$$

As we enforce $\partial \mathcal{L} = 0$, we require the above terms to be zero and thus, we arrive at the expression for $\boldsymbol{u}^\dagger$ presented in section §2.3.4.

Next, we turn to the derivation of the adjoint pressure equation, beginning with the first variation of the augmented Lagrangian with respect to p. We define the same replacement terms ($\star$), ($\star\star$) and ($\star\star\star$) as before to retain a simple analytic expression, i.e.,

$$\begin{aligned}\int_0^T \left(\frac{\partial \mathcal{L}}{\partial p}\right) \delta p \ \mathrm{d}t = \int_0^T \left(\frac{\partial \mathcal{J}}{\partial p}\right) \delta p \ \mathrm{d}t - \int_0^T \left(\frac{\partial (\star)}{\partial p}\right) \delta p \ \mathrm{d}t \\ - \int_0^T \left(\frac{\partial (\star\star)}{\partial p}\right) \delta p \ \mathrm{d}t - \int_0^T \left(\frac{\partial (\star\star\star)}{\partial p}\right) \delta p \ \mathrm{d}t. \qquad \text{(C.11)}\end{aligned}$$

Beginning with ($\star$), we note that the only term that depends on p is the pressure derivative

$$\begin{aligned}\int_0^T \left(\frac{\partial (\star)}{\partial p}\right) \delta p \ \mathrm{d}t &= \int_0^T (\boldsymbol{u}_k^\dagger) \frac{\partial f_k(\boldsymbol{u})}{\partial p} \delta p \ \mathrm{d}t & \text{(C.12)} \\ &= \int_0^T (\boldsymbol{u}_k^\dagger)^H \mathsf{A}_k \delta p \ \mathrm{d}t & \text{(C.13)} \\ &= \int_0^T \left[\mathsf{A}_k^H \boldsymbol{u}_k^\dagger\right]^H \delta p \ \mathrm{d}t. & \text{(C.14)}\end{aligned}$$

Turning to ($\star\star$), we observe a similar pattern

$$\begin{aligned}\int_0^T \left(\frac{\partial (\star\star)}{\partial p}\right) \delta p \ \mathrm{d}t &= \int_0^T \frac{\partial}{\partial p} \Bigg\{ p^{\dagger,H} \Bigg[[\mathsf{A}_j \mathsf{A}_j] p + \mathsf{A}_k (\boldsymbol{u}_j \circ [\mathsf{A}_j \boldsymbol{u}_k]) \\ &\qquad + \mathsf{A}_k \left[\frac{\chi}{C_n} \circ \boldsymbol{u} + \frac{\chi_i}{C_n} \circ \boldsymbol{u}_{s,i} \right] \Bigg] \Bigg\} \delta p \ \mathrm{d}t & \text{(C.15)} \\ &= \int_0^T p^{\dagger,H} [\mathsf{A}_j \mathsf{A}_j] \delta p \ \mathrm{d}t & \text{(C.16)} \\ &= \int_0^T \left[\mathsf{A}_j^H \mathsf{A}_j^H p^\dagger\right]^H \delta p \ \mathrm{d}t. & \text{(C.17)}\end{aligned}$$

The passive scalar has no pressure dependency and so ($\star\star\star$) makes no contribution to the derivation. Inserting these findings into equation (C.11), we are finally left with

$$\begin{aligned}\int_0^T \left(\frac{\partial \mathcal{L}}{\partial p}\right) \delta p \ \mathrm{d}t = \int_0^T \left(\frac{\partial \mathcal{J}}{\partial p}\right) \delta p \ \mathrm{d}t - \int_0^T \left[\mathsf{A}_k^H \boldsymbol{u}_k^\dagger\right]^H \delta p \ \mathrm{d}t \\ - \int_0^T \left[\mathsf{A}_j^H \mathsf{A}_j^H p^\dagger\right]^H \delta p \ \mathrm{d}t \qquad \text{(C.18)} \\ = \int_0^T \left[\left(\frac{\partial \mathcal{J}}{\partial p}\right)^H - \mathsf{A}_k^H \boldsymbol{u}_k^\dagger - \mathsf{A}_j^H \mathsf{A}_j^H p^\dagger\right]^H \delta p \ \mathrm{d}t. \qquad \text{(C.19)}\end{aligned}$$

