Fast iterative solvers for an optimal transport problem

Fast iterative solvers for an optimal transport problem

Roland Herzog

Technische Universität Chemnitz, Faculty of Mathematics, Reichenhainer Straße 41, 09126 Chemnitz, Germany (roland.herzog@mathematik.tu-chemnitz.de)

John W. Pearson

School of Mathematics, The University of Edinburgh, James Clerk Maxwell Building, The King's Buildings, Peter Guthrie Tait Road, EH9 3FD, United Kingdom (j.pearson@ed.ac.uk)

Martin Stoll

Technische Universität Chemnitz, Faculty of Mathematics, Reichenhainer Straße 41, 09126 Chemnitz, Germany (martin.stoll@mathematik.tu-chemnitz.de)

Abstract

Optimal transport problems pose many challenges when considering their numerical treatment. We investigate the solution of a PDE-constrained optimisation problem subject to a particular transport equation arising from the modelling of image metamorphosis. We present the nonlinear optimisation problem, and discuss the discretisation and treatment of the nonlinearity via a Gauss–Newton scheme. We then derive preconditioners that can be used to solve the linear systems at the heart of the (Gauss–)Newton method.

Keywords: PDE-constrained optimisation, Saddle point systems, Time-dependent PDE-constrained optimisation, Preconditioning, Krylov subspace solver, Optical Flow, Optimal Transport

1. Introduction

The problem of optimal transport is a longstanding and active area of research in applied mathematics and the sciences [1]. Its numerical treatment provides many challenges to the mathematical community (see [2, 3] and the references therein). Our goal in this paper is to discuss the solution of a PDE-constrained optimisation problem where the constraint is given by a transport equation. In the field of PDE-constrained optimisation one typically wants to minimise an objective function with the constraints given by one or more PDEs [4, 5].

Much of our analysis for this formulation builds upon the previous work [6], for which we wish to devise new iterative methods and discretisation schemes.

The study of the original transport problem goes back to the 18th century but a modern formulation was given in [7, 8]. Recent developments include the seminal paper [9], where the problem is phrased as a fluid mechanics problem. A very similar formulation of minimising an objective function subject to a transport equation constraint is found in *optical flow* (cf. for example [10, 11, 12, 13, 14, 15, 16]), which models the apparent 'motion' of an image.

In this manuscript, we examine the optimisation and discretisation of this problem, with a particular focus on the efficient solution of the linear systems that arise at the heart of the outer nonlinear solver. Our primary choice of nonlinear solver for the nonlinear optimisation problem is a Gauss–Newton scheme [6, 17], though we also consider methods based on a full Newton method. As one can typically follow the route of first performing the discretisation followed by deriving the appropriate optimality conditions, or vice versa, we discuss both approaches. We also briefly analyse a modified formulation of the classical transport model. Our main goal is the derivation of the linear system of equations, followed by the introduction of suitable preconditioners that allow a parameter-robust solution of the linear system that is the computational heart of the nonlinear iteration. Such preconditioners have recently received much attention (cf. [18, 19, 20, 21]). We then illustrate that the proposed preconditioners are efficient both for synthetic data as well as practical imaging examples.

The paper is structured as follows. In Section 2 we introduce the problem formulation considered in this work. Section 3 introduces the discretisation of the optimisation problem and the constraint via a finite difference approach. We discuss both discretise-then-optimise and optimise-then-discretise schemes. After introducing a modification to the problem formulation, we discuss two general preconditioning strategies in Section 4. In Section 5 we present numerical experiments for the finite difference method, demonstrating the effectiveness of our discretisation and preconditioning approaches.

2. The optimal transport problem

The problem we examine in this paper is one of minimising the functional

$$\begin{aligned}\mathcal{E}(y,\vec{m}) = \frac{1}{2\gamma}\int_\Omega (y(\vec{x},1)-y_1(\vec{x}))^2 \,\mathrm{d}\Omega + \frac{\delta}{2}\int_0^1\int_\Omega (y(\vec{x},t)-\bar{y}(\vec{x},t))^2 \,\mathrm{d}\Omega\,\mathrm{d}t \\ + \frac{\beta}{2}\int_0^1\int_\Omega (Q\vec{m}(\vec{x},t))^2 \,\mathrm{d}\Omega\,\mathrm{d}t,\end{aligned} \tag{1}$$

where β and γ are (positive) parameters that can be understood as *regularisation* or *penalty parameters*. The parameter γ is chosen in such a way that the computed state $y(\vec{x},1)$ is close to the true final state y_1 at time $T=1$. Here, the velocity $\vec{m}$ represents a control variable, Q is a differential operator (possibly $Q=\mathrm{blkdiag}(I,I)$ or $Q=\mathrm{blkdiag}(\nabla,\nabla)$), and $\bar{y}$ is a specified desired state. The problem is solved on a space-time grid $(\vec{x},t) := ([x_1,x_2],t) \in \Omega\times[0,1]$, where $\Omega\subset\mathbb{R}^2$ denotes the domain occupied by the image.

