SMAI-JCM
SMAI Journal of Computational Mathematics

A robust, discrete-gradient descent procedure for optimisation with time-dependent PDE and norm constraints

Paul M. Mannix, Calum S. Skene,
Didier Auroux & Florence Marcotte

Volume 10 (2024), p. 1-28.
https://doi.org/10.5802/smai-jcm.104

The SMAI Journal of Computational Mathematics is a member of the Centre Mersenne for Open Scientific Publishing
http://www.centre-mersenne.org/
Submissions at https://smai-jcm.centre-mersenne.org/ojs/submission
e-ISSN: 2426-8399

SMAI Journal of Computational Mathematics
Vol. 10, 1-28 (2024)

A robust, discrete-gradient descent procedure for optimisation with time-dependent PDE and norm constraints

Paul M. Mannix [1]
Calum S. Skene [2]
Didier Auroux [3]
Florence Marcotte [4]

[1] Université Côte d'Azur, Inria, CNRS, LJAD, France
E-mail address: Paul.Mannix@univ-cotedazur.fr
[2] Department of Applied Mathematics, University of Leeds, West Yorkshire, UK
[3] Université Côte d'Azur, Inria, CNRS, LJAD, France
[4] Université Côte d'Azur, Inria, CNRS, LJAD, France.

Abstract. Many physical questions in fluid dynamics can be recast in terms of norm constrained optimisation problems; which in-turn, can be further recast as unconstrained problems on spherical manifolds. Due to the non-linearities of the governing PDEs, and the computational cost of performing optimal control on such systems, improving the numerical convergence of the optimisation procedure is crucial. Borrowing tools from the optimisation on manifolds community we outline a numerically consistent, discrete formulation of the direct-adjoint looping method accompanied by gradient descent and line-search algorithms with global convergence guarantees. We numerically demonstrate the robustness of this formulation on three example problems of relevance in fluid dynamics and provide an accompanying library `SphereManOpt`.

2020 Mathematics Subject Classification. 65N35, 15A15.

Keywords. optimal control, adjoint-based methods, optimisation on manifolds.

1. Introduction

Applications of norm constrained, nonlinear optimal control abound in fluid dynamics: spanning from the improvement of transport and mixing efficiencies in complex fluids (e.g. [12, 16]), to the stability analysis of thermo-acoustic oscillations (e.g. [27]), the modeling of transitions between a dynamical system's multiple basins of attraction (e.g. [28, 33]) or the forecasts of oceanographic systems (e.g. [4, 42, 53]). Because the governing partial differential equations (PDEs) are fully nonlinear, solutions to the corresponding optimisation problems must in general be calculated numerically. Due to the large dimension of both the control and state vectors and the significant cost of direct numerical simulations, efficient optimisation routines must be applied to address such problems. As our motivation is essentially numerical, we will restrict our attention to finite-dimension problems in what follows.

Given a discretised set of equations describing the evolution of the state vector $\boldsymbol{X} \in \mathbb{R}^n$, a general formulation of such norm-constrained problem is as follows

$$\begin{aligned} \min_{\boldsymbol{X}_0} \quad & J(\boldsymbol{X}, \boldsymbol{X}_0), \\ \text{s.t.} \quad & \boldsymbol{G}(\boldsymbol{X}, \boldsymbol{X}_0) \equiv \begin{cases} \frac{\mathrm{d}\boldsymbol{X}}{\mathrm{d}t} - \boldsymbol{F}(\boldsymbol{X}, t) = 0, \\ \boldsymbol{X}_0 - \boldsymbol{X}(t=0) = 0, \end{cases} \\ & c(\boldsymbol{X}_0) \equiv \|\boldsymbol{X}_0\|^2 - E_0 = 0, \end{aligned} \tag{1.1}$$

https://doi.org/10.5802/smai-jcm.104

where $\boldsymbol{X}_0 \in \mathbb{R}^n$ is the control variable we seek to optimise, and E_0 its energy. The objective function $J(\boldsymbol{X}, \boldsymbol{X}_0)$ is a quantity of interest to be minimised (or maximised), and $\boldsymbol{F}(\boldsymbol{X}, t)$ is a nonlinear operator describing how the system evolves in time. For example, $\boldsymbol{X}_0$ can be the initial velocity field of kinetic energy E_0, yielding optimal mixing over a finite time window $t \in (0, T]$ as the flow evolves according to its governing equations. In this case, well-mixedness can be achieved by minimising some measure of the fluid heterogeneity (such as the scale-sensitive mix-norms [24, 38, 54]). In the context of stability analysis, $\boldsymbol{X}_0$ can be the initial perturbation, which, for a given energy input, maximises for example the nonlinear growth of perturbations - thus allowing for a better characterisation of laminar-turbulent transitions in shear flows [28, 46]. Although (1.1) assumes control of the initial condition only, this can be easily modified to consider a time-dependent forcing or an alternative control variable.

1.1. **Adjoint formalism**

To solve the optimisation problem (1.1) one formulates the Lagrangian $\mathcal{L}$, related to the pure objective functional, as

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{X}_0; \boldsymbol{q}, \lambda) \equiv \widehat{J}(\boldsymbol{X}_0) + \lambda c(\boldsymbol{X_0}) = J(\boldsymbol{X}, \boldsymbol{X}_0) + \int_0^T \langle \boldsymbol{q}, \boldsymbol{G}(\boldsymbol{X}, \boldsymbol{X_0}) \rangle \, \mathrm{d}t + \lambda(\|\boldsymbol{X}_0\|^2 - E_0), \tag{1.2}$$

where $\boldsymbol{q}, \lambda$ denote the Lagrange multipliers (also termed adjoint variables) used to enforce these constraints, $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ denotes a vector inner product and $\|\boldsymbol{a}\|^2 = \langle \boldsymbol{a}, \boldsymbol{a} \rangle$ the associated norm. Here we choose to denote the pure objective functional by $\widehat{J}(\boldsymbol{X}_0)$, thus assuming $\boldsymbol{X}(\boldsymbol{X}_0)$ is implicitly defined for all $\boldsymbol{X}_0$ by the constraint equations. Any stationary point of $\mathcal{L}$ satisfies the following Euler-Lagrange equations:

$$\frac{\delta \mathcal{L}}{\delta \lambda} = \|\boldsymbol{X}_0\|^2 - E_0 = 0, \tag{1.3}$$

$$\frac{\delta \mathcal{L}}{\delta \boldsymbol{q}} = \boldsymbol{G}(\boldsymbol{X}, \boldsymbol{X_0}) = 0, \tag{1.4}$$

$$\frac{\delta \mathcal{L}}{\delta \boldsymbol{X}_T} = \frac{\delta J}{\delta \boldsymbol{X}_T} + \boldsymbol{q}_T = 0, \tag{1.5}$$

$$\frac{\delta \mathcal{L}}{\delta \boldsymbol{X}} = \frac{\partial \boldsymbol{G}^\dagger}{\partial \boldsymbol{X}} \boldsymbol{q} + \frac{\delta J}{\delta \boldsymbol{X}} = 0, \tag{1.6}$$

$$\frac{\delta \mathcal{L}}{\delta \boldsymbol{X}_0} = \frac{\delta J}{\delta \boldsymbol{X}_0} - \boldsymbol{q}_0 + 2\lambda \boldsymbol{X}_0 = 0, \tag{1.7}$$

where $\boldsymbol{X}_T, \boldsymbol{q}_T$ refers to the vectors at the final time, while $\dagger$ notation is used to denote the adjoint operator such that $\langle x, Ly \rangle = \langle L^\dagger x, y \rangle$. In general, reaching a stationary point of (1.2) demands an iterative procedure, whereby we solve (1.3) to (1.7) from top to bottom as given. That is: a random initial vector $\boldsymbol{X}_0$ is normalised to be of norm E_0, and the PDE constraint equations are solved forward in time from $t = 0 \rightarrow T$ using an appropriate time-stepping scheme [2, 3], whilst simultaneously calculating $\widehat{J}(\boldsymbol{X}_0)$. Subject to the final time condition (1.5), the adjoint equations (1.6)

$$\left(\frac{\partial}{\partial t} + \frac{\partial \boldsymbol{F}^\dagger}{\partial \boldsymbol{X}} \right) \boldsymbol{q} = \frac{\delta J}{\delta \boldsymbol{X}}, \tag{1.8}$$

are then solved backwards from $t = T \rightarrow 0$. Denoting the current and subsequent iterates of a gradient descent procedure by $k, k+1$ respectively, the control parameter is finally updated, using the gradient at iteration k (1.5), via

$$\boldsymbol{X}_0^{k+1} = \boldsymbol{X}_0^k - \alpha \frac{\delta \mathcal{L}}{\delta \boldsymbol{X}_0}^k (\lambda), \tag{1.9}$$

where $\alpha \in \mathbb{R}^+$ is an appropriately chosen step-size to ensure descent [59] and λ is determined to enforce $\|\boldsymbol{X}^{k+1}\|^2 = E_0$.

The first challenge when performing gradient descent is to accurately and efficiently estimate the gradient $\frac{\delta \mathcal{L}}{\delta \boldsymbol{X}_0}^k(\lambda)$. Using open-source automatic differentiation (AD) to obtain discrete gradients is one possible solution [23]. While this approach works well on forward codes written so as to be amenable to AD, such as dolfin-adjoint for example [13, 40], it can otherwise consume large amounts of memory and be extremely challenging to debug [43, 44]. Moreover, using AD to differentiate MPI-parallel codes also raises important difficulties regarding how to manage communication dataflow and dependencies [39, 56], which in practice hinders its applicability to computationally expensive fluid dynamics problems. When impractical to use AD for some of the reasons mentioned above, the alternative approach is to derive analytically the discrete adjoint equations using summation by parts. This approach, albeit established in the optimisation literature [20, 61], has been less discussed until recently in the aeronautics [18, 60] and fluid dynamics literature [14, 30, 51, 57].

Even when armed with an accurate gradient however, we must move in the correct direction and use an efficient gradient descent routine. Typically one enforces the equality constraint $c(\boldsymbol{X}_0)$ using a Lagrange multiplier λ as outlined above. Given however that this constraint naturally restricts $\boldsymbol{X}_0 \in \mathbb{R}^n$ to the spherical manifold $\mathcal{S}^{n-1} = \{\boldsymbol{X}_0 \mid \|\boldsymbol{X}_0\|^2 = E_0\}$, we can implement a consistent and therefore more efficient routine by exploiting this structure [1, 48], which we recall in Section 3.

1.2. **Issue 1: Discrete versus continuous**

To highlight the inconsistencies that typically arise when discretizing the previous (continuous) derivation, we recall that, although a linear operator $L = \partial_{xx}$ acting on function $u(x), -1 \leq x \leq 1$ with boundary data $u(\pm 1) = 0$ is self-adjoint with respect to $\int_{-1}^{1} pLu \,\mathrm{d}x = \int_{-1}^{1} Lpu \,\mathrm{d}x$, its corresponding matrix operator although dependent on the discretization used $L^{\dagger}_{i,j} \neq L_{i,j}$ seldom conserves this property [25, 34]. Similarly, integration by parts of the temporal derivative followed by discretisation, does not generally commute with discretisation followed by discrete integration by parts. To clarify this issue we consider the following discrete Lagrangian consisting of a linear constraint equation $\boldsymbol{X}_t = L\boldsymbol{X}$, discretised using a Crank-Nicolson temporal scheme

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{X}_0; \boldsymbol{q}, \lambda) = J(\boldsymbol{X}, \boldsymbol{X}_0) - \langle \boldsymbol{q}, M\boldsymbol{X} - \boldsymbol{f} \rangle + \lambda(\|\boldsymbol{X}_0\|^2 - E_0), \tag{1.10}$$

where $\Delta t = T/N, \boldsymbol{X} = (\boldsymbol{X}_0, \cdots, \boldsymbol{X}_N)$ and the constraint equations $M\boldsymbol{X} = \boldsymbol{f}$ (incorporating the constraint on the initial condition) now in discrete form read

$$\begin{pmatrix} I & 0 & \cdots & & \\ b & a & \ddots & & \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ & \cdots & 0 & b & a \end{pmatrix} \begin{pmatrix} \boldsymbol{X}_0 \\ \boldsymbol{X}_1 \\ \vdots \\ \vdots \\ \boldsymbol{X}_N \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}(t=0) \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix}, \tag{1.11}$$

where $a, b = (\frac{I}{\Delta t} - \frac{L}{2}), -(\frac{I}{\Delta t} + \frac{L}{2})$. Taking variations with respect to $\boldsymbol{X}$ we obtain $M^\dagger \boldsymbol{q} = \frac{\delta J}{\delta \boldsymbol{X}} + 2\lambda \boldsymbol{X}_0$ which in component form reads

$$\begin{pmatrix} I & b^\dagger & 0 & \cdots & \\ 0 & a^\dagger & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ & & \ddots & \ddots & b^\dagger \\ & & \cdots & 0 & a^\dagger \end{pmatrix} \begin{pmatrix} \boldsymbol{q}_0 \\ \boldsymbol{q}_1 \\ \vdots \\ \vdots \\ \boldsymbol{q}_N \end{pmatrix} = \begin{pmatrix} \frac{\delta J}{\delta \boldsymbol{X}_0} + 2\lambda \boldsymbol{X}_0 \\ \frac{\delta J}{\delta \boldsymbol{X}_1} \\ \vdots \\ \vdots \\ \frac{\delta J}{\delta \boldsymbol{X}_N} \end{pmatrix}. \tag{1.12}$$

Comparing these discrete Euler-Lagrange equations (1.12) with their continuous counterpart (1.5)–(1.7) one notices that: (1) the adjoint reverses the direction of information propagation, (2) the discrete adjoint of a *centered* differencing scheme (here for the temporal derivative) would remain consistent with its continuous adjoint if and only if $L = L^\dagger$ and (3) the discrete terminal/compatibility conditions at $t = 0, T$ differ from their continuous counterpart at $n = 0, N$. As we will show in this paper, the later issue is compounded for higher order schemes which contribute additional variations to more of the top and bottom rows. Throughout the remainder of this paper the term *discrete gradient* will be used to refer to the approach where the adjoint numerical scheme is obtained by differentiating the already-discretized direct equations. Conversely *continuous gradient* will be used here to refer to the approach where the adjoint equations are first derived from the continuous problem, then discretized. A scheme for which the solutions to both approaches coincide is termed adjoint/dual consistent [5, 22].