Enforcing $\partial \mathcal{L} = 0$, we obtain the evolution equation for $p^\dagger$

$$\mathsf{A}_j^H \mathsf{A}_j^H p^\dagger + \mathsf{A}_k^H \boldsymbol{u}_k^\dagger - \left(\frac{\partial \mathcal{J}}{\partial p}\right)^H = 0. \qquad \text{(C.20)}$$

To conclude the adjoint derivation we now turn to the first variation with respect to θ, which allows us to derive the governing equation for $\theta^\dagger$. We begin with

$$\begin{aligned}\int_0^T \left(\frac{\partial \mathcal{L}}{\partial \theta}\right) \delta \theta \ \mathrm{d}t = \int_0^T \left(\frac{\partial \mathcal{J}}{\partial \theta}\right) \delta \theta \ \mathrm{d}t - \int_0^T \left(\frac{\partial (\star)}{\partial \theta}\right) \delta \theta \ \mathrm{d}t \\ - \int_0^T \left(\frac{\partial (\star\star)}{\partial \theta}\right) \delta \theta \ \mathrm{d}t - \int_0^T \left(\frac{\partial (\star\star\star)}{\partial \theta}\right) \delta \theta \ \mathrm{d}t. \qquad \text{(C.21)}\end{aligned}$$

Only $(\star\star\star)$ depends on θ and, thus, all other terms do not contribute to the final expression. Therefore we only need to consider

$$\begin{aligned}\int_0^T \frac{\partial (\star\star\star)}{\partial \theta} \delta \theta \ \mathrm{d}t = \int_0^T \frac{\partial}{\partial \theta}\Bigg\{\theta^{\dagger,H}\Big[\partial_t \theta + (\mathbf{1} - \chi) \circ \boldsymbol{u}_j \circ [\mathsf{A}_j \theta] + \chi_i \circ (\boldsymbol{u}_{s,i})_j \circ [\mathsf{A}_j \theta] \\ - \mathsf{A}_i([Pe^{-1}(\mathbf{1} - \chi) + \kappa \chi] \circ \mathsf{A}_i \theta)\Big]\Bigg\} \delta \theta \ \mathrm{d}t. \qquad \text{(C.22)}\end{aligned}$$

Employing the rules of the Hadamard product, we isolate the $\partial \theta$ terms and then simplify to obtain~~.~~

$$
\begin{aligned}
&= [\theta^{\dagger,H}\delta\theta]_0^H + \int_0^T \Big[-\partial_t\theta^\dagger + \mathsf{A}_j^H[(\mathbf{1}-\chi)\circ \boldsymbol{u}_j \circ \theta^\dagger] \\
&\quad + \mathsf{A}_j^H[\chi_i(\boldsymbol{u}_{s,i})_j\circ\theta^\dagger] - \mathsf{A}_i^H([Pe^{-1}(\mathbf{1}-\chi)+\kappa\chi]\circ \mathsf{A}_i^H\theta^\dagger)\Big]^H \delta\theta \ \mathrm{d}t.
\end{aligned} \tag{C.23}
$$

Substituting this expression into (C.21), and then gathering all like terms into their respective time domains, we arrive at

$$
\int_0^T \left(\frac{\partial\mathcal{L}}{\partial\theta}\right)\delta\theta \ \mathrm{d}t = \int_0^T \left(\frac{\partial\mathcal{J}}{\partial\theta}\right)\delta\theta \ \mathrm{d}t - \int_0^T \left(\frac{\partial(\star\star\star)}{\partial\theta}\right)\delta\theta \ \mathrm{d}t \tag{C.24}
$$