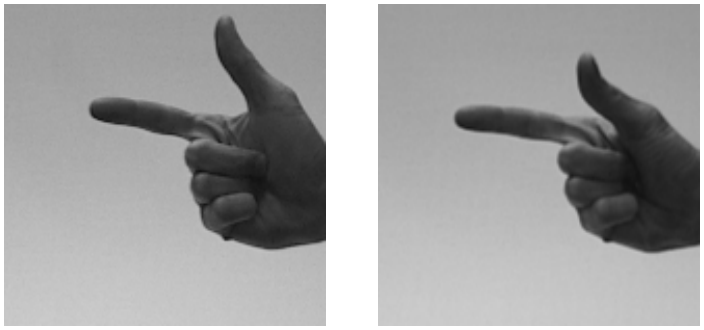
For the majority of the analysis presented in this paper we will consider the case $\delta = 0$, i.e., where

$$\mathcal{E}(y, \vec{m}) = \frac{1}{2\gamma} \int_\Omega (y(\vec{x}, 1) - y_1(\vec{x}))^2 \, \mathrm{d}\Omega + \frac{\beta}{2} \int_0^1 \int_\Omega (Q\vec{m}(\vec{x}, t))^2 \, \mathrm{d}\Omega \, \mathrm{d}t, \qquad (2)$$

however on a number of occasions we will describe modifications which are taken into account when δ is a positive parameter, measuring the deviation of y from the desired state $\bar{y}$ during the *entire* time interval. We typically refer to the minimisation of (1) with $\delta > 0$ as the "full observation case", and the minimisation of (2) (where the state is measured within $\mathcal{E}$ at the final time only) as the "partial observation case". The goal is to minimise the above energy subject to the continuity transport equation

$$y_t + \nabla \cdot (\vec{m} y) = 0, \qquad (3)$$

with the initial condition $y(\vec{x}, 0) = y_0$ as well as appropriate boundary conditions, for instance periodic boundary conditions or Dirichlet conditions. Here, $\vec{m} = [m_1, \, m_2]^T$ is defined for the two-dimensional domain Ω. While such a problem can be found in many areas of science, we wish to apply the above formulation to the estimation of an optical flow. To illustrate a particular set-up, examples for y_0 and y_1 are the two images shown in Fig. 1[1].



(a) Initial image y_0 (b) Target image y_1

Figure 1: An example of an optical flow problem where we have a starting picture on the left and a target picture on the right.

3. Discretisation using finite differences

In this section, we wish to present how we discretise the optimisation problem (2) with constraint (3). An outline is as follows: in Section 3.1 we examine the approach where such a PDE-constrained optimisation problem is discretised

[1]The images were taken from `http://cs.brown.edu/people/mjblack/images.html`.

first, upon which optimality conditions are found. In Section 3.2 we then extend this methodology to the setting where optimality conditions are first derived (on a formal basis) on the continuous level, whereupon these are then discretised. In Section 3.3 we then discuss the application of our methodology to a slight modification of the PDE (3). In Section 3.4, we explain how the optimality conditions vary if one instead considers the cost functional (1), with an additional non-zero parameter δ measuring the deviation of the state y from $\bar{y}$ throughout the entire time interval.

3.1. Discretise-then-optimise

A control problem using the formulation (3) of the transport equation was introduced in [6], and we therefore follow this approach for the derivation of the *discretise-then-optimise* system. We start by discretising the objective function and nonlinear PDE constraint to build a discrete Lagrangian, which then allows us to compute the solution via a Gauss–Newton or Lagrange–Newton scheme [22, Ch. 10.3 & 18]. We employ an implicit Lax–Friedrichs method [3, 6] for the forward PDE

$$\begin{aligned}\frac{1}{\tau}\left(y_{i,j}^{(k+1)} - \frac{1}{4}\left[y_{i+1,j}^{(k)} + y_{i-1,j}^{(k)} + y_{i,j+1}^{(k)} + y_{i,j-1}^{(k)}\right]\right)\\ + \frac{1}{2h}\left((m_1 \odot y)_{i+1,j}^{(k+1)} - (m_1 \odot y)_{i-1,j}^{(k+1)} + (m_2 \odot y)_{i,j+1}^{(k+1)} - (m_2 \odot y)_{i,j-1}^{(k+1)}\right) = 0,\end{aligned}$$

that we can manipulate to arrive at the following system for each time-step:

$$\left(I + \frac{\tau}{2h}K(\mathbf{m}^{(k+1)})\right)\mathbf{y}^{(k+1)} = D_t\mathbf{y}^{(k)}, \quad k = 0, 1, ..., N_t - 1.$$

Here, $\odot$ denotes the (componentwise) Hadamard product of vectors, τ represents the time-step with h the spatial mesh parameter, the matrix D_t arises from the four point stencil used to approximate the time derivative, and

$$K(\mathbf{m}^{(k+1)}) = \begin{bmatrix} D_1 & D_