1.3. **Issue 2: Constrained gradient update**

The second issue is that we still have a norm-constrained update step (1.9). For every choice of step-size α we must apply the normalisation constraint to choose a corresponding λ. This relegates the applicability of line-search algorithms for unconstrained optimisation, which allow us to choose step-sizes α guaranteeing descent [59]. Aside from the in-applicability of standard line-search algorithms however, further issues exist with the Lagrange multiplier approach. Using $\nabla f = \frac{\delta f}{\delta \boldsymbol{X}_0}$ to denote the variational derivative, it follows that for an optimum to exist, a necessary condition is that a λ exists such that

$$\frac{\delta \mathcal{L}}{\delta \boldsymbol{X}_0} = \nabla \widehat{J}(\boldsymbol{X}_0) + \lambda \nabla c(\boldsymbol{X}_0) = 0, \tag{1.13}$$

implying that the gradient of the constraint equations $\nabla \widehat{J}(\boldsymbol{X}_0)$ and norm constraint $\nabla c(\boldsymbol{X}_0)$ are parallel. To render this condition sufficient requires that λ satisfy the Karush-Kuhn-Tucker (KKT) conditions [59], which in-turn requires computing the Hessian of (1.13). If $\nabla \widehat{J}(\boldsymbol{X}_0), \nabla c(\boldsymbol{X}_0)$ are not parallel, (1.13) is not satisfied and we ideally, according to a first order approximation, choose a search direction $\boldsymbol{d}$ such that $\nabla \widehat{J}(\boldsymbol{X}_0)^T \boldsymbol{d} < 0$ and $\nabla c(\boldsymbol{X}_0)^T \boldsymbol{d} = 0$. A good candidate which also defines the gradient on the spherical manifold $\mathcal{S}$ [6], is the tangent-vector

$$-\boldsymbol{d} = \nabla \widehat{J}(\boldsymbol{X}_0) - \left(\frac{\nabla c(\boldsymbol{X}_0)^T \nabla \widehat{J}(\boldsymbol{X}_0)}{\|\nabla c(\boldsymbol{X}_0)\|^2} \right) \nabla c(\boldsymbol{X}_0), \tag{1.14}$$

as it satisfies the previous inequality and equality constraints but crucially is independent of λ. Thus by restricting $\boldsymbol{X}_0$ to $\mathcal{S}$ and updating $\boldsymbol{X}_0$ on geodesics of $\mathcal{S}$ for all values of the step size α [11], we can omit the equality constraint in (1.2). Furthermore $\nabla \widehat{J}(\boldsymbol{X}_0) = 0$ now provides a necessary condition for a local minimiser [6].

What remains, following [48, 49], is to appropriately adapt the existing line-search algorithms such that updating $\boldsymbol{X}_0^k$ leads to global convergence, and to assess their efficiency on a suite of non-trivial fluid dynamics examples. Aside from the `Pymanopt` library for optimisation on matrix manifolds [55],

SLSQP [31] presents the only alternative for problems with equality constraints. Owing to the construction, storage $\mathcal{O}(n^2)$ and inversion $\mathcal{O}(n^3)$ of the Hessian matrix SLSQP is however numerically too demanding to handle high-dimension control variables [26, 59]. Conversely while `Pymanopt` implements computationally efficient methods, its linesearch does not satisfy the necessary conditions in order to guarantee global convergence [49].

In what follows, we address the previously highlighted issues and test their numerical remedy against a suite of physical problems. Given the additional complexity represented by the derivation of a consistent, discrete adjoint numerical scheme, compared to a continuous gradient approach (where the same numerical scheme is used to discretize the forward and backward equations), we also conduct a systematic numerical comparison between discrete and continuous adjoint optimisation, which excluding the studies of [57] and [15] is currently lacking. To this end, we first derive the discrete adjoint equations in Section 2; following [48] we then outline a consistent gradient update with guaranteed global descent in Section 3; and finally we combine these developments in Section 4 to demonstrate their superior performance. Perspectives for further work are then discussed in the conclusive section.

2. **Discrete adjoint equations**

A set of *spatially discretised* coupled evolution equations can be written as

$$\mathcal{M}\partial_t \boldsymbol{X} + \mathcal{L}\boldsymbol{X} = \mathcal{F}(\boldsymbol{X}, t), \tag{2.1}$$

where $\boldsymbol{X}$ is the vector of unknowns, $\mathcal{M}, \mathcal{L}$ are linear operators encompassing both the constraint equations and system boundary conditions and the right hand side $\mathcal{F}$ encompasses all nonlinear and non-autonomous terms. Then we can write a general *s-step* multistep implicit-explicit (IMEX) method [3] as

$$\sum_{j=0}^{s} a_j \mathcal{M}\boldsymbol{X}_{n-j} + \sum_{j=0}^{s} b_j \mathcal{L}\boldsymbol{X}_{n-j} = \sum_{j=1}^{s} c_j \mathcal{F}_{n-j}, \tag{2.2}$$

where index n denotes the n^{th} time step, j the internal stages and the right hand side is evaluated and treated as a vector. Notably when initiating time-stepping from a single initial condition, we have insufficient data to use 2nd or higher order schemes directly and must revert to a first order scheme. While diagonally implicit Runge-Kutta (DIRK/SDIRK) integration schemes [2] solve this issue and are L-stable, their numerical overhead, both in terms of CPU and memory, is much higher however and their discrete adjoint derived by [20, 47, 57, 60] complexifies faster.

We first present the discrete adjoint equations in their full generality, and subsequently consider their numerical performance on particular examples by making comparisons with their continuous formulation. As multistep schemes of order $s > 2$ are not A-stable and objective functions can be time-integrated with $\mathcal{O}(\Delta t^2)$ accuracy at best, higher order schemes are not considered. We note that the adjoint of a subclass of IMEX schemes known as backward differencing schemes (BDF) has previously been derived up to third order by [45].

2.1. **A second order IMEX scheme**

Partitioning the time interval $t \in (0, T]$ as $t_n = n\Delta t, n \in (0, N = T/\Delta t]$ the discrete Lagrangian, with normalisation constraints omitted, is given as

$$\begin{aligned}\mathcal{L} = J(\boldsymbol{X}) &- \langle \boldsymbol{q}_0, \boldsymbol{X}_0 - \boldsymbol{X}(\boldsymbol{x}, t=0)\rangle \\ &- \langle \boldsymbol{q}_1, \mathcal{M}(\tilde{a}_0\boldsymbol{X}_1 + \tilde{a}_1\boldsymbol{X}_0) + \mathcal{L}(\tilde{b}_0\boldsymbol{X}_1 + \tilde{b}_1\boldsymbol{X}_0) - \tilde{c}_1\mathcal{F}(\boldsymbol{X}_0)\rangle \\ &- \sum_{n=1}^{N-1}\Big\langle \boldsymbol{q}_{n+1}, \mathcal{M}(\tilde{a}_0\boldsymbol{X}_{n+1} + \tilde{a}_1\boldsymbol{X}_n + a_2\boldsymbol{X}_{n-1}) + \mathcal{L}(\tilde{b}_0\boldsymbol{X}_{n+1} + \tilde{b}_1\boldsymbol{X}_n + b_2\boldsymbol{X}_{n-1}) \\ &\qquad - \Big[\tilde{c}_1\mathcal{F}(\boldsymbol{X}_n) + c_2\mathcal{F}(\boldsymbol{X}_{n-1})\Big]\Big\rangle,\end{aligned} \tag{2.3}$$

where tilde notation is used to denote the coefficients retained if considering a first order scheme only, and the objective function is for example given by either

$$J(\boldsymbol{X}) = \frac{\Delta t}{2}(\langle \boldsymbol{X}_0, W\boldsymbol{X}_0\rangle + \langle \boldsymbol{X}_N, W\boldsymbol{X}_N\rangle) + \Delta t\sum_{n=1}^{N-1}\langle \boldsymbol{X}_n, W\boldsymbol{X}_n\rangle, \tag{2.4}$$

or $J_T(\boldsymbol{X}) = \langle \boldsymbol{X}_N, W\boldsymbol{X}_N\rangle$ where W is a known matrix operator. Taking variations with respect to the adjoint variables $\boldsymbol{q}_n$ and setting $\frac{\delta\mathcal{L}}{\delta \boldsymbol{q}_n} = 0$ we recover the initial condition and the constraint equations, while with respect to the primal variables $\boldsymbol{X}_n$ we obtain

$$\begin{aligned}\sum_{n=0}^{N}\Big\langle \delta\boldsymbol{X}_n, \frac{\delta\mathcal{L}}{\delta\boldsymbol{X}_n}\Big\rangle = \sum_{n=0}^{N}\Big\langle \delta\boldsymbol{X}_n, \frac{\partial J}{\partial \boldsymbol{X}_n}\Big\rangle &- \langle \delta\boldsymbol{X}_0, \boldsymbol{q}_0\rangle \\ &- \langle \delta\boldsymbol{X}_1, \tilde{a}_0\mathcal{M}^\dagger\boldsymbol{q}_1 + \tilde{b}_0\mathcal{L}^\dagger\boldsymbol{q}_1\rangle - \Big\langle \delta\boldsymbol{X}_0, \tilde{a}_1\mathcal{M}^\dagger\boldsymbol{q}_1 + \tilde{b}_1\mathcal{L}^\dagger\boldsymbol{q}_1 - \tilde{c}_1\frac{\partial F}{\partial \boldsymbol{X}_0}^\dagger\boldsymbol{q}_1\Big\rangle \\ &- \sum_{n=2}^{N}\langle \delta\boldsymbol{X}_n, \tilde{a}_0\mathcal{M}^\dagger\boldsymbol{q}_n + \tilde{b}_0\mathcal{L}^\dagger\boldsymbol{q}_n\rangle \\ &- \sum_{n=1}^{N-1}\langle \delta\boldsymbol{X}_n, \tilde{a}_1\mathcal{M}^\dagger\boldsymbol{q}_{n+1} + \tilde{b}_1\mathcal{L}^\dagger\boldsymbol{q}_{n+1}\rangle \\ &- \sum_{n=0}^{N-2}\langle \delta\boldsymbol{X}_n, a_2\mathcal{M}^\dagger\boldsymbol{q}_{n+2} + b_2\mathcal{L}^\dagger\boldsymbol{q}_{n+2}\rangle \\ &- \sum_{n=1}^{N-1}\Big\langle \delta\boldsymbol{X}_n, -\tilde{c}_1\frac{\partial F}{\partial \boldsymbol{X}_n}^\dagger\boldsymbol{q}_{n+1}\Big\rangle - \sum_{n=0}^{N-2}\Big\langle \delta\boldsymbol{X}_n, -c_2\frac{\partial F}{\partial \boldsymbol{X}_n}^\dagger\boldsymbol{q}_{n+2}\Big\rangle,\end{aligned} \tag{2.5}$$

where we have permuted the indices in each of these terms and $\frac{\partial F}{\partial \boldsymbol{X}_n}$ is used to denote the Jacobian of the nonlinear term. Grouping terms of (2.5) by their indices and setting them equal to zero we obtain

the discrete Euler-Lagrange equations for $n = N, N-1$, $N-2 \geq n \geq 1$ and $n = 0$ respectively

$$(\tilde{a}_0\mathcal{M}^\dagger + \tilde{b}_0\mathcal{L}^\dagger)\boldsymbol{q}_N = \frac{\partial J}{\partial \boldsymbol{X}_N}, \quad (2.6)$$

$$(\tilde{a}_0\mathcal{M}^\dagger + \tilde{b}_0\mathcal{L}^\dagger)\boldsymbol{q}_{N-1} + (\tilde{a}_1\mathcal{M}^\dagger + \tilde{b}_1\mathcal{L}^\dagger)\boldsymbol{q}_N - \tilde{c}_1\frac{\partial F}{\partial \boldsymbol{X}_{N-1}}^\dagger \boldsymbol{q}_N = \frac{\partial J}{\partial \boldsymbol{X}_{N-1}}, \quad (2.7)$$

$$(\tilde{a}_0\mathcal{M}^\dagger+\tilde{b}_0\mathcal{L}^\dagger)\boldsymbol{q}_n+(\tilde{a}_1\mathcal{M}^\dagger+\tilde{b}_1\mathcal{L}^\dagger)\boldsymbol{q}_{n+1}+(a_2\mathcal{M}^\dagger+b_2\mathcal{L}^\dagger)\boldsymbol{q}_{n+2}-\frac{\partial F}{\partial \boldsymbol{X}_n}^\dagger[\tilde{c}_1\boldsymbol{q}_{n+1}+c_2\boldsymbol{q}_{n+2}] = \frac{\partial J}{\partial \boldsymbol{X}_n}, \quad (2.8)$$

$$\boldsymbol{q}_0 + (\tilde{a}_1\mathcal{M}^\dagger + \tilde{b}_1\mathcal{L}^\dagger)\boldsymbol{q}_1 + (a_2\mathcal{M}^\dagger + b_2\mathcal{L}^\dagger)\boldsymbol{q}_2 - \frac{\partial F}{\partial \boldsymbol{X}_0}^\dagger[\tilde{c}_1\boldsymbol{q}_1 + c_2\boldsymbol{q}_2] = \frac{\partial J}{\partial \boldsymbol{X}_0}. \quad (2.9)$$

The solution of the $\mathcal{O}(\Delta t)$ accurate counterpart of these equations using a preconditioned scheme, applicable for example to discretisations of wall bounded domains is given in Appendix A. Concentrating on the minor and feasible modifications of simple schemes which help achieve accurate convergence, we have implemented primarily first order schemes in this paper.

Comparing expressions (2.6)–(2.9) with their continuous counterpart (1.5)–(1.7), we see that while the time-stepping scheme is unaffected for $s \leq n \leq N-1$, the compatibility conditions demand the inversion of non-local linear operators at $n = N, t = T$ and an explicit evaluation at $n = 0, t = 0$. For a fully periodic domain discretised using a self-adjoint Fourier basis the error is likely to be small, but for wall bounded domains where boundary layers are present, such as the optimal mixing example presented in 2.2.3, the error is potentially significant.

2.2. **Numerical performance**

To validate the adjoint of the discrete equations and justify the additional complexity of its implementation we now report numerical experiments on three example cases that showcase its superior performance. The consistency of the adjoint is checked by testing the Taylor remainder, i.e. verifying that for any arbitrary perturbation $\delta\boldsymbol{X}$ of the initial condition,

$$\lim_{h\to 0} |\widehat{J}(\boldsymbol{X}_0 + h\delta\boldsymbol{X}) - \widehat{J}(\boldsymbol{X}_0)| \to 0, \quad \text{at} \quad \mathcal{O}(|h|), \quad (2.10)$$

and

$$\lim_{h\to 0} \left|\widehat{J}(\boldsymbol{X}_0 + h\delta\boldsymbol{X}) - \widehat{J}(\boldsymbol{X}_0) - h\left\langle \delta\boldsymbol{X}, \frac{\delta\mathcal{L}}{\delta\boldsymbol{X}_0}\right\rangle\right| \to 0, \quad \text{at} \quad \mathcal{O}(|h|^2). \quad (2.11)$$

An additional test of interest is that the first difference in (2.11), divided by h, is equivalent to the inner product in the last term, to a number of decimal places. This is preferable when $h \ll 1$ as it avoids multiplying very small and large numbers. In all tests which follow we generate $\boldsymbol{X}_0, \delta\boldsymbol{X}_0$ randomly and re-scale such that $\|\boldsymbol{X}_0\|^2 = \|\delta\boldsymbol{X}_0\|^2 = 1$. In practice, it is strongly advised whenever possible to also choose $\boldsymbol{X}_0$ close to a representative state of interest of the problem studied, as this ensures consistency of gradient definition with respect to problem relevant parameter vectors.