$$
\begin{aligned}
&= \int_0^T \left(\frac{\partial\mathcal{J}}{\partial\theta}\right)\delta\theta \ \mathrm{d}t - [\theta^\dagger\delta\theta]_0^H - \int_0^T \Big[-\partial_t\theta^\dagger + \mathsf{A}_j^H[(\mathbf{1}-\chi)\circ \boldsymbol{u}_j \circ \theta^\dagger] \\
&\quad + \mathsf{A}_j^H[\chi_i\circ(\boldsymbol{u}_{s,i})_j\circ\theta^\dagger] - \mathsf{A}_i^H([Pe^{-1}(\mathbf{1}-\chi)+\kappa\chi]\circ \mathsf{A}_i^H\theta^\dagger)\Big]^H \delta\theta \ \mathrm{d}t
\end{aligned} \tag{C.25}
$$

$$
\begin{aligned}
&= -[\theta^{\dagger,H}\delta\theta]_0^H + \int_0^T \Bigg[\left(\frac{\partial\mathcal{J}}{\partial\theta}\right)^H + \partial_t\theta^\dagger - \mathsf{A}_j^H[(\mathbf{1}-\chi)\circ \boldsymbol{u}_j \circ \theta^\dagger] \\
&\quad - \mathsf{A}_j^H[\chi_i\circ(\boldsymbol{u}_{s,i})_j\circ\theta^\dagger] \\
&\quad + \mathsf{A}_i^H([Pe^{-1}(\mathbf{1}-\chi)+\kappa\chi]\circ \mathsf{A}_i^H\theta^\dagger)\Bigg]^H \delta\theta \ \mathrm{d}t.
\end{aligned} \tag{C.26}
$$

Exploiting the extremality of $\mathcal{L}$, which implies these terms are equal to zero, we make the following conclusions: the final time condition for $\theta^\dagger$ becomes

$$
\theta^\dagger|_T = \left.\frac{\partial\mathcal{J}}{\partial\theta}\right|_T, \tag{C.27}
$$

and the evolution equation is

$$
\begin{aligned}
&\partial_t\theta^\dagger - \mathsf{A}_j^H[(\mathbf{1}-\chi)\circ \boldsymbol{u}_j \circ \theta^\dagger] - \mathsf{A}_j^H[\chi_i\circ(\boldsymbol{u}_{s,i})_j \ \circ\theta^\dagger] + \mathsf{A}_i^H([Pe^{-1}(\mathbf{1}-\chi) \\
&\qquad +\kappa\chi]\circ \mathsf{A}_i^H\theta^\dagger) + \left(\frac{\partial\mathcal{J}}{\partial\theta}\right)^H = 0.
\end{aligned} \tag{C.28}
$$

The final pieces of the adjoint system are the optimality conditions with respect to the optimisation variables, i.e., the first variations of $\mathcal{L}$ with respect to χ_i, u_{C_i} and ω_i. We begin with the optimisation condition of the mask, $\chi_i^\dagger$. We note that the starred terms remain, but in this instance we also include the condition for the mask. This implies the following starting point

$$\begin{aligned}\int_0^T \left(\frac{\partial \mathcal{L}}{\partial \chi_i}\right) \delta\chi_i \,\mathrm{d}t = \int_0^T \left(\frac{\partial \mathcal{J}}{\partial \chi_i}\right) \delta\chi_i \,\mathrm{d}t - \int_0^T \left(\frac{\partial (\star)}{\partial \chi_i}\right) \delta\chi_i \,\mathrm{d}t \\ - \int_0^T \left(\frac{\partial (\star\star)}{\partial \chi_i}\right) \delta\chi_i \,\mathrm{d}t - \int_0^T \left(\frac{\partial (\star\star\star)}{\partial \chi_i}\right) \delta\chi_i \,\mathrm{d}t \\ - \int_0^T \chi_i^{\dagger,T} \delta\chi_i \,\mathrm{d}t. \qquad \text{(C.29)}\end{aligned}$$