2.2.1. *First test-case: a dynamo problem*

In many astrophysical objects, magnetic fields are generated by the motions of electrically conducting fluids, through a process known as the dynamo effect [32]. A dynamo optimisation problem first considered by [58], comprises finding the velocity field $\boldsymbol{U}$ and the initial magnetic field $\boldsymbol{B}_0$ (constrained by some energy budget), which maximise the magnetic field's growth by some target time T in a triply periodic box. Here the velocity field is assumed to be frozen, i.e. independent of time and the evolution

of the magnetic field. Following [58], this optimisation problem is written in dimensionless form as

$$\begin{aligned} \max_{\boldsymbol{B}_0,\boldsymbol{U}} \quad & \widehat{J}_T(\boldsymbol{B}_0,\boldsymbol{U}) = \langle \boldsymbol{B}_T, \boldsymbol{B}_T \rangle \equiv \frac{1}{V}\int \boldsymbol{B}_T^2 \,\mathrm{d}V, \\ \text{s.t.} \quad & \langle \boldsymbol{B}_0, \boldsymbol{B}_0 \rangle = 1, \quad \langle \boldsymbol{U}, \boldsymbol{U} \rangle = 1, \\ & \frac{\partial \boldsymbol{B}}{\partial t} - \nabla \times (\boldsymbol{U} \times \boldsymbol{B}) - \nabla \Pi - Rm^{-1}\nabla^2 \boldsymbol{B} = 0, \\ & \nabla \cdot \boldsymbol{B} = 0, \quad \nabla \cdot \boldsymbol{U} = 0, \end{aligned} \tag{2.12}$$

where V is the volume of the box and Rm (the magnetic Reynolds number) is a control parameter quantifying the relative strength of electromagnetic induction compared to ohmic dissipation. Due to the domain periodicity the problem can be spatially discretised using a Fourier basis. Spatial differentiation amounts to multiplication by $i\mathbf{k}$ in Fourier space (where $\mathbf{k}$ is the wavenumber) such that integration by parts discretely or continuously yield the same adjoint operators. As a result, only errors associated with the temporal discretization can arise in the gradient estimation.

Equation (2.12) and its adjoint (C.2) are time-integrated in the open-source code Dedalus [8] using a Crank-Nicolson Adams-Bashforth 1 (CNAB1) scheme. The problem setup and numerical resolutions are the same as in [58] and the dimensionless control parameters are here both set to $Rm = T = 1$. First we test the gradient with respect to the magnetic field $\delta\mathcal{L}/\delta\boldsymbol{B}_0$, holding $\boldsymbol{U}$ fixed as prescribed in [58], and subsequently the gradient with respect to the velocity field $\delta\mathcal{L}/\delta\boldsymbol{U}$ by holding $\boldsymbol{B}_0$ fixed. Table 2.1 compares the gradient approximation obtained from the continuous adjoint equations (C.2) with that of the discrete adjoint equations (C.6)–(C.11). Both are derived in terms of a Lagrangian formulation of (2.12) in Section 2. To calculate the order of convergence when assessing the validity of (2.11), we assume that for any h_i

$$\mathcal{W}(h_i) = \left| \widehat{J}(\boldsymbol{X}_0 + h_i \delta\boldsymbol{X}) - \widehat{J}(\boldsymbol{X}_0) - h_i \left\langle \delta\boldsymbol{X}, \frac{\delta\mathcal{L}}{\delta\boldsymbol{X}_0} \right\rangle \right| \approxeq C|h_i|^\gamma, \tag{2.13}$$

where C and the exponent γ are constants. Making the approximation (2.13) for two different h_i we estimate

$$\gamma \approxeq \frac{\log\left(\mathcal{W}(h_1)/\mathcal{W}(h_2)\right)}{\log(|h_1|/|h_2|)}. \tag{2.14}$$

In all tables a significant improvement is seen when using the discrete adjoint while the continuous formulation cannot provide the mathematically consistent approximation demanded by (2.11).

2.2.2. *Second test-case: Swift-Hohenberg multi-stability*

The Swift-Hohenberg equation constitutes a paradigm for studying pattern formation in physics. It is known to have multiple stable solutions coexisting for the same parameter values, characterized by different energy levels [29]. This naturally raises the question of triggering transition from one stable state to another. Following [33] we consider the problem of selecting an initial condition $u_0 = u(x, t = 0)$ of energy E_0, which when added to the stable, zero-energy equilibrium $u = 0$ induces a transition to a new, higher-energy equilibrium. This can be achieved by considering

$$\begin{aligned} \max_{u_0} \quad & \widehat{J}(u) = \int_{t=0}^{t=T}\int_x |u(x,t)|^2 \,\mathrm{d}x\,\mathrm{d}t, \\ \text{s.t.} \quad & \int_x \frac{1}{2}|u_0|^2 \,\mathrm{d}x = E_0, \\ & \partial_t u + (1+\partial_x^2)^2 u - au = 1.8u^2 - u^3, \end{aligned} \tag{2.15}$$

on a periodic domain of dimensionless length $L = 12\pi$ with parameter $a = -0.3$ and target time $T = 50$ following [33]. These values ensure that the zero-energy state $u = 0$ is linearly stable to

Table 2.1. Comparison of the continuous and discrete gradient for $N_{x,y,z} = 24$ Fourier modes, time-step Δt =1e-03 as per [58]. (a) The velocity field $\boldsymbol{U}$ as given in [58] is held fixed. (b) The initial magnetic field $\boldsymbol{B}_0$ is randomly chosen and held fixed. In both cases the vectors $\boldsymbol{B}_0, \delta\boldsymbol{B}, \delta\boldsymbol{U}$ of unit norm are generated by time-stepping vectors of noise a small number of iterations and then re-normalising. This ensures the boundary and divergence free conditions are satisfied whilst it avoids biasing the initial data.

(a)		Continuous		Discrete $\mathcal{O}(\Delta t)$	
h	$\lvert\widehat{J}(\boldsymbol{B}_0 + h\delta\boldsymbol{B}) - \widehat{J}(\boldsymbol{B}_0)\rvert/h$	$\lvert\langle\delta\boldsymbol{B}, \frac{\delta\mathcal{L}}{\delta\boldsymbol{B}_0}\rangle\rvert$	Order γ	$\lvert\langle\delta\boldsymbol{B}, \frac{\delta\mathcal{L}}{\delta\boldsymbol{B}_0}\rangle\rvert$	Order γ
5.00e-05	5.60377e-03	5.60419e-03	0.41015	5.60363e-03	1.9999
2.50e-05	5.60370e-03	5.60419e-03	0.77621	5.60363e-03	2.0000
1.25e-05	5.60367e-03	5.60419e-03	0.89991	5.60363e-03	1.9999
6.25e-06	5.60365e-03	5.60419e-03	0.95244	5.60363e-03	2.0000
(b)		Continuous		Discrete $\mathcal{O}(\Delta t)$	
h	$\lvert\widehat{J}(\boldsymbol{U} + h\delta\boldsymbol{U}) - \widehat{J}(\boldsymbol{U})\rvert/h$	$\lvert\langle\delta\boldsymbol{U}, \frac{\delta\mathcal{L}}{\delta\boldsymbol{U}}\rangle\rvert$	Order γ	$\lvert\langle\delta\boldsymbol{U}, \frac{\delta\mathcal{L}}{\delta\boldsymbol{U}}\rangle\rvert$	Order γ
5.00e-05	2.65775e-05	2.60762e-05	1.0024	2.65767e-05	2.0002
2.50e-05	2.65771e-05	2.60762e-05	1.0012	2.65767e-05	2.0009
1.25e-05	2.65769e-05	2.60762e-05	1.0006	2.65767e-05	2.0039
6.25e-06	2.65768e-05	2.60762e-05	1.0003	2.65767e-05	2.0226

infinitesimal perturbations, but unstable to some finite-amplitude perturbations which can drive the system away to a different stable state. Discretising the spatial operators using a Fourier basis we once again ensure that only temporal errors can contaminate the gradient estimate. Equation (2.15) and its discrete adjoint (B.5) are time-integrated in Dedalus using a first order backwards differencing (SBDF1) scheme.

Table 2.2. Test of gradient's accuracy for $N = 256, \Delta t = 0.05$ and an SBDF1 time-stepping scheme as per [33]. The discrete equations achieve a consistent gradient valid to 5 decimal places, however smaller h will provide a yet more consistent approximation.

		Continuous		Discrete $\mathcal{O}(\Delta t)$	
h	$\lvert\widehat{J}(u + h\delta u) - \widehat{J}(u)\rvert/h$	$\lvert\langle\delta u, \frac{\delta\mathcal{L}}{\delta u}\rangle\rvert$	Order γ	$\lvert\langle\delta u, \frac{\delta\mathcal{L}}{\delta u}\rangle\rvert$	Order γ
5.000e-05	2.797111	3.171736	0.999860	2.797075	2.002582
2.500e-05	2.797093	3.171736	0.999930	2.797075	2.001365
1.250e-05	2.797084	3.171736	0.999965	2.797075	2.000622
6.125e-06	2.797080	3.171736	0.999982	2.797075	2.000059

Table 2.2 reports a comparison of the continuous and discrete gradient by considering a fixed perturbations of noise for different h. While several digits of accuracy and a consistent gradient are obtained using the discrete equations we were unable to obtain a consistent gradient using the continuous equations for any choice of Δt or h. Having demonstrated the superiority of the discrete adjoint approach on two spatially self-adjoint discretisations, we now consider a wall bounded problem in which both the temporal and spatial discretisation can contaminate the gradient.

2.2.3. *Third test-case: optimal mixing problem*

In both the natural environment and industrial applications the mixing of fluids plays a crucial role. In the ocean for example, the irreversible mixing which occurs is the consequence of a complex interplay between density stratification and background shear playing out in a vast range of length-scales. A

grand challenge in fluid dynamics community is to understand how the turbulent motions which ensue from these effects transport warm saline water, biomass and organisms such as phytoplankton across density surfaces in the oceans [9]. An idealised example of optimal mixing is the optimal stirring strategy in a plane, pressure-driven channel flow [17, 24, 37]. This example considers what is, for example, the most efficient way of mixing a stably stratified layer of hot and cold fluid? This question may be posed as an optimisation problem, where the initial velocity perturbation $\boldsymbol{u}_0$ is determined so as to optimise well-mixedness. This can be achieved by minimising the following mix-norm for the density variation:

$$\begin{aligned}
\min_{\boldsymbol{u}_0} \quad & \widehat{J}(\boldsymbol{u}, b) = \frac{1}{2}\langle|\nabla^{-1} b(\boldsymbol{x}, T)|^2\rangle, \\
\text{s.t.} \quad & \frac{1}{2}\langle|\boldsymbol{u}_0|^2\rangle = E_0, \\
& \frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{U}_0 \cdot \nabla)\boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)[\boldsymbol{U}_0 + \boldsymbol{u}] - \nabla p + Rib\widehat{\boldsymbol{z}} - \frac{1}{Re}\nabla^2 \boldsymbol{u}, \\
& \frac{\partial b}{\partial t} + ([\boldsymbol{U}_0 + \boldsymbol{u}] \cdot \nabla) b - \frac{1}{Pe}\nabla^2 b, \quad \nabla \cdot \boldsymbol{u} = 0,
\end{aligned} \tag{2.16}$$

where the non-dimensional Reynolds and Péclet numbers $Re, Pe = 500$ characterise the strength of flow inertia to viscous and thermal diffusion respectively and the Richardson number $Ri = 0.05$ quantifies the strength of buoyancy relative to the background shear. Although we do not present it here, in `SphereManOpt` we include the additional test case where the time averaged kinetic energy is maximised instead, as in [37]. We assume periodicity in the streamwise direction $x \in [0, 4\pi]$ while in the shearwise direction $z \in [-1, 1]$ we impose no-slip velocity, the pressure gauge, the no-flux condition on the deviation density and its accompanying gauge

$$\boldsymbol{u}(x, z = \pm 1, t) = 0, \ \int_V p \, \mathrm{d}V = 0, \quad \partial_z b(x, z = \pm 1, t) = 0, \ \int_V b \, \mathrm{d}V = 0. \tag{2.17}$$

The latter gauge is essential as it (1) ensures that the mix-norm is well defined [54] and (2) prevents the optimisation from stalling due to an undefined gauge freedom. The background shear flow and the initial density deviation profile are given by

$$\boldsymbol{U}_0 = (1 - z^2)\widehat{\boldsymbol{x}}, \quad b(\boldsymbol{x}, t = 0) = -\frac{1}{2}\operatorname{erf}\left(\frac{z}{\delta_0}\right), \quad \text{with} \quad \delta_0 = 0.125, \tag{2.18}$$

a value larger than that of previous studies to facilitate the use of a spectral method. Following [37] the initial amplitude is set to $E_0 = 0.02$ and the optimisation window to $T = 5$. This slightly shorter optimisation window is chosen so as to avoid excessively small values of the mix-norm which were found to induce rounding errors. (2.16) and its discrete adjoint (D.2) are time-integrated in Dedalus using a first order backwards differencing (SBDF1). In contrast to the previous examples the spatial operators are discretised using a Fourier basis to treat the streamwise direction and a Chebyshev basis in the shearwise direction. Consequently the discrete operators appearing in (2.6)–(2.9) are no longer self-adjoint, such that both temporal and spatial errors can now contaminate the gradient estimate.

Table 2.3 reports a comparison of the continuous and discrete gradient by considering a fixed perturbations of noise for different h and for different objective functions. We find that several digits of accuracy and a consistent gradient are obtained using the discrete equations while we were unable to obtain a consistent gradient using the continuous equations for any choice of Δt or h.

2.3. **Run time**

Table 2.4 reports the efficiency of this section's gradient tests. The adjoint of the kinematic dynamo problem having twice the forward solve's dimension costs almost double as anticipated. The optimal

TABLE 2.3. Test of gradient's accuracy for $N_x, N_z = 256, 128, \Delta t = 1e-03$ using an SBDF1 scheme. The initial perturbation velocities $\boldsymbol{u}, \delta\boldsymbol{u}$ of unit norm are generated by time-stepping vectors of noise a small number of iterations and then re-normalising.

		Continuous		Discrete $\mathcal{O}(\Delta t)$	
h	$\lvert\widehat{J}(\boldsymbol{u}+h\delta\boldsymbol{u})-\widehat{J}(\boldsymbol{u})\rvert/h$	$\lvert\langle\delta\boldsymbol{u}, \frac{\delta\mathcal{L}}{\delta\boldsymbol{u}}\rangle\rvert$	Order γ	$\lvert\langle\delta\boldsymbol{u}, \frac{\delta\mathcal{L}}{\delta\boldsymbol{u}}\rangle\rvert$	Order γ
5.00e-05	7.51875e-03	1.4081e-02	1.0079	7.51848e-03	1.9999
2.50e-05	7.51862e-03	1.4081e-02	1.0040	7.51848e-03	1.9995
1.25e-05	7.51855e-03	1.4081e-02	1.0020	7.51848e-03	1.9974
6.25e-06	7.51852e-03	1.4081e-02	1.0010	7.51848e-03	1.9886

TABLE 2.4. Ratio of the adjoint integration time versus the forwards integration time as computed using a single node with 2 AMD Epyc 7302 @ 3 GHz - 16 cores processors. To minimise memory usage only the forwards solve in spectral space is stored.

Problem	Forward Solve (cpu-s)	Adjoint Solve (cpu-s)	Ratio
Kinematic dynamo	50.432	91.94	1.823
Swift-Hohenberg	2.473	0.388	0.156
Optimal mixing	2794.185	2476.635	0.886

mixing and Swift-Hohenberg problems cost equal or substantially less. Computations require the forwards solve in coefficient space only, thus minimising memory usage as compared with automatic differentiation.

3. **Gradient descent on a spherical manifold**

In the previous section we detailed how to obtain a numerically consistent approximation of the gradient of the pure functional $\widehat{J}(\boldsymbol{X}_0)$. Extending this further we now consider

$$\min_{\widehat{\boldsymbol{X}}_0 \in \mathcal{S}} \quad \widehat{J}(\widehat{\boldsymbol{X}}_0) = J(\boldsymbol{X}(\widehat{\boldsymbol{X}}_0), \widehat{\boldsymbol{X}}_0), \tag{3.1}$$

where

$$\mathcal{S} = \{\widehat{\boldsymbol{X}} \mid \|\widehat{\boldsymbol{X}}\|^2 = E_0\}, \tag{3.2}$$

such that, by restricting $\widehat{\boldsymbol{X}}_0$ to the spherical manifold $\mathcal{S}$, the equality constraints are built-in. By defining (3.1) as an unconstrained optimisation problem we can now proceed in outlining an appropriate gradient descent procedure, either by using the rotation method [11] (also termed exponential map in the literature on optimisation on Riemannian manifolds [1]) which moves $\widehat{\boldsymbol{X}}_0$ along geodesics of $\mathcal{S}$ in the search direction $\boldsymbol{d}_k$, or using the retraction formulae of [1] which is computationally simpler and easily generalises to the consideration of L^p norms [50] (our discussion concentrates exclusively on L^2-norms).

In the fluid mechanics community, the rotation method was first used to perform norm-constrained optimisation on a nonlinear PDE problem by [16] who combined it with a Polak-Ribière conjugate gradient method [21]. They achieved faster convergence with the rotation method than with the Lagrange multiplier method. More recently [52] highlighted this link and utilised Riemannian optimisation techniques for performing a unit-norm constrained linear optimisation problem. In what follows we outline the recent theoretical developments concerning steepest descent [6] and conjugate-gradient descent developed by [48, 49]. This is followed by our numerical validation of their formulae, which using a stringent line-search is shown to provide convergence consistent with theoretical guarantees.

3.1. **Conjugate gradient formulation**

Following [6, 48] we denote the parameter vector at iterate k by $\widehat{\boldsymbol{X}}_k$, the Euclidean gradient of the objective function by $\nabla\widehat{J}(\widehat{\boldsymbol{X}}_k)$ and the tangent-gradient of the objective function (i.e. its gradient on the spherical manifold $\mathcal{S}$) by

$$\boldsymbol{g}_k = \left(I - \frac{\widehat{\boldsymbol{X}}_k\widehat{\boldsymbol{X}}_k^T}{\|\widehat{\boldsymbol{X}}_k\|^2}\right)\nabla\widehat{J}(\widehat{\boldsymbol{X}}_k). \tag{3.3}$$

Starting from an initial guess $\widehat{\boldsymbol{X}}_0$, restricted to $\mathcal{S}$, gradient descent proceeds via the recurrence

$$\begin{aligned}\widehat{\boldsymbol{X}}_{k+1} &= \frac{(\widehat{\boldsymbol{X}}_k + \alpha_k\boldsymbol{d}_k)}{\|\widehat{\boldsymbol{X}}_k + \alpha_k\boldsymbol{d}_k\|}\sqrt{E_0},\\ \boldsymbol{d}_k &= -\boldsymbol{g}_k + \beta_k\ s_k\ \mathcal{T}_{\widehat{\boldsymbol{X}}_k}(\boldsymbol{d}_{k-1}),\end{aligned} \tag{3.4}$$

where β_k is the conjugate-gradient update parameter [48, 49] and the step size α_k is determined using a line-search procedure in order to find an improved minimum of $\widehat{J}(\widehat{\boldsymbol{X}}_0)$. For steepest-descent $\beta_k = 0$ is set, while in the conjugate-gradient method we use a combined Riemannian Polak-Ribière and Fletcher-Reeves type update following a general formulation put forward by [49], who demonstrated that the following choice of β_k yields global convergence of the conjugate-gradient algorithm:

$$\beta_k = \max\{0, \min\{\beta_k^{R-PR}, \beta_k^{R-FR}\}\}, \tag{3.5}$$

with

$$\beta_k^{R-PR} = \frac{\langle\boldsymbol{g}_k, \boldsymbol{g}_k - \mathcal{T}_{\widehat{\boldsymbol{X}}_k}(\boldsymbol{g}_{k-1})\rangle}{\|\boldsymbol{g}_{k-1}\|^2}, \quad \beta_k^{R-FR} = \frac{\|\boldsymbol{g}_k\|^2}{\|\boldsymbol{g}_{k-1}\|^2}. \tag{3.6}$$

The vector transport operator

$$\mathcal{T}_{\widehat{\boldsymbol{X}}_k}(\boldsymbol{d}_{k-1}) = \frac{\sqrt{E_0}}{\|\widehat{\boldsymbol{X}}_k\|}\left(I - \frac{\widehat{\boldsymbol{X}}_k\widehat{\boldsymbol{X}}_k^T}{\|\widehat{\boldsymbol{X}}_k\|^2}\right)\boldsymbol{d}_{k-1}, \tag{3.7}$$

and scaling coefficient

$$s_k = \begin{cases}\frac{\|\boldsymbol{d}_{k-1}\|}{\|\mathcal{T}_{\widehat{\boldsymbol{X}}_k}(\boldsymbol{d}_{k-1})\|}, & \text{if } \|\mathcal{T}_{\widehat{\boldsymbol{X}}_k}(\boldsymbol{d}_{k-1})\| > \|\boldsymbol{d}_{k-1}\|,\\ 1, & \text{if } \|\mathcal{T}_{\widehat{\boldsymbol{X}}_k}(\boldsymbol{d}_{k-1})\| \leq \|\boldsymbol{d}_{k-1}\|,\end{cases} \tag{3.8}$$

are used to account for the following: (1) vector addition on the manifold $\mathcal{S}$ only makes sense if the vectors live in the same tangent space, and (2) the inequality (3.8) must be satisfied for convergence of (3.4); that is the transport operator must not increase the vectors norm [49]. In all cases the iterative procedure repeats until the gradient residual

$$r = \|\boldsymbol{g}_k\| = \|\nabla\widehat{J}_k\|\sin(\theta_k), \quad \text{where} \quad \cos(\theta_k) = \frac{\langle\widehat{\boldsymbol{X}}_k, \nabla\widehat{J}_k\rangle}{\|\widehat{\boldsymbol{X}}_k\|\ \|\nabla\widehat{J}_k\|}, \tag{3.9}$$

involving the measure of the angle between the parameter vector $\widehat{\boldsymbol{X}}_k$ and Euclidean gradient $\nabla\widehat{J}_k$, becomes less than some numerically prescribed tolerance, meaning that no further progress can be made without departing too much from the constraint manifold. We note that this measure provides an immediate connection to, both the Lagrangian method (1.13) and to the Armijo condition (3.10) when $\boldsymbol{d}_k \approx -\boldsymbol{g}_k$.

3.2. **Conditions on the step size**

Convergence guarantees on the scheme (3.4) can be achieved by imposing conditions on the step-size α [48, 49]. Assuming that $\widehat{J}(\widehat{\boldsymbol{X}}_K)$ is continuously differentiable, bounded from below and that the gradient is Lipschitz continuous for all $\widehat{\boldsymbol{X}}_k \in \mathcal{S}$, the (Riemannian) strong Wolfe-conditions

$$\widehat{J}(\widehat{\boldsymbol{X}}_{k+1}(\alpha_k)) \leq \widehat{J}(\widehat{\boldsymbol{X}}_k) + c_1\alpha_k\langle \boldsymbol{g}_k, \boldsymbol{d}_k\rangle, \tag{3.10}$$

$$|\langle \boldsymbol{g}_{k+1}(\alpha_k), \mathcal{T}_{\widehat{\boldsymbol{X}}_{k+1}}(\boldsymbol{d}_k)\rangle| \leq c_2|\langle \boldsymbol{g}_k, \boldsymbol{d}_k\rangle|, \tag{3.11}$$

with $0 < c_1 < c_2 < 1/2$ ensure global convergence to a stationary point [49]. Here we have used the notation $\widehat{\boldsymbol{X}}_{k+1}(\alpha_k)$ to denote the step-size dependent choice of parameter vector and $\boldsymbol{g}_{k+1}(\alpha_k)$ to denote its corresponding tangent-gradient. In contrast to Euclidean gradient descent, the Riemannian Wolfe-conditions include an additional transport of the search direction $\mathcal{T}_{\widehat{\boldsymbol{X}}_{k+1}}(\boldsymbol{d}_k)$ to ensure that the inner product, which is now a metric on $\mathcal{S}$, remains well defined.

3.3. **Principle component analysis**

To assess the efficacy of the update procedure and the practical cost of enforcing (3.10), (3.11), we solve

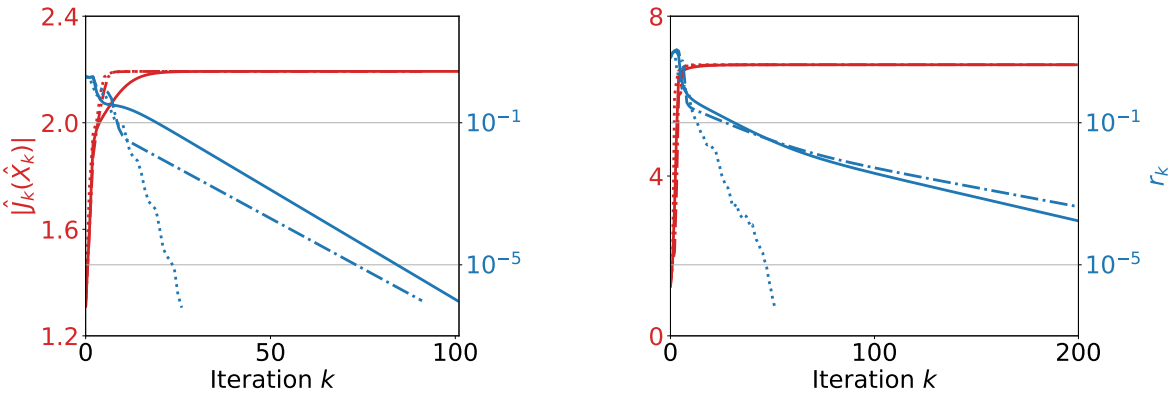
$$\begin{aligned} \min_{\boldsymbol{X}} \quad & \widehat{J}(\boldsymbol{X}) = -\frac{1}{2}\boldsymbol{X}^T M \boldsymbol{X}, \\ \text{s.t.} \quad & \|\boldsymbol{X}\|^2 = 1, \quad \text{where} \quad M = M^T, \quad \boldsymbol{X}^T M \boldsymbol{X} > 0, \end{aligned} \tag{3.12}$$

which is equivalent to principle component analysis (PCA) or if M corresponds to the discretisation of a self-adjoint linear operator, finding its leading eigenvector. Assuming an exact line-search, the rate of convergence of gradient descent to a stationary point follows analytically from the condition number $\kappa(M) = \lambda_n/\lambda_1$ of the Hessian where λ_1, λ_n refers to the smallest/largest eigenvalue of M respectively. Denoting the solution of (3.12) by $\boldsymbol{X}^*$ the result of [35] states for steepest descent

$$\widehat{J}(\widehat{\boldsymbol{X}}_{k+1}) - \widehat{J}(\boldsymbol{X}^*) \leq r^2[\widehat{J}(\widehat{\boldsymbol{X}}_k) - \widehat{J}(\boldsymbol{X}^*)], \quad \text{where} \quad r \in \left]\frac{\kappa-1}{\kappa+1}, 1\right[, \tag{3.13}$$

for large k, such that $\widehat{\boldsymbol{X}}_k \to \boldsymbol{X}^*$ at a linear rate. To test this bound we generate random $M \in \mathbb{R}^{N\times N}$ and perform gradient descent. For $N = 100$ we calculate $(\kappa(M)-1/\kappa(M)+1)^2 = 0.9799$ and using the slope of $(\widehat{J}_{k+1} - \widehat{J}^*)/(\widehat{J}_k - \widehat{J}^*) = 0.9187$, showing that the calculated estimate obeys the bound (3.13) and monotonic linear convergence is observed. Performing the same optimisation using the conjugate gradient method (3.4) subject to the strong Wolfe conditions we obtained super-linear convergence as shown in Table 3.1 and Figure 3.1.

To facilitate the application of the descent routine outlined in this section, we have combined the update formula of Section 3.1 with an implementation of an Armijo (3.10) line-search based on cubic interpolation and an implementation of a strong Wolfe condition line-search (3.11) following [41, 59]. This is available in the companion package to this paper `SphereManOpt` [36] which also contains all examples presented in this paper. In contrast to the libraries `manopt` and `pymanopt` [7, 55], our library focuses exclusively on spherical manifolds of variable radii (as desirable in fluid dynamics applications) and implements a line-search routine satisfying the strong Wolfe conditions. Such a line-search while necessary to ensure global convergence of the conjugate gradient method [48] is currently unavailable in `manopt` or `pymanopt` [7, 55].



(A) $N = 10$ (B) $N = 100$

FIGURE 3.1. Convergence history of the optimisation (3.12) using steepest-descent with an Armijo line-search (solid line), conjugate-gradient with an Armijo line-search (chained line) and conjugate-gradient with a Wolfe I & II line-search (dotted line) method. The absolute value of the objective function $|\widehat{J}_k|$ is shown in red and the gradient residual r_k in blue. Despite conducting a more expensive line-search the conjugate-gradient method, which exhibits super-linear convergence, ultimately requires fewer iterations to converge (see Table 3.1).

TABLE 3.1. Comparison of the steepest descent and conjugate gradient methods applied to (3.12). We set the max step-size $\alpha_{\max} = 1$ and terminate optimisation when $r \leq$ 1e-06 or when iterations $>$ 200. Star $*$ denotes a stalled run. Despite conducting a more expensive line-search the conjugate-gradient routine converges super-linearly thus requiring fewer evaluations as shown in Figure 3.1.