Applying the derivative to each individual term, and then simplifying according to the properties of the Hadamard product we obtain

$$\begin{aligned}&= \int_0^T \left(\frac{\partial \mathcal{J}}{\partial \chi_i}\right) \delta\chi_i \,\mathrm{d}t - \int_0^T \boldsymbol{u}_j^{\dagger,T} \frac{\delta\chi_i}{C_n} \circ \boldsymbol{u}_j - \boldsymbol{u}_j^{\dagger,T} \frac{\delta\chi_i}{C_n} \circ (\boldsymbol{u}_{s,i})_j \,\mathrm{d}t \\ &\quad - \int_0^T p^{\dagger,H} \mathsf{A}_j \left[\frac{\delta\chi_i}{C_n} \circ \boldsymbol{u}_j - \frac{\delta\chi_i}{C_n} \circ (\boldsymbol{u}_{s,i})_j\right] \,\mathrm{d}t \\ &\qquad - \int_0^T \theta^{\dagger,H} \Big[-\delta\chi_i \circ \boldsymbol{u}_j \circ [\mathsf{A}_j\theta] + \delta\chi_i \circ (\boldsymbol{u}_{s,i})_j \circ [\mathsf{A}_j\theta] \\ &\qquad\quad - \mathsf{A}_j([Pe^{-1}(-\delta\chi_i) + \kappa\delta\chi_i] \circ \mathsf{A}_j\theta) \Big] \mathrm{d}t - \int_0^T \chi_i^{\dagger,T} \delta\chi_i \,\mathrm{d}t \qquad \text{(C.30)} \\ &= \int_0^T \left(\frac{\partial \mathcal{J}}{\partial \chi_i}\right) \delta\chi \,\mathrm{d}t - \int_0^T \frac{(\boldsymbol{u}_j^\dagger \circ \boldsymbol{u}_j - \boldsymbol{u}_j^\dagger \circ (\boldsymbol{u}_{s,i})_j)^H}{C_n} \delta\chi \,\mathrm{d}t \\ &\quad - \int_0^T \left[\frac{\left([\mathsf{A}_j^H p^\dagger] \circ \boldsymbol{u}_j - [\mathsf{A}_j^H p^\dagger] \circ (\boldsymbol{u}_{s,i})_j\right)^H}{C_\eta}\right] \delta\chi \,\mathrm{d}t \\ &\qquad - \int_0^T \Big[- \left(\theta^\dagger \circ (\boldsymbol{u}_j \circ [\mathsf{A}_j\theta])\right)^H \delta\chi + \left(\theta^\dagger \circ (\boldsymbol{u}_{s,i})_j \circ [\mathsf{A}_j\theta]\right)^H \delta\chi_i \\ &\qquad\quad - (\kappa - Pe^{-1})[\mathsf{A}_i^H \theta^\dagger \circ \mathsf{A}_i\theta]^H \delta\chi_i \Big] \mathrm{d}t - \int_0^T \chi_i^{\dagger,T} \delta\chi_i \,\mathrm{d}t. \qquad \text{(C.31)}\end{aligned}$$

Once again we collect all terms that fall under the same time interval to form

$$
\begin{aligned}
= \int_0^T \Bigg[\left(\frac{\partial \mathcal{J}}{\partial \chi_i} \right)^H & - \frac{\boldsymbol{u}_j^\dagger \circ (\boldsymbol{u}_j - (\boldsymbol{u}_{s,i})_j) + [\mathsf{A}_j^H p^\dagger] \circ (\boldsymbol{u}_j - (\boldsymbol{u}_{s,i})_j)}{C_\eta} \\
& + (\theta^\dagger \circ [\mathsf{A}_j \theta]) \circ (\boldsymbol{u}_j - (\boldsymbol{u}_{s,i})_j) \\
& \quad + (\kappa - Pe^{-1})[\mathsf{A}_j^H \theta^\dagger \circ \mathsf{A}_j \theta] - \chi^\dagger \Bigg] \delta \chi_i \, \mathrm{d}t. \qquad \text{(C.32)}
\end{aligned}
$$

Setting this to zero we obtain the following optimality condition for $\chi^\dagger$

$$
\begin{aligned}
\chi_i^\dagger = \left(\frac{\partial \mathcal{J}}{\partial \chi_i} \right)^H & - \frac{\boldsymbol{u}_j^\dagger \circ (\boldsymbol{u}_j - (\boldsymbol{u}_{s,i})_j) + [\mathsf{A}_j^H p^\dagger] \circ (\boldsymbol{u}_j - (\boldsymbol{u}_{s,i})_j)}{C_\eta} \\
& + (\theta^\dagger \circ [\mathsf{A}_j \theta]) \circ (\boldsymbol{u}_j - (\boldsymbol{u}_{s,i})_j) + (\kappa - Pe^{-1}) \mathsf{A}_j^H \theta^\dagger \circ \mathsf{A}_j \theta. \qquad \text{(C.33)}
\end{aligned}
$$