Routine	Size N	line-search	Obj evals $\widehat{J}$	Grad evals $\nabla\widehat{J}$	Residual r
Gradient descent	10	Armijo	105	102	9.2117e-07
Conjugate gradient	10	Armijo	95	92	9.6906e-07
Conjugate gradient	10	Wolfe I & II	51	34	6.2045e-07
Gradient descent	100	Armijo	205	200	*2.9087e-05
Conjugate gradient	100	Armijo	205	200	*1.5407e-04
Conjugate gradient	100	Wolfe I & II	94	66	7.4820e-07

4. Numerical performance

Having outlined the calculation of a consistent gradient estimate in Section 2 and a line-search following [48, 49] with global convergence to a stationary point in Section 3, we now combine the developments of these sections and apply them to solving for the optimal solution of the examples previously considered in Section 2.

4.1. Swift-Hohenberg multi-stability

For the choice of parameters given in Section 2 the Swift-Hohenberg equation has four stable equilibria. Following [33], we find the smallest perturbation u_0 of energy M_2 which, when added to the zero-energy state O, triggers a transition to the next least-energy state u_T of energy S_2. (In practice, M_2 is determined by repeatedly computing the optimal u_0, of gradually decreasing energy M, that maximises

the energy growth by target time T. Since larger energies characterize the searched-for transition, M_2 corresponds to the smallest value of M for which the optimal u_0 still triggers the transition.)

Figure 4.1 compares amplitudes of the perturbation M_2 and stable equilibrium S_2 with the results of [33]. All steady equilibrium are correctly identified to four digits of precision while the perturbation amplitudes are correct to one digit. The differences with the result of [33] are explained by the fact we performed a small number of runs initiated from random noise, to probe a cost functional with numerous local minima. In contrast they perform several hundred runs so as to explore many possible basins of attraction.

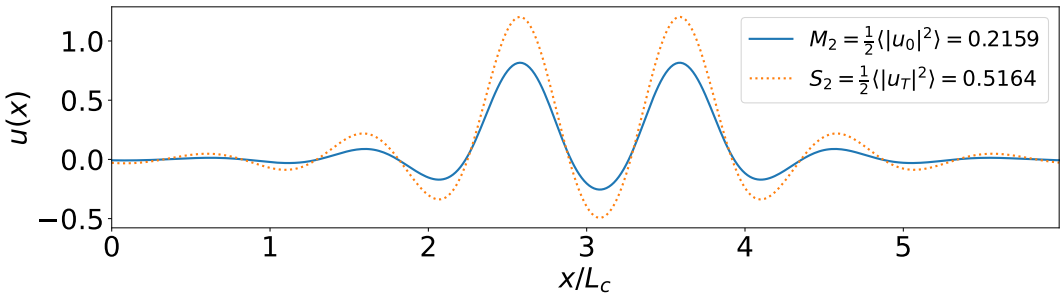


Figure 4.1. Optimisation using the discrete gradient recovers the energies of the initial condition M_2 and equilibrium state S_2 (inset). These are given in Figures 1 and 5 of [33] as $M_2, S_2 = (0.2048, 0.5164)$.

4.1.1. *Discrete vs. continuous adjoint*

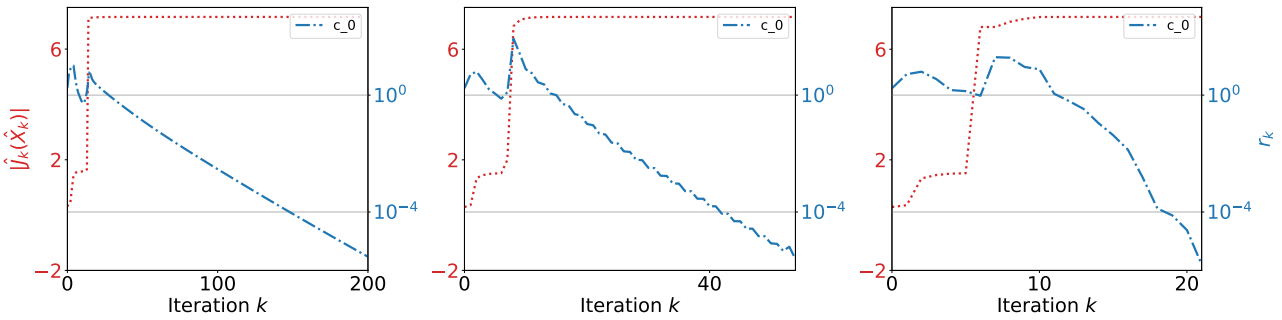
Having established that our code faithfully reproduces the results of [33] we now compare the performance of the discrete and continuous gradient when optimising for the perturbation u_0. The convergence of these runs are reported in rows (a), (c) of Table 4.1. Both runs use steepest-descent (SD) with an Armijo line-search and are initiated using random noise. While the continuous gradient seems to converge faster to a small residual, this appearance is in fact misleading. By running the optimum (found with the "continuous", i.e. differentiate-then-discretize approach) once through the direct and discrete adjoint solvers (i.e. discretize-then-differentiate), we obtain the corresponding "discrete residual" which is in fact much higher. This is highlighted in row (a) of Table 4.1 where the discrete residual is 6 orders of magnitude greater that the continuous one, showing that only the discrete gradient can accurately yield convergence to a stationary point. A further disadvantage of the continuous gradient as shown in row (b) of Table 4.1 is that when supplied to the conjugate-gradient (CG) routine it stalls, thus eliminating the possibility of super-linear convergence.

4.1.2. *Line-search algorithms*

With the superiority of the discrete gradient established we now consider the efficiency and convergence behaviour of three different descent routines. The convergence of each routine is reported in rows (c)–(f) of Table 4.1 and frames (c)–(f) of Figure 4.2. While all routines reach a small residual error, the most commonly used routine, a backtracking line-search accompanied by an Armijo condition, performs the worst. As anticipated the conjugate-gradient method implementing the strong Wolfe conditions and thus enjoying super-linear convergence performs best.

Table 4.1. Test summary and numerical cost of reaching a stationary point using different descent routines as shown in Figure 4.2. We set the max step-size $\alpha_{\max} = \pi$ and terminate optimisation when $r \leq$ 1e-06 or when iterations > 200. Star $*$ denotes a stalled run. In rows (a)–(c) we have reported the `continuous residual` as well as the discrete residual.

Routine	Gradient	line-search	Obj evals $\widehat{J}(\widehat{\boldsymbol{u}}_0)$	Grad evals $\frac{\delta\mathcal{L}}{\delta \boldsymbol{u}_0}$	Residual r
(a) SD	Continuous	Armijo	163	173	1.5190/`9.2551e − 07`
(b) CG	Continuous	Wolfe I & II	63	37	*1.4662/`1.1718e − 01`
(c) CG	Continuous	Armijo	217	200	*3.7429/`2.8598`
(d) SD	Discrete	Armijo	213	197	2.9400e-06
(e) SD	Discrete	Wolfe I & II	118	92	2.4488e-06
(f) CG	Discrete	Armijo	29	13	*3.8577e+01
(g) CG	Discrete	Wolfe I & II	91	44	1.6003e-06

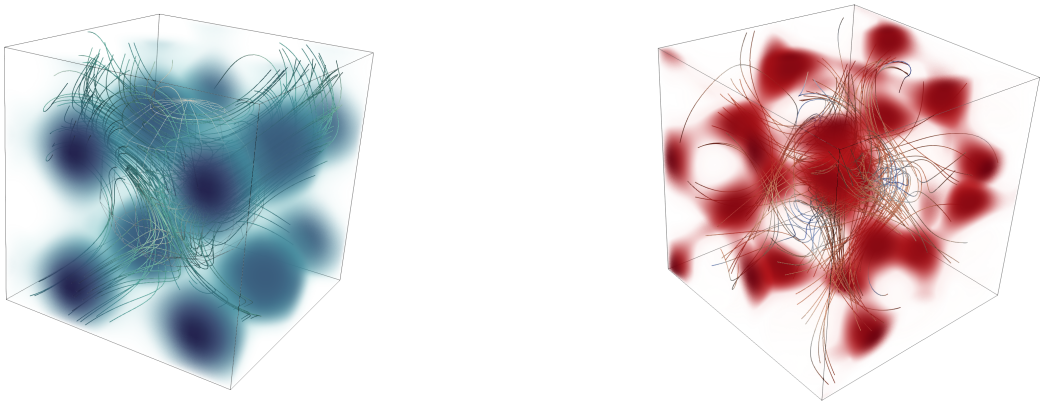


(d) SD (Armijo) (e) SD (Wolfe I & II) (g) CG (Wolfe I & II)

Figure 4.2. Convergence of the optimisation runs in rows (d)–(g) of Table 4.1, in terms of the cost functional J (dotted, red) and gradient residual r (chained, blue), to the perturbation M_2 when initiated with random noise of the same amplitude. The steepest descent (SD) methods (d), (e) converge linearly while the conjugate-gradient (CG) method (g) super-linearly, thus demanding fewer iterations.

4.2. **Kinematic dynamo**

Having established the validity of the gradient in 2.2, where only the discrete adjoint gives a consistent estimate, and similarly the validity of our code by comparison with [58](see Figure 4.3), we now first demonstrate the superiority of the discrete gradient when optimising for fields $\boldsymbol{U}$ and $\boldsymbol{B}_0$. In contrast to the previous examples this demands choosing a single step-size α_k to simultaneously update all n components of the parameter vector $\widehat{\boldsymbol{X}}_k$, whilst retaining a termination condition based on the residual r_k^i of each component. We argue that this is preferable to taking a summed quantity such as $r_k = \sum_{i=0}^{n} r_k^i$ which could lead to over and under converged directions compensating one-another. One weakness of the outlined approach is that when different components of $\widehat{\boldsymbol{X}}_k$ converge at different rates, we will limit the convergence rate by restricting ourselves to a single step-size. Preconditioning the update step presents a possible solution to this problem [21].



(A) Magnetic field $\boldsymbol{B}_T$ (B) Velocity field $\boldsymbol{U}$

FIGURE 4.3. The optimal (a) magnetic and (b) velocity field for $Rm = T = 1$ which yield $\langle \boldsymbol{B}_T^2 \rangle = 0.4329$. To be compared with Figures 1 and 2 of [58]. This optimum was calculated using an Armijo based line-search corresponding to row (c) Table 4.2.

4.2.1. *Discrete vs. continuous adjoint*

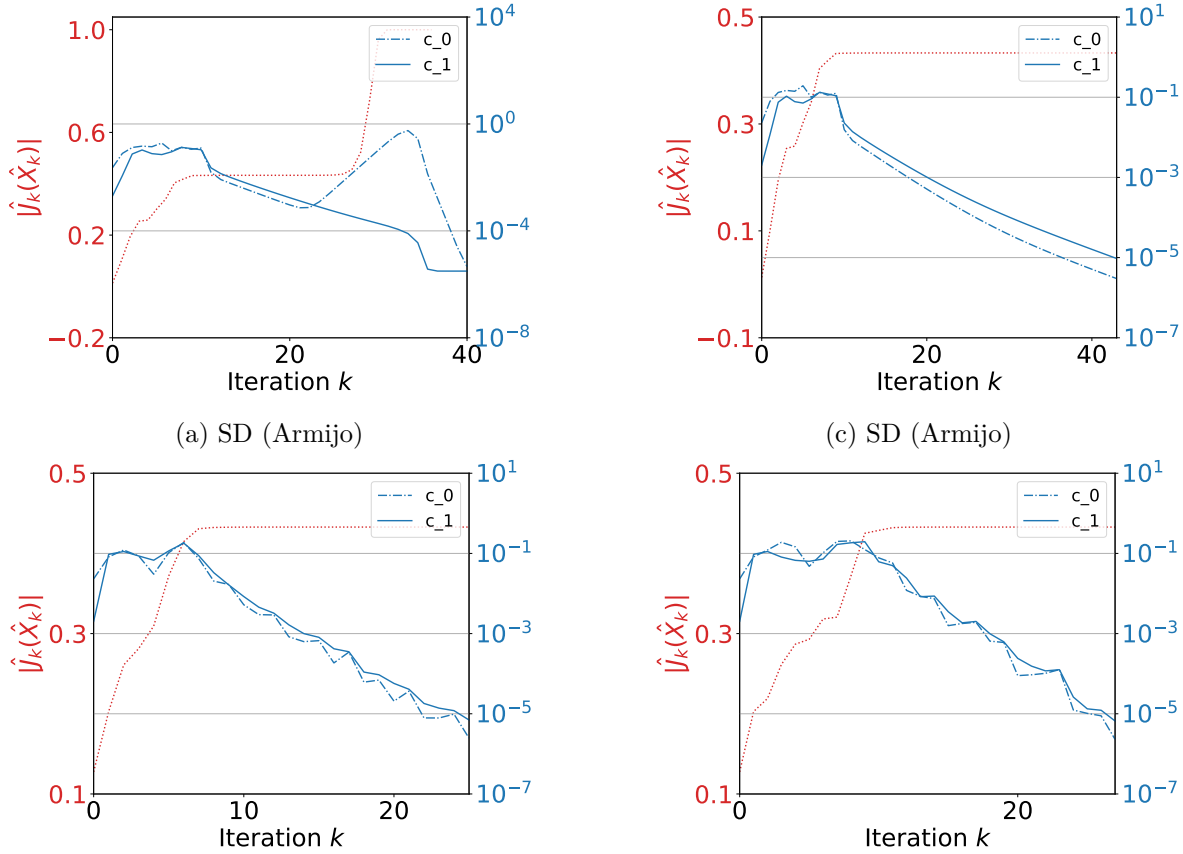
Rows (a) and (c) of Table 4.2 compare the performance of the continuous and discrete gradient using steepest descent (SD) combined with an Armijo line-search. While the continuous gradient appears to converge in fewer gradient/function evaluations, we find that upon diverging after ~ 25 iterations it does so to a spurious solution characterised by a non-zero magnetic divergence and net magnetic flux (Figure 4.4(a)). The optimisation using the discrete gradient by contrast converges monotonically to the desired convergence criteria Figure 4.4(c). As previous it was not possible to avail of the conjugate-gradient (CG) method when using a continuous gradient (row (b) in Table 4.2).

TABLE 4.2. Numerical cost of reaching a stationary point using different descent routines as shown in Figure 4.4. We set $\alpha_{\max} = 100$ and terminate optimisation when $r_k^i \leq$ 1e-06 or when iterations > 200. Star $*$ denotes a stalled run, while $**$ a non-physical solution. All rows are given in terms of their discrete gradient residual.

Routine	Gradient	Line-search	evals $\widehat{J}(\widehat{\boldsymbol{B}}_0, \widehat{\boldsymbol{U}})$	evals $\frac{\delta \mathcal{L}}{\delta \boldsymbol{U}}, \frac{\delta \mathcal{L}}{\delta \boldsymbol{B}_0}$	r_k^B	r_k^U
(a) SD	Continuous	Armijo	42	37	**1.2519e-05	**3.0875e-06
(b) CG	Continuous	Wolfe I & II	2	7	*2.3210e-02	*2.0199e-03
(c) SD	Discrete	Armijo	48	44	2.9980e-06	9.4994e-06
(d) SD	Discrete	Wolfe I & II	53	53	2.4662e-06	6.9963e-06
(e) CG	Discrete	Armijo	55	42	3.1154e-06	9.7692e-06
(f) CG	Discrete	Wolfe I & II	61	57	2.3031e-06	6.6113e-06

4.2.2. *Line-search algorithms*

Having established the necessity of the discrete gradient, we now consider the efficiency of the strong Wolfe line-search and conjugate-gradient routines when using a combined update approach. Columns (d) and (f) of Table 4.2 repeat the same calculation using a strong Wolfe line-search with and without the conjugate-gradient update. While both routines converge in fewer iterations when comparing



(a) SD (Armijo)

(c) SD (Armijo)

(d) SD (Wolfe I & II)

(f) CG (Wolfe I & II)

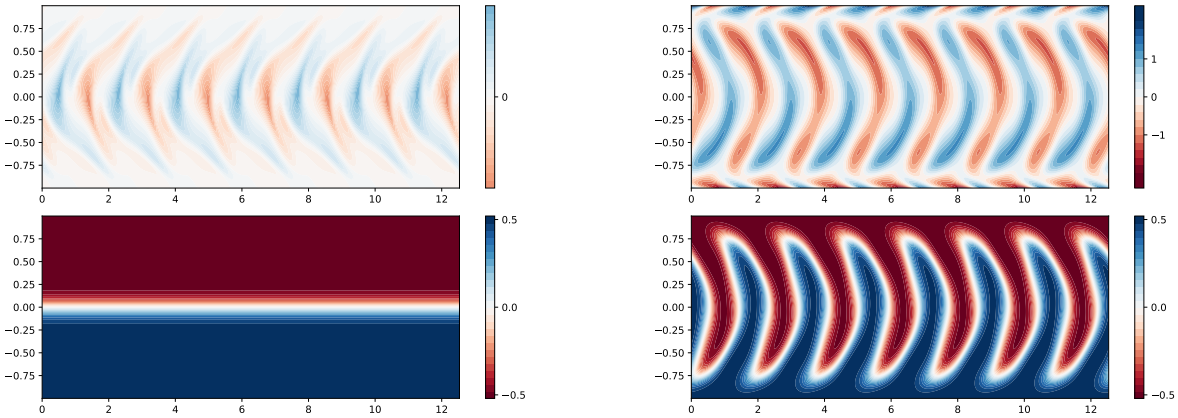
FIGURE 4.4. Convergence history in terms of the residual errors r_k^B (shown as c_0) and r_k^U (shown as c_1). Optimisation (a) used a continuous gradient and (c)–(f) a discrete gradient while their routines and convergence cost are summarised in Table 4.2. While the continuous gradient appears to converge faster it is in fact to a spurious solution with non-zero divergence and net magnetic flux. Routines implementing the Wolfe conditions converge in fewer iterations but their line-search procedures are slightly more costly.

Figure 4.4(c) with Figures 4.4(d)(f) their total number of function and gradient evaluations is however slightly larger.

4.3. **Optimal Mixing**

The previous examples demonstrate that our gradient descent algorithm efficiently achieves a high degree of convergence, but that this is possible only when using the discrete gradient. We now combine all this paper's developments to treat a PDE discretised using a combined Fourier/Chebyshev basis, as is required to solve the optimal mixing problem (2.16). As of yet adjoint optimisation studies with a discrete gradient have used finite difference [57], finite element [40, 60] or finite volume methods [51] and in doing so encountered some difficulties in treating boundary conditions and pressure gauge conditions. By using the Dedalus code which treats the pressure and divergence free condition (and similar integral conditions) alongside all other variables during a single time-step no splitting methods

or complications arise in our implementation (A). Given the prevalence of flow features with large gradients such as velocity boundary layers or density fronts, the accurate solution of control problems using a Chebyshev pseudospectral method is an essential step.



(A) b_0 (bottom) and $\nabla \times \boldsymbol{u}_0$ (top).

(B) b_T (bottom) and $\nabla \times \boldsymbol{u}_T$ (top).

FIGURE 4.5. The optimal (a) initial velocity perturbation (given in terms of the vorticity $\nabla \times \boldsymbol{u}_0$) and (b) the resulting deviation density and perturbation velocity fields at $T = 5$ for $Ri = 0.05$. To be compared with Figure 2 of [37]. This optimum was calculated using a Wolfe based line-search and conjugate gradient method corresponding to Table 4.3(f).

Figure 4.5(a) shows the sharp initial density deviation b_0 and optimal velocity perturbation (here plotted as the vorticity $\nabla \times \boldsymbol{u}_0$), which as shown Figure 4.5(b) allows the mixing process to unfold via Taylor dispersion [37]. These figures, computed by solving (2.16) using the discrete gradient and conjugate gradient method accompanied by a Wolfe line-search (Table 4.3(f)), also demonstrate that our implementation faithfully reproduces the results of [37].

4.3.1. *Discrete vs. continuous adjoint*

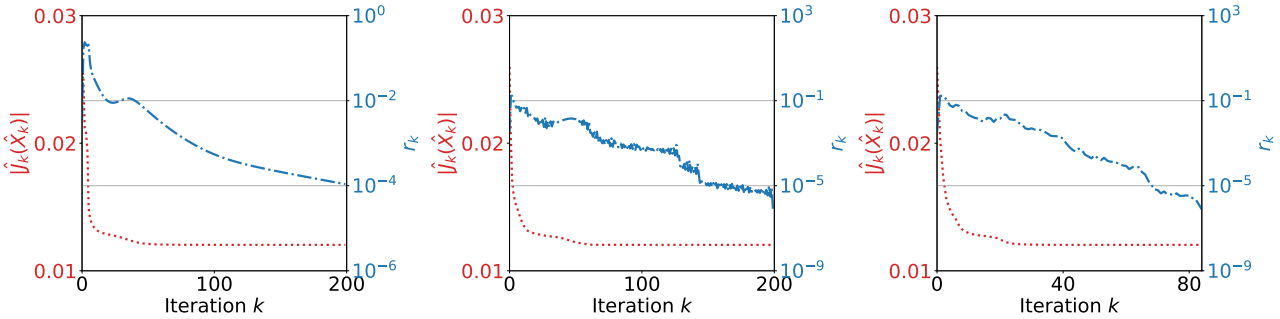
With our code's validity established we now compare the performance of the discrete and continuous gradient when optimising for the perturbation $\boldsymbol{u}_0$. The convergence of these runs reported in Table 4.3, establish that only the discrete gradient can satisfy the imposed convergence criteria. Although the conjugate gradient method using the continuous gradient works well initially, it subsequently stalls due to a poor gradient estimate. While both gradient estimates choose a dominant wavelength $k_x = 7$ the discrete gradient consistently achieves a slightly better optimum than the continuous gradient.

4.3.2. *Line-search algorithms*

As shown in Figure 4.6 and Table 4.3 the discrete gradient allows us to use the conjugate gradient algorithm and obtain fast convergence. Combining the discrete gradient with either a conjugate gradient method (e) or a strong Wolfe line-search (f) the imposed convergence criteria can also be satisfied, albeit less efficiently. Comparing row (e) with (f) and row (c) with (d), it is also apparent that the strong Wolfe line-search greatly influences convergence despite its greater cost.

TABLE 4.3. Test summary and numerical cost of reaching a stationary point using different descent routines as shown in Figures 4.5 and 4.6. We set $\alpha_{\max} = 100$ and terminate optimisation when $r_k^i \leq$1e-06 or iterations > 200. Star $*$ denotes a stalled run. Rows (a), (b) report the `continuous residual` and (c)–(f) the discrete residual.

Routine	Gradient	Line-search	evals $\widehat{J}(\boldsymbol{u}, b)$	evals $\frac{\delta \mathcal{L}}{\delta \boldsymbol{u}_0}$	r_k	$\widehat{J}(\boldsymbol{u}, b)$
(a) SD	Continuous	Armijo	209	200	`4.9943e-04`	1.2033e-02
(b) CG	Continuous	Wolfe I & II	65	61	*`5.5593e-04`	*1.2032e-02
(c) SD	Discrete	Armijo	209	200	1.0852e-04	1.2030e-02
(d) SD	Discrete	Wolfe I & II	258	248	7.5664e-07	1.2029e-02
(e) CG	Discrete	Armijo	209	200	9.5897e-07	1.2030e-02
(f) CG	Discrete	Wolfe I & II	123	116	7.6475e-07	1.2029e-02



(c) SD (Armijo) (d) SD (Wolfe I & II) (f) CG (Wolfe I & II)

FIGURE 4.6. Convergence of the optimisation runs in rows (c)–(f) of Table 4.3, in terms of the cost function J (dotted, red) and gradient residual r (chained, blue), to the perturbation $\boldsymbol{u}_0$ when initiated with random noise. The steepest descent (SD) methods (c), (d) converge sub-linearly while the conjugate-gradient (CG) method (f) at least linearly, thus demanding fewer iterations.

4.4. **Maximum step-size**

For each problem considered we have prescribed a hyper-parameter $\alpha_{\max}$. For an Armijo line-search it is the initial step-size $\alpha_0 = \alpha_{\max}$ from which back-tracking begins, while in a strong Wolfe line-search this constitutes the maximum admissible step-size. By reducing or increasing the number of iterations required to achieve convergence, this parameter was found to have a substantial impact on performance, in all problems considered. Typically as was the case in the PCA and Swift-Hohenberg problems setting $\alpha_{\max} \approx 1$ works well. In the kinematic dynamo and optimal mixing problems however we found that the range of step sizes considered by the line-search must be larger, particularly when using the strong Wolfe-conditions $(\alpha_{\min}, \alpha_{\max}) = (10^{-6}, 10^2)$; although steepest descent was also found to converge in 20-40 fewer iterations when a large maximum/initial step-size was chosen for these problems. We understand this behaviour to be a consequence of $\widehat{J}(\widehat{\boldsymbol{X}}_{k+1})$ being (informally speaking) *poorly bounded* during initial iterations due to the unfavourable initial dynamo velocity field $\boldsymbol{U}$ or mixing perturbation velocity field $\boldsymbol{u}$. Consequently this requires selecting a larger than usual α_k for the Armijo condition (3.10) to become active and for (3.11) be easily satisfied. A further discussion of this issue is given in the introduction of [41].

5. Conclusion

In this paper we have focused on inconsistencies arising from the gradient estimation and application of equality constraints in the numerical direct adjoint-based control procedure [28]; to show why these issues occur and how they can be resolved on practical examples. Deriving the discrete adjoint equations of time-dependent PDE constraints for (preconditioned) multistep schemes [3], and considering three example problems we demonstrated in Section 2 a consistent gradient approximation, readily obtainable with minor code modifications. In Section 3 we recalled following [6, 48] how an equality constrained problem can be solved as an unconstrained problem on a spherical manifold. Considering three examples, we numerically demonstrate that this procedure, available as `SphereManOpt` [36], is capable of giving robust linear to super-linear convergence to stationary points with small residual error when both nonlinear PDE and multiple norm constraints are present. Useful extensions of this work would consider the discrete adjoint of Runge-Kutta schemes, preconditioning schemes for problems with multiple norm constraints and exploiting additional geometric structures of the optimisation problems considered. Further developments of `SphereManOpt` would include L_p norms and the integration of an optimal checkpointing strategy [19].

Appendix A. Preconditioned first order scheme

Departing from (2.2) we write a general *s-step* multistep IMEX method as

$$(a_0\mathcal{M}_p + b_0\mathcal{L}_p)\widehat{\boldsymbol{X}}_p^n = \sum_{j=1}^{2} c_j\widehat{\mathcal{F}}_p^{n-j} - a_j\mathcal{M}_p\widehat{\boldsymbol{X}}_p^{n-j} - b_j\mathcal{L}_p\widehat{\boldsymbol{X}}_p^{n-j}, \tag{A.1}$$

where $\widehat{\boldsymbol{X}} = T\boldsymbol{X}$ denotes the Chebyshev transform of the first kind and the matrix operators $\mathcal{M}, \mathcal{L}$ are assumed to be in spectral space such that $\widehat{\boldsymbol{X}}$ corresponds to a vector of Fourier modes and Chebyshev polynomial coefficients. For the purposes of parallelisation, the previous equation is separated into pencils i.e. equations separable in Fourier modes such that only different Chebyshev modes couple. In 2D for example the number of pencils corresponds exactly to the number of Fourier modes divided by two, owning to a complex to real conversion. We thus consider the solution procedure for individual pencils as indexed by subscript p in (A.1), where only the right hand side couples different Fourier/transverse modes.

To efficiently solve (A.1) Dedalus employs preconditioners P_p^L, P^R with the former depending on the pencil considered. By using: a first order formulation, converting from Chebyshev polynomials of the first kind $T_n(z)$ to Chebyshev polynomials of the second kind $U_n(z)$, performing Dirichlet recombination and grouping by Chebyshev modes of the second kind rather than variables, all left hand side (LHS) matrices are rendered sparse [8]. In practice these preconditioners are employed as

$$\underbrace{(a_0\overline{\mathcal{M}_p} + b_0\overline{\mathcal{L}_p})}_{A_p}\underbrace{(P^R)^{-1}\widehat{\boldsymbol{X}}_p^n}_{\widehat{\boldsymbol{Y}}_p^n} = \underbrace{P_p^L\sum_{j=1}^{s} c_j\widehat{\mathcal{F}}_p^{n-j} - a_j\mathcal{M}_p\widehat{\boldsymbol{X}}_p^{n-j} - b_j\mathcal{L}_p\widehat{\boldsymbol{X}}_p^{n-j}}_{B_p}, \tag{A.2}$$

with $\overline{\mathcal{M}_p} = P_p^L\mathcal{M}_pP^R$ to allow for a sparse inversion of the banded system

$$\widehat{\boldsymbol{Y}}_p^n = A_p^{-1}B_p, \quad \widehat{\boldsymbol{X}}_p^n = (P^R)\widehat{\boldsymbol{Y}}_p^n. \tag{A.3}$$

Similarly $\widehat{\boldsymbol{X}}_p^n$ is easily recovered by left multiplication using P^R. Using the recurrence relation $2T_n(z) = U_n(z) - U_{n-2}(z)$ we have a relation between the two sets of polynomials. To convert between them we use the left preconditioner P_p^L. The above constitutes the solution method followed for the optimal mixing problem (D.1) and what follows the method for computing its adjoint equations.