We can rearrange this equation, and introduce the quantity $\Pi_i^\dagger = \boldsymbol{u}^\dagger + \mathsf{A}_i^H p^\dagger$, to simplify this further

$$
= \left(\frac{\partial \mathcal{J}}{\partial \chi_i} \right)^H + \left[\theta^\dagger \circ [\mathsf{A}_j \theta] - \frac{\Pi_j^\dagger}{C_\eta} \right] \circ (\boldsymbol{u}_j - (\boldsymbol{u}_{s,i})_j) + (\kappa - Pe^{-1}) \mathsf{A}_j^H \theta^\dagger \circ \mathsf{A}_j \theta. \tag{C.34}
$$

The $\boldsymbol{u}_{C_i}$ and ω_i derivations do not differ significantly from the above approach, as the dependencies of the augmented Lagrangian, $\mathcal{L}$, on these control parameters are almost identical. The method in obtaining the optimality conditions also follows in the same fashion. Thus, the full explicit derivation is omitted here and only the full results presented in §2.3.4.

This concludes the full derivation of the penalisation-first, adjoint-second approach.

D

Backward time derivation

The last step in the numerical implementation of the adjoint is deriving the backward time stepping approach. This derivation is relatively straightforward and primarily done to gain understanding for the implementation. It is presented here for completeness sake.

Following a similar method as in Schneider *et al.*[57], we begin by deriving a time-stepping scheme with constant step size, i.e., $t_i - t_{i-1} = \Delta t$ for all i. We commence by rewriting the evolution equation for $\boldsymbol{u}^\dagger$ (2.73) as follows

$$\begin{aligned}\partial_t \boldsymbol{u}_i^\dagger = \Pi_k^\dagger \circ [\mathsf{A}_i \boldsymbol{u}_k] + \mathsf{A}_j^H [\boldsymbol{u}_j \circ \Pi_i^\dagger] + \frac{\chi}{C_n} \circ \Pi_i^\dagger - Re^{-1} \mathsf{A}_j^H (F_{kj}(\boldsymbol{u}) \circ \mathsf{A}_k^H \boldsymbol{u}_i^\dagger) \\ - Re^{-1} \left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial \boldsymbol{u}_i} \right)^H (\mathsf{A}_l^H \boldsymbol{u}_k^\dagger \circ (\mathsf{A}_j \boldsymbol{u}_k)) + (\mathbf{1} - \chi) \circ \theta^\dagger \circ [\mathsf{A}_i \theta] - \left(\frac{\partial \mathcal{J}}{\partial \boldsymbol{u}_i} \right)^H,\end{aligned} \tag{D.1}$$

where we define the right hand side as $\mathrm{RHS}(t)$. Performing an integration over time we obtain

$$\boldsymbol{u}_{t_n}^\dagger - \boldsymbol{u}_{t_{n+1}}^\dagger = \int_{t_{n+1}}^{t_n} \mathrm{RHS}(s)\ \mathrm{d}s. \tag{D.2}$$

Using Lagrange polynomials we can then interpolate RHS at the points t_{n+1} and

t_{n+2}

$$p(s) = \frac{s - t_{n+1}}{\Delta t}\mathrm{RHS}(t_{n+2}) - \frac{s - t_{n+2}}{\Delta t}\mathrm{RHS}(t_{n+1}), \tag{D.3}$$

where $\Delta t = t_{n+2} - t_{n+1}$. Therefore, we now rewrite the above as

$$\boldsymbol{u}^{\dagger}_{t_n} - \boldsymbol{u}^{\dagger}_{t_{n+1}} \approx \int_{t_{n+1}}^{t_n} p(s)\ \mathrm{d}s \tag{D.4}$$

$$= \int_{t_{n+1}}^{t_n} \frac{s - t_{n+1}}{\Delta t}\mathrm{RHS}(t_{n+2}) - \frac{s - t_{n+2}}{\Delta t}\mathrm{RHS}(t_{n+1})\ \mathrm{d}s. \tag{D.5}$$

Evaluating the above integral and then rearranging the resulting expression, we arrive at an update condition for $\boldsymbol{u}^{\dagger}_{t_n}$

$$\boldsymbol{u}^{\dagger}_{t_n} = \boldsymbol{u}^{\dagger}_{t_{n+1}} + \frac{\Delta t}{2}\mathrm{RHS}(t_{n+2}) - \frac{3\Delta t}{2}\mathrm{RHS}(t_{n+1}). \tag{D.6}$$

We stress that the above only holds true for a constant step-size. However, in our simulations we use variable step-sizes for time. Therefore, it is necessary to derive a suitable time-stepping scheme for this scenario. We begin by defining our variable time interval as $\Delta t_i = t_i - t_{i-1}$. Then, starting with the same integration over time and applying Lagrange polynomials as done previously we obtain

$$\boldsymbol{u}^{\dagger}_{t_n} - \boldsymbol{u}^{\dagger}_{t_{n+1}} \approx \int_{t_{n+1}}^{t_n} p(s)\ \mathrm{d}s \tag{D.7}$$

$$= \int_{t_{n+1}}^{t_n} \frac{s - t_{n+1}}{\Delta t_{n+2}}\mathrm{RHS}(t_{n+2}) - \frac{s - t_{n+2}}{\Delta t_{n+2}}\mathrm{RHS}(t_{n+1})\ \mathrm{d}s. \tag{D.8}$$

Integrating the above explicitly and inserting the values at the limits we obtain

$$= \left[\frac{(t_n - t_{n+1})^2}{2\Delta t_{n+2}}\mathrm{RHS}(t_{n+2})\right] - \left[\frac{(t_n - t_{n+2})^2 - (t_{n+1} - t_{n+2})^2}{2\Delta t_{n+2}}\mathrm{RHS}(t_{n+1})\right]. \tag{D.9}$$

To make the numerical implementation more palatable, and in line with relevant literature, we simplify the above to obtain the final form of the update step, which mirrors the form of the forward AB2 scheme

$$\boldsymbol{u}^\dagger_{t_n} = \boldsymbol{u}^\dagger_{t_{n+1}} + \frac{\Delta t^2_{n+1}}{2\Delta t_{n+2}}\mathrm{RHS}(t_{n+2}) - \frac{\Delta t_{n+1}}{2\Delta t_{n+2}}[(\Delta t_{n+1} + 2\Delta t_{n+2}]\mathrm{RHS}(t_{n+1}). \quad \text{(D.10)}$$

Equation (D.10) requires the previous values of $\boldsymbol{u}^\dagger$ and the RHS to be known to perform the update. Therefore, to initialise this scheme, the 'start up' is performed using a backwards Euler pre-step of the form

$$\boldsymbol{u}^\dagger_{t_n} = \boldsymbol{u}^\dagger_{t_{n+1}} - \Delta t_{n+1}\mathrm{RHS}(t_{n+1}). \quad \text{(D.11)}$$

This completes the derivation of the backward AB2 scheme. However, we make one caveat; the above derivation requires the computation of the terms of RHS, e.g.,

$$-Re^{-1}\mathsf{A}^H_j(F_{kj}(\boldsymbol{u}) \circ \mathsf{A}^H_k\boldsymbol{u}^\dagger_i) - Re^{-1}\left(\frac{\partial F_{lj}(\boldsymbol{u})}{\partial \boldsymbol{u}_i}\right)^H (\mathsf{A}^H_l\boldsymbol{u}^\dagger_k \circ (\mathsf{A}_j\boldsymbol{u}_k)), \quad \text{(D.12)}$$

which is computationally expensive. These terms, and particularly the non-Newtonian diffusion term of $\theta^\dagger$, cannot be avoided and thus non-negligible computational resources will need to be expended here.