To solve (2.6) to (2.9) using this formulation, we retain only the first order coefficients denoted by a tilde, multiply by the corresponding left preconditioner P_p^L, insert the right preconditioner and its inverse so as to form A_p and finally form their corresponding adjoint equations at indices $n = N$, $1 \leq n \leq N-1$ and $n = 0$ respectively

$$(a_0\overline{\mathcal{M}_p^\dagger} + b_0\overline{\mathcal{L}_p^\dagger})\widehat{\boldsymbol{q}}_p^N = (P^R)^\dagger \frac{\partial J}{\partial \widehat{\boldsymbol{X}}_N}, \tag{A.4}$$

$$(a_0\overline{\mathcal{M}_p^\dagger} + b_0\overline{\mathcal{L}_p^\dagger})\widehat{\boldsymbol{q}}_p^n = -\left[(\tilde{a}_1\overline{\mathcal{M}_p^\dagger} + \tilde{b}_1\overline{\mathcal{L}_p^\dagger}) - (P^R)^\dagger \tilde{c}_1 \frac{\partial F}{\partial \boldsymbol{X}_n}^\dagger (P_p^L)^\dagger\right]\widehat{\boldsymbol{q}}_p^{n+1} + (P^R)^\dagger \frac{\partial J}{\partial \widehat{\boldsymbol{X}}_n} \tag{A.5}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{X}(t=0)} \equiv \widehat{\boldsymbol{q}}_p^0 = -\left[(\tilde{a}_1\mathcal{M}^\dagger + \tilde{b}_1\mathcal{L}^\dagger) - \tilde{c}_1 \frac{\partial F}{\partial \boldsymbol{X}_0}^\dagger\right](P_p^L)^\dagger \boldsymbol{q}_1 + \frac{\partial J}{\partial \widehat{\boldsymbol{X}}_0}. \tag{A.6}$$

The derivation shows that when implemented in Dedalus or an alternative code that avails of preconditioning matrices an additional source of error arises from the non-commutative nature of the Hermitian adjoint operation. To compute the variation of the objective function requires computing the adjoint of the Chebyshev transforms to see why this arises we consider the $\mathcal{O}(\Delta t)$ accurate time integration of the objective function

$$J(\widehat{\boldsymbol{X}}) = \sum_{n=0}^{N} \frac{1}{2V}\langle T^{-1}\widehat{\boldsymbol{X}}_n, WT^{-1}\widehat{\boldsymbol{X}}_n\rangle \Delta t, \tag{A.7}$$

where $W = W^\dagger$. Taking variations of the above we obtain

$$\left\langle \delta\widehat{\boldsymbol{X}_n}, \frac{\partial J}{\partial \widehat{\boldsymbol{X}}_n} \right\rangle = \left\langle \delta\widehat{\boldsymbol{X}}_n, T^{-\dagger}WT^{-1}\widehat{\boldsymbol{X}}_n \frac{\Delta t}{V} \right\rangle. \tag{A.8}$$

Appendix B. Discrete adjoint of the Swift-Hohenberg problem

Writing the optimisation problem (2.15) as a minimisation problem in terms of the Lagrangian

$$\mathcal{L}(u, u^\dagger) = -\int_{t=0}^{T} \langle u(x,t), u(x,t)\rangle dt - \int_{t=0}^{T} \langle u^\dagger, \mathcal{M}\partial_t u + \mathcal{L}u - \mathcal{F}(u,t)\rangle dt, \tag{B.1}$$

where $\langle f, g\rangle = \int_x fg\, dx$, $\mathcal{M} = \mathbb{I}, \mathcal{L} = (1+\partial_x^2)^2 - a$ and $\mathcal{F}(u,t) = 1.8u^2 - u^3$. The adjoint equations and compatibility conditions corresponding to (B.1) give

$$-\partial_t u^\dagger + (1+\partial_x^2)^2 u^\dagger - a u^\dagger = (3.6u - 3u^2)u^\dagger - 2u, \qquad u^\dagger(x,T) = 0. \tag{B.2}$$

Letting $\boldsymbol{u}(t)$ denote the spatial discretisation of $u(x,t)$ and $\tilde{\boldsymbol{u}} = (\boldsymbol{u}_0, \cdots, \boldsymbol{u}_N)$ its temporal discretisation such that $t_n = n\Delta t, 0 \leq n \leq N$, its $\mathcal{O}(\Delta t)$ accurate discrete counterpart is given by

$$(\tilde{a}_0\mathcal{M} + \tilde{b}_0\mathcal{L})\boldsymbol{u}_N^\dagger = -2\boldsymbol{u}_N, \tag{B.3}$$

$$(\tilde{a}_0\mathcal{M} + \tilde{b}_0\mathcal{L})\boldsymbol{u}_n^\dagger + (\tilde{a}_1\mathcal{M} + \tilde{b}_1\mathcal{L})\boldsymbol{u}_{n+1}^\dagger - \tilde{c}_1 \frac{\partial \mathcal{F}}{\partial \boldsymbol{u}}^\dagger |_n \boldsymbol{u}_{n+1}^\dagger = -2\boldsymbol{u}_n, \tag{B.4}$$

$$\frac{\boldsymbol{u}_0^\dagger}{\Delta t} - (\tilde{a}_1\mathcal{M} + \tilde{b}_1\mathcal{L})\boldsymbol{u}_1^\dagger - \tilde{c}_1 \frac{\partial \mathcal{F}}{\partial \boldsymbol{u}}^\dagger |_0 \boldsymbol{u}_1^\dagger = -2\boldsymbol{u}_0, \tag{B.5}$$

for $n = N, 1 \leq n \leq N-1, n = 0$ respectively, where $\frac{\partial \mathcal{F}}{\partial \boldsymbol{u}}^\dagger |_n \boldsymbol{u}_{n+1}^\dagger = (3.6\boldsymbol{u}_n - 3\boldsymbol{u}_n^2)\boldsymbol{u}_{n+1}^\dagger$.

Appendix C. Discrete adjoint of the kinematic dynamo problem

The continuous Lagrangian corresponding to (2.12) is given by

$$\begin{aligned}\mathcal{L} = \langle \boldsymbol{B}_T^2 \rangle - \int_0^T \langle \boldsymbol{B}^\dagger, \frac{\partial \boldsymbol{B}}{\partial t} - \nabla \times (\boldsymbol{U} \times \boldsymbol{B}) - \nabla \Pi - Rm^{-1} \nabla^2 \boldsymbol{B} \rangle dt \\ - \int_0^T \langle \Pi^\dagger, (\nabla \cdot \boldsymbol{B}) \rangle dt - \int_0^T \langle P^\dagger, (\nabla \cdot \boldsymbol{U}) \rangle dt,\end{aligned} \tag{C.1}$$

where $\boldsymbol{B}^\dagger, \Pi^\dagger, P^\dagger$ are the adjoint variables enforcing the constraint equations for the adjoint variables. Taking variations with respect to the primal variables we obtain

$$\begin{aligned}\frac{\delta \mathcal{L}}{\delta \boldsymbol{B}} &= \frac{\partial \boldsymbol{B}^\dagger}{\partial t} + Rm^{-1} \nabla^2 \boldsymbol{B}^\dagger + \nabla \Pi^\dagger + (\nabla \times \boldsymbol{B}^\dagger) \times \boldsymbol{U} = 0 \\ \frac{\delta \mathcal{L}}{\delta \boldsymbol{\Pi}} &= \nabla \cdot \boldsymbol{B}^\dagger = 0, \\ \frac{\delta \mathcal{L}}{\delta \boldsymbol{B}_0} &= \boldsymbol{B}_0^\dagger, \\ \frac{\delta \mathcal{L}}{\delta \boldsymbol{B}_T} &= 2\boldsymbol{B}_T - \boldsymbol{B}_T^\dagger = 0, \\ \frac{\delta \mathcal{L}}{\delta \boldsymbol{U}} &= \int_0^T \left[\boldsymbol{B} \times (\nabla \times \boldsymbol{B}^\dagger) + \nabla P^\dagger \right] dt,\end{aligned} \tag{C.2}$$

for which the last line can be replaced by

$$\frac{\partial}{\partial t} \left(\frac{\delta \mathcal{L}}{\delta \boldsymbol{U}} \right) = \boldsymbol{B} \times (\nabla \times \boldsymbol{B}^\dagger) + \nabla P^\dagger, \quad \nabla \cdot \frac{\delta \mathcal{L}}{\delta \boldsymbol{U}} = 0, \tag{C.3}$$

to ensure a divergence free gradient. The discrete counterpart of (C.1) is given by

$$\begin{aligned}\mathcal{L} = &\langle \boldsymbol{X}_N, \mathcal{M} \boldsymbol{X}_N \rangle - \langle \boldsymbol{q}_0, \boldsymbol{X}_0 - \boldsymbol{X}(\boldsymbol{x}, t=0) \rangle \\ &- \sum_{n=0}^{N-1} \langle \boldsymbol{q}_{n+1}, \mathcal{M}(\tilde{a}_0 \boldsymbol{X}_{n+1} + \tilde{a}_1 \boldsymbol{X}_n) + \mathcal{L}(\tilde{b}_0 \boldsymbol{X}_{n+1} + \tilde{b}_1 \boldsymbol{X}_n) - \tilde{c}_1 \underbrace{(\nabla \times (\boldsymbol{U} \times \boldsymbol{B}_n),\ 0)}_{\mathcal{F}(\boldsymbol{X}_n)} \rangle \Delta t,\end{aligned} \tag{C.4}$$

where the vectors and operators are given by

$$\boldsymbol{X}_n = (\boldsymbol{B}, \Pi)_n^T, \quad \boldsymbol{q}_n = (\boldsymbol{B}^\dagger, \Pi^\dagger)_n, \quad \mathcal{M} = diag(\boldsymbol{I}, 0), \quad \mathcal{L} = \begin{pmatrix} -Rm^{-1}\boldsymbol{\Delta} & \nabla \\ \nabla^T & 0 \end{pmatrix}. \tag{C.5}$$

Taking variations with respect to $\boldsymbol{X}_n$ we recover the discrete adjoint equations

$$(\tilde{a}_0 \mathcal{M} + \tilde{b}_0 \mathcal{L}) \boldsymbol{q}_N = \Delta t^{-1} 2 \mathcal{M} \boldsymbol{X}_N, \tag{C.6}$$

$$(\tilde{a}_0 \mathcal{M} + \tilde{b}_0 \mathcal{L}) \boldsymbol{q}_n + (\tilde{a}_1 \mathcal{M} + \tilde{b}_1 \mathcal{L}) \boldsymbol{q}_{n+1} - \tilde{c}_1 \frac{\partial \mathcal{F}}{\partial \boldsymbol{X}_n}^\dagger \boldsymbol{q}_{n+1} = 0, \tag{C.7}$$

$$\boldsymbol{q}_0 \Delta t^{-1} - (\tilde{a}_1 \mathcal{M} + \tilde{b}_1 \mathcal{L}) \boldsymbol{q}_1 - \tilde{c}_1 \frac{\partial \mathcal{F}}{\partial \boldsymbol{X}_n}^\dagger \boldsymbol{q}_1 = 0, \tag{C.8}$$

for $n = N, N-1 \geq n \geq 1, n = 0$ respectively, where

$$\frac{\partial \mathcal{F}}{\partial \boldsymbol{X}_n}^\dagger \boldsymbol{q}_{n+1} = \left((\nabla \times \boldsymbol{B}_{n+1}^\dagger) \times \boldsymbol{U},\ 0 \right). \tag{C.9}$$

In the case where $\boldsymbol{U}$ is allowed to be updated, we take variations with respect to $\boldsymbol{U}$ directly in (C.4) in order to obtain the additional equation

$$\frac{\delta\mathcal{L}(\boldsymbol{X}_N, \boldsymbol{q}_N)}{\delta \boldsymbol{U}} = \sum_{n=0}^{N-1} [-\tilde{c}_1 \boldsymbol{B}_n \times (\nabla \times \boldsymbol{B}^\dagger_{n+1}) + \nabla P^\dagger_{n+1}] \Delta t, \tag{C.10}$$

which is time-stepped by re-writing the above as

$$(\tilde{a}_0 \mathcal{M} + \tilde{b}_0 \mathcal{L}) \boldsymbol{w}_n + (\tilde{a}_1 \mathcal{M} + \tilde{b}_1 \mathcal{L}) \boldsymbol{w}_{n+1} = -\tilde{c}_1 \left(\boldsymbol{B}_n \times (\nabla \times \boldsymbol{B}^\dagger_{n+1}),\ 0 \right), \tag{C.11}$$

where $\boldsymbol{w}_n = (\frac{\delta \mathcal{L}}{\delta \boldsymbol{U}},\ P^\dagger)_n$ and $\mathcal{L}$ is as above but with $Rm^{-1} = 0$ such that $\nabla P^\dagger$ is used to enforce $\nabla \cdot \frac{\boldsymbol{\delta \mathcal{L}}}{\boldsymbol{\delta U}} = 0$. By enforcing both divergence free constraints at all times using the Lagrange multipliers $\Pi, \Pi^\dagger, P^\dagger$ we eliminates the need for projection steps including Chorin's method which can complicate gradient computations [10]. In all cases we apply the gauge condition that the zero'th Fourier mode of each field $f_{k_x=0,k_y=0,k_z=0} = 0$.

Appendix D. Discrete adjoint of the optimal mixing problem

The continuous Lagrangian corresponding to (2.16) is given by

$$\begin{aligned} \mathcal{L} = \mathcal{J}(\boldsymbol{u}, b) - \int_0^T \left\langle \boldsymbol{u}^\dagger, \frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{U}_0 \cdot \nabla)\boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)[\boldsymbol{U}_0 + \boldsymbol{u}] - \nabla p + Rib\widehat{\boldsymbol{z}} - \frac{1}{Re}\nabla^2 \boldsymbol{u} \right\rangle \mathrm{d}t \\ - \int_0^T \left\langle b^\dagger, \frac{\partial b}{\partial t} + ([\boldsymbol{U}_0 + \boldsymbol{u}] \cdot \nabla) b - \frac{1}{Pe}\nabla^2 b \right\rangle \mathrm{d}t - \int_0^T \langle p^\dagger, \nabla \cdot \boldsymbol{u} \rangle \ \mathrm{d}t, \end{aligned} \tag{D.1}$$

where $\boldsymbol{u}^\dagger, p^\dagger, b^\dagger$ are the adjoint variables enforcing the constraint equations for the dual variables. Taking variations with respect to the primal variables we obtain the adjoint equations

$$\begin{aligned} \frac{\partial \boldsymbol{u}^\dagger}{\partial t} - \frac{1}{Re}\nabla^2 \boldsymbol{u}^\dagger - \nabla p^\dagger - (\boldsymbol{U}_0 \cdot \nabla)\boldsymbol{u}^\dagger + \boldsymbol{u}^\dagger \cdot (\nabla \boldsymbol{U}_0)^T = (\boldsymbol{u} \cdot \nabla)\boldsymbol{u}^\dagger - \boldsymbol{u}^\dagger \cdot (\nabla \boldsymbol{u})^T - b^\dagger \nabla b, \\ \frac{\partial b^\dagger}{\partial t} - \frac{1}{Pe}\nabla^2 b^\dagger + Ri\boldsymbol{u}^\dagger \cdot \widehat{\boldsymbol{z}} - (\boldsymbol{U}_0 \cdot \nabla) b^\dagger = (\boldsymbol{u} \cdot \nabla) b^\dagger, \qquad \nabla \cdot \boldsymbol{u}^\dagger = 0, \end{aligned} \tag{D.2}$$

and compatibility conditions

$$\boldsymbol{u}^\dagger(\boldsymbol{x}, t = T) = 0, \qquad b^\dagger(\boldsymbol{x}, t = T) = -\nabla^{-2} b(\boldsymbol{x}, T), \tag{D.3}$$

where $b^\dagger$ follows from solving the linear boundary value problem

$$\nabla^2 \psi = b, \quad \text{s.t.} \quad \partial_z \psi(x, z = \pm 1) = 0, \ \int_V \psi \ \mathrm{d}V = 0. \tag{D.4}$$

Whilst the discrete adjoint for these equations can be computed in a similar manner following A, the discrete implementation of this mix-norm cost functional, and its adjoint, merits special attention. The mix norm cost functional takes the form

$$\mathcal{J} = \langle \nabla \psi, W \nabla \psi \rangle, \tag{D.5}$$

where W is an appropriate weight matrix and the symbol ∇ is now taken to mean the discretised gradient operator. With this in consideration, the cost functional can be rewritten in terms of b as

$$\mathcal{J} = \langle \nabla \nabla^{-2} b, W \nabla \nabla^{-2} b \rangle, \tag{D.6}$$

giving the final-time condition

$$b^\dagger(\boldsymbol{x}, T) = (\nabla^{-2})^\dagger \nabla^\dagger W \nabla \psi. \tag{D.7}$$

Note that in deriving (D.7) we have assumed that $W^\dagger = W$. Numerically we decompose ∇ as $(\partial_x, \partial_z)^T$ where ∂_x and ∂_z are the discrete x and z derivatives and denote by $(\partial_x^\dagger, \partial_z^\dagger)$ their discrete-adjoint

counterpart. The discrete adjoint for ∇^{-2} is found directly from the Dedalus implementation of (D.4). In this manner, we find $b^\dagger(\boldsymbol{x}, T)$ by solving $(\nabla^2)^\dagger b^\dagger(\boldsymbol{x}, T) = \nabla^\dagger \mathbf{W}^\dagger \nabla \psi$.