However, we note that if $F(\boldsymbol{u})$ is a constant, i.e., we consider Newtonian fluids, this expensive computation can be avoided in the following fashion: consider equation (2.73) with $F(\boldsymbol{u})$ chosen to be 1 and thus written as

$$\begin{aligned}\partial_t\boldsymbol{u}^\dagger_i + Re^{-1}\mathsf{A}^H_j\mathsf{A}^H_j\boldsymbol{u}^\dagger_i = \Pi^\dagger_k \circ [\mathsf{A}_i\boldsymbol{u}_k] + \mathsf{A}^H_j[\boldsymbol{u}_j \circ \Pi^\dagger_i] + \frac{\chi}{C_n} \circ \Pi^\dagger_i + (\mathbf{1} - \chi) \circ \theta^\dagger \circ [\mathsf{A}_i\theta] \\ - \left(\frac{\partial \mathcal{J}}{\partial \boldsymbol{u}_i}\right)^H. \quad \text{(D.13)}\end{aligned}$$

We define $\mathrm{RHS}(\boldsymbol{u}^\dagger)$ as the right hand side of equation (D.13). Considering the homogeneous solution, (i.e., $\mathrm{RHS}(\boldsymbol{u}^\dagger) = 0$) we obtain the exact solution given by

$$\boldsymbol{u}^\dagger(t) = \boldsymbol{u}^\dagger_T \exp(Re^{-1}(T - t)\mathsf{A}_i\mathsf{A}_i), \quad \text{(D.14)}$$

where $\mathsf{A}_i\mathsf{A}_i$ are the matrices that represent the spatial differentiation. By simple manipulation it can be seen that

$$\boldsymbol{u}^{\dagger}_{\boldsymbol{t_k}} = \boldsymbol{u}^{\dagger}_{\boldsymbol{t_{k+1}}} \exp(Re^{-1}\Delta t_{k+1}\mathsf{A}_i\mathsf{A}_i), \tag{D.15}$$

where $\Delta t_{k+1} = t_{k+1} - t_k$. Next, we transform this into Fourier space and note that the $\mathsf{A}_i\mathsf{A}_i$ operator converts to the Fourier wavenumbers, i.e., $-|k|^2$. Therefore, we are left with an expression of the form

$$\hat{\boldsymbol{u}}^{\dagger}_{\boldsymbol{t_k}} = \hat{\boldsymbol{u}}^{\dagger}_{\boldsymbol{t_{k+1}}} \exp(Re^{-1}\Delta t_{k+1}|k|^2). \tag{D.16}$$

We now extend this to the inhomogeneous case

$$\partial_t \boldsymbol{u}^{\dagger} + Re^{-1}\mathsf{A}_i\mathsf{A}_i\boldsymbol{u}^{\dagger} = \mathrm{RHS}(\boldsymbol{u}^{\dagger}). \tag{D.17}$$

We observe that this can be manipulated to yield

$$\exp(-Re^{-1}t\mathsf{A}_i\mathsf{A}_i)\partial_t \left[\exp(Re^{-1}t\mathsf{A}_i\mathsf{A}_i)u^{\dagger}\right] = \mathrm{RHS}(\boldsymbol{u}^{\dagger}). \tag{D.18}$$

Rearranging and then integrating over time we obtain

$$\exp(Re^{-1}t\mathsf{A}_i\mathsf{A}_i)u^{\dagger}_t = \exp(Re^{-1}T\mathsf{A}_i\mathsf{A}_i)u^{\dagger}_T - \int_t^T \exp(Re^{-1}s\mathsf{A}_i\mathsf{A}_i)\mathrm{RHS}(\boldsymbol{u}^{\dagger}(s))\ \mathrm{d}s. \tag{D.19}$$

After dividing both sides by the exponential on the left-hand side we arrive at

$$u(t)^{\dagger} = \exp(Re^{-1}(T-t)\mathsf{A}_i\mathsf{A}_i)u^{\dagger}_T - \int_t^T \exp(Re^{-1}(s-t)\mathsf{A}_i\mathsf{A}_i)\mathrm{RHS}(\boldsymbol{u}^{\dagger}(s))\ \mathrm{d}s. \tag{D.20}$$

As the limits of the integration are, in essence, dummy variables we can rewrite equation (D.20) as the following

$$u^{\dagger}_{t_k} = \exp(Re^{-1}\Delta t_{k+1}\mathsf{A}_i\mathsf{A}_i)u^{\dagger}_{t_{k+1}} + \int_{t_{k+1}}^{t_k} \underbrace{\exp(Re^{-1}(s-t_k)\mathsf{A}_i\mathsf{A}_i)\mathrm{RHS}(\boldsymbol{u}^{\dagger}(s))}_{g(s)}\ \mathrm{d}s. \tag{D.21}$$

Using the Lagrangian polynomial approach of the constant time-step derivation, we can evaluate $g(s)$ in the following manner

$$\begin{aligned}\int_{t_{k+1}}^{t_k} g(s)\ \mathrm{d}s &\approx \frac{\Delta t_{k+1}^2}{2\Delta t_{k+2}} g(t_{k+2}) - \frac{\Delta t_{k+1}}{2\Delta t_{k+2}}[(\Delta t_{k+1} + 2\Delta t_{k+2}] g(t_{k+1}) \qquad \text{(D.22)} \\ &= \Bigg[\frac{\Delta t_{k+1}^2}{2\Delta t_{k+2}} \exp(Re^{-1}(\Delta t_{k+2})\mathsf{A}_i\mathsf{A}_i)\mathrm{RHS}(\boldsymbol{u}^\dagger(t_{k+2})) \\ &\quad - \frac{\Delta t_{k+1}}{2\Delta t_{k+2}}[(\Delta t_{k+1} + 2\Delta t_{k+2}]\mathrm{RHS}(\boldsymbol{u}^\dagger(t_{k+1}))\Bigg] \exp(Re^{-1}(\Delta t_{k+1})\mathsf{A}_i\mathsf{A}_i). \\ & \qquad \text{(D.23)}\end{aligned}$$

Inserting this value into equation (D.21) we arrive at

$$\begin{aligned}u_{t_k}^\dagger &= \exp(Re^{-1}\Delta t_{k+1}\mathsf{A}_i\mathsf{A}_i) u_{t_{k+1}}^\dagger + \Bigg[\frac{\Delta t_{k+1}^2}{2\Delta t_{k+2}} \exp(Re^{-1}\Delta t_{k+2}\mathsf{A}_i\mathsf{A}_i)\mathrm{RHS}(\boldsymbol{u}^\dagger(t_{k+2})) \\ &\quad - \frac{\Delta t_{k+1}}{2\Delta t_{k+2}}[(\Delta t_{k+1} + 2\Delta t_{k+2}]\mathrm{RHS}(\boldsymbol{u}^\dagger(t_{k+1}))\Bigg] \exp(Re^{-1}(\Delta t_{k+1})\mathsf{A}_i\mathsf{A}_i). \\ & \qquad \text{(D.24)}\end{aligned}$$

Numerically, we will be solving the majority of direct and adjoint variables in Fourier space. This avoids multiple Fourier transformations, and keeps computational costs manageable. Therefore, the concluding step, then, is to transform equation (D.24) into Fourier space, where the $\mathsf{A}_i\mathsf{A}_i$ operator simply transforms into a multiplication by the wave-number k. We finally arrive at a simple implementation of the form

$$\begin{aligned}\hat{\boldsymbol{u}}_{t_k}^\dagger &= \exp(-Re^{-1}\Delta t_{k+1}|k|^2)\hat{\boldsymbol{u}}_{t_{k+1}}^\dagger + [\frac{\Delta t_{k+1}^2}{2\Delta t_{k+2}} \exp(-Re^{-1}\Delta t_{k+2}|k|^2)\mathrm{RHS}(\hat{\boldsymbol{u}}^\dagger(t_{k+2})) \\ &\quad - \frac{\Delta t_{k+1}}{2\Delta t_{k+2}}[(\Delta t_{k+1} + 2\Delta t_{k+2}]\mathrm{RHS}(\hat{\boldsymbol{u}}^\dagger(t_{k+1}))] \exp(-Re^{-1}\Delta t_{k+1}|k|^2). \\ & \qquad \text{(D.25)}\end{aligned}$$

This backwards-in-time exponential time stepping then becomes the backbone of our adjoint implementation and allows us - in cases of Newtonian fluids - to avoid costly evaluations of the diffusion terms.