Acknowledgements

The authors wish to thank Ashley P. Willis for his careful reading and recommendations for improving this manuscript. P. M. M. and F. M. acknowledge support from the French program "T-ERC" from Agence Nationale de la Recherche (ANR) under grant agreement ANR-19-ERC7-0008). C. S. S. acknowledges partial support from a grant from the Simons Foundation (Grant No. 662962, GF). He would also like to acknowledge support of funding from the European Union Horizon 2020 research and innovation programme (grant agreement no. D5S-DLV-786780). This work was also supported by the French government, through the UCAJEDI Investments in the Future project managed by the ANR under grant agreement ANR-15-IDEX-0001. The authors are grateful to the OPAL infrastructure from Université Côte d'Azur, Université Côte d'Azur's Center for High-Performance Computing computer facilities for providing resources and support.

References

[1] Pierre-Antoine Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[2] Uri M. Ascher, Steven J. Ruuth, and Raymond J. Spiteri. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Appl. Numer. Math.*, 25:151–167, 1997.

[3] Uri M. Ascher, Steven J. Ruuth, and Brian T. R. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM J. Numer. Anal.*, 32:797–823, 1995.

[4] Didier Auroux and Jacques Blum. Data assimilation methods for an oceanographic problem. In *Multidisciplinary methods for analysis optimization and control of complex systems*, volume 6 of *Mathematics in Industry*, pages 180–194. Springer, 2005.

[5] Jens Berg and Jan Nordström. On the impact of boundary conditions on dual consistent finite difference discretizations. *J. Comput. Phys.*, 236:41–55, 2013.

[6] Nicolas Boumal. An introduction to optimization on smooth manifolds. `https://www.nicolasboumal.net/book`, to appear with Cambridge University Press, 2022.

[7] Nicolas Boumal, Bamdev Mishra, Pierre-Antoine Absil, and Rodolphe Sepulchre. Manopt, a Matlab Toolbox for Optimization on Manifolds. *J. Mach. Learn. Res.*, 15(42):1455–1459, 2014.

[8] Keaton J. Burns, Geoffrey M. Vasil, Jeffrey S. Oishi, Daniel Lecoanet, and Benjamin P. Brown. Dedalus: A flexible framework for numerical simulations with spectral methods. *Phys. Rev. Res.*, 2: article no. 023068, 2020.

[9] Thierry Dauxois, Thomas Peacock, Peter Bauer, Colm P. Caulfield, Claudia Cenedese, Catherine Gorlé, George Haller, Greg N. Ivey, Paul F. Linden, Eckart Meiburg, et al. Confronting grand challenges in environmental fluid mechanics. *Phys. Rev. Fluids*, 6(2): article no. 020501, 2021.

[10] Miguel Fosas De Pando, Denis Sipp, and Peter J. Schmid. Efficient evaluation of the direct and adjoint linearized dynamics from compressible flow solvers. *J. Comput. Phys.*, 231(23):7739–7755, 2012.

[11] S. C. Douglas, S. Amari, and S. Y. Kung. Gradient adaptation under unit-norm constraints. In *Ninth IEEE Signal Processing Workshop on Statistical Signal and Array Processing (Cat. No.98TH8381)*. IEEE, 1998.

[12] Maximilian F. Eggl and Peter J. Schmid. A gradient-based framework for maximizing mixing in binary fluids. *J. Comput. Phys.*, 368:131–153, 2018.

[13] Patrick E. Farrell, David A. Ham, Simon W. Funke, and Marie E. Rognes. Automated derivation of the adjoint of high-level transient finite element programs. *SIAM J. Sci. Comput.*, 35(4):C369–C393, 2013.

[14] Alexandru Fikl, Vincent Le Chenadec, and Taraneh Sayadi. Control and optimization of interfacial flows using adjoint-based techniques. *Fluids*, 5(3):156, 2020.

[15] Alexandru Fikl, Vincent Le Chenadec, Taraneh Sayadi, and Peter J. Schmid. A comprehensive study of adjoint-based optimization of non-linear systems with application to Burgers' equation. In *46th AIAA Fluid Dynamics Conference*, page 3805, 2016.

[16] Dimitry P. G. Foures, Colm P. Caulfield, and Peter J. Schmid. Localization of flow structures using-norm optimization. *J. Fluid Mech.*, 729:672–701, 2013.

[17] Dimitry P. G. Foures, Colm P. Caulfield, and Peter J. Schmid. Optimal mixing in two-dimensional plane Poiseuille flow at finite Péclet number. *J. Fluid Mech.*, 748:241–277, 2014.

[18] Michael Franco, Per-Olof Persson, Will Pazner, and Matthew J. Zahr. An adjoint method using fully implicit runge-kutta schemes for optimization of flow problems. In *AIAA Scitech 2019 Forum*. 2019.

[19] Andreas Griewank and Andrea Walther. Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Trans. Math. Softw.*, 26(1):19–45, 2000.

[20] William W. Hager. Runge–Kutta methods in optimal control and the transformed adjoint system. *Numer. Math.*, 87:247–282, 2000.

[21] William W. Hager and Hongchao Zhang. A survey of nonlinear conjugate gradient methods. *Pac. J. Optim.*, 2(1):35–58, 2006.

[22] Ralf Hartmann. Adjoint consistency analysis of discontinuous Galerkin discretizations. *SIAM J. Numer. Anal.*, 45(6):2671–2696, 2007.

[23] Laurent Hascoët and Valérie Pascual. The Tapenade Automatic Differentiation tool: Principles, Model, and Specification. *ACM Trans. Math. Softw.*, 39(3): article no. 20 (43 pages), 2013.

[24] Conor Heffernan and Colm P. Caulfield. Robust and efficient identification of optimal mixing perturbations using proxy multiscale measures. *Philos. Trans. R. Soc. Lond., Ser. A*, 380(2225): article no. 26 (17 pages), 2022.

[25] Michael Hinze, René Pinnau, Michael Ulbrich, and Stefan Ulbrich. *Optimization with PDE constraints*, volume 23. Springer, 2008.

[26] Steven G. Johnson. The NLopt nonlinear-optimization package. `http://github.com/stevengj/nlopt`, 2014.

[27] Matthew P. Juniper and R. I. Sujith. Sensitivity and nonlinearity of thermoacoustic oscillations. *Annu. Rev. Fluid Mech.*, 50:661–689, 2018.

[28] Rich R. Kerswell. Nonlinear Nonmodal Stability Theory. *Annu. Rev. Fluid Mech.*, 50:319–345, 2018.

[29] E. Knobloch. Spatial Localization in Dissipative Systems. *Ann. Rev. Cond. Matter Phys.*, 6(1):325–359, 2015.

[30] Ali Kord and Jesse Capecelatro. Optimal perturbations for controlling the growth of a Rayleigh–Taylor instability. *J. Fluid Mech.*, 876:150–185, 2019.

[31] Dieter Kraft et al. *A software package for sequential quadratic programming*, volume 28 of *DFVLR Forschungsber*. DFVLR Obersfaffeuhofen, 1988.

[32] Joseph Larmor. 17. How Could a Rotating Body such as the Sun Become a Magnet? *Reports of the British Association*, 87:159–160, 1919.

[33] Daniel Lecoanet and Rich R. Kerswell. Connection between nonlinear energy optimization and instantons. *Phys. Rev. E*, 97: article no. 012212, 2018.

[34] Dmitriy Leykekhman. Investigation of commutative properties of discontinuous Galerkin methods in PDE constrained optimal control problems. *J. Sci. Comput.*, 53(3):483–511, 2012.

[35] David G. Luenberger. *Introduction to linear and nonlinear programming*. Addison-wesley Reading, 1973.

[36] Paul M. Mannix, Calum S. Skene, Didier Auroux, and Florence Marcotte. mannixp/SphereManOpt: Initial Release. `https://zenodo.org/record/7271474`, 2022.

[37] Florence Marcotte and Colm P. Caulfield. Optimal mixing in two-dimensional stratified plane Poiseuille flow at finite Péclet and Richardson numbers. *J. Fluid Mech.*, 853:359–385, 2018.

[38] George Mathew, Igor Mezić, and Linda Petzold. A multiscale measure for mixing. *Physica D*, 211(1-2):23–46, 2005.

[39] Bruno Maugars, Sébastien Bourasseau, Cédric Content, Bertrand Michel, Bérenger Berthoul, Jorge Nunez Ramirez, Pascal Raud, and Laurent Hascoët. Algorithmic Differentiation for an efficient CFD solver. In *ECCOMAS 2022-8th European Congress on Computational Methods in Applied Sciences and Engineering*, 2022.

[40] Sebastian K. Mitusch, Simon W. Funke, and Jørgen S. Dokken. dolfin-adjoint 2018.1: automated adjoints for FEniCS and Firedrake. *J. Open Source Softw.*, 4(38):1292, 2019.

[41] Jorge J. Moré and David J. Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Trans. Math. Softw.*, 20(3):286–307, 1994.

[42] Mu Mu, W. Duan, Q. Wang, and R. Zhang. An extension of conditional nonlinear optimal perturbation approach and its applications. *Nonlinear Process. Geophys.*, 17(2):211–220, 2010.

[43] Jens-Dominik Müller and P. Cusdin. On the performance of discrete adjoint CFD codes using automatic differentiation. *Int. J. Numer. Methods Fluids*, 47(8-9):939–945, 2005.

[44] Uwe Naumann. *The art of differentiating computer programs: an introduction to algorithmic differentiation*. Society for Industrial and Applied Mathematics, 2011.

[45] Eric J. Nielsen and Boris Diskin. Discrete adjoint-based design for unsteady turbulent flows on dynamic overset unstructured grids. *AIAA J.*, 51(6):1355–1373, 2013.

[46] Chris C. T. Pringle and Rich R. Kerswell. Using nonlinear transient growth to construct the minimal seed for shear flow turbulence. *Phys. Rev. Lett.*, 105, 2010.

[47] Adrian Sandu. On the properties of Runge–Kutta discrete adjoints. In *Computational Science – ICCS 2006*, volume 3994 of *Lecture Notes in Computer Science*, pages 550–557. 2006.

[48] Hiroyuki Sato. *Riemannian Optimization and Its Applications*. Springer, 2021.

[49] Hiroyuki Sato. Riemannian conjugate gradient methods: General framework and specific algorithms with convergence analyses. *SIAM J. Optim.*, 32(4):2690–2717, 2022.

[50] Hiroyuki Sato. Riemannian optimization on unit sphere with p-norm and its applications. `https://arxiv.org/abs/2202.11597`, 2022.

[51] Peter J. Schmid, Miguel Fosas de Pando, and Nigel Peake. Stability analysis for n-periodic arrays of fluid systems. *Phys. Rev. Fluids*, 2(11): article no. 113902, 2017.

[52] Calum S. Skene, Chi-An Yeh, Peter J. Schmid, and Kunihiko Taira. Sparsifying the resolvent forcing mode via gradient-based optimisation. *J. Fluid Mech.*, 944: article no. A52 (24 pages), 2022.

[53] A. D. Terwisscha van Scheltinga and H. A. Dijkstra. Conditional nonlinear optimal perturbations of the double-gyre ocean circulation. *Nonlinear Process. Geophys.*, 15(5):727–734, 2008.

[54] Jean-Luc Thiffeault. Using multiscale norms to quantify mixing and transport. *Nonlinearity*, 25(2):R1–R44, 2012.

[55] James Townsend, Niklas Koep, and Sebastian Weichwald. Pymanopt: A Python Toolbox for Optimization on Manifolds using Automatic Differentiation. *J. Mach. Learn. Res.*, 17(137):1–5, 2016.

[56] Jean Utke, Laurent Hascoet, Patrick Heimbach, Chris Hill, Paul Hovland, and Uwe Naumann. Toward adjoinable MPI. In *2009 IEEE International Symposium on Parallel & Distributed Processing*, pages 1–8. IEEE, 2009.

[57] Ramanathan Vishnampet, Daniel J. Bodony, and Jonathan B. Freund. A practical discrete-adjoint method for high-fidelity compressible turbulence simulations. *J. Comput. Phys.*, 285:173–192, 2015.

[58] Ashley P. Willis. Optimization of the magnetic dynamo. *Phys. Rev. Lett.*, 109: article no. 251101, 2012.

[59] Stephen Wright and Jorge Nocedal. *Numerical optimization*, volume 35 of *Springer Series in Operations Research and Financial Engineering*. Springer, 1999.

[60] Matthew J. Zahr and Per-Olof Persson. An adjoint method for a high-order discretization of deforming domain conservation laws for optimization of flow problems. *J. Comput. Phys.*, 326:516–543, 2016.

[61] Hong Zhang, Emil M. Constantinescu, and Barry F. Smith. Petsc TSAdjoint: A Discrete Adjoint ODE Solver for First-Order and Second-Order Sensitivity Analysis. *SIAM J. Sci. Comput.*, 44(1):C1–C24, 2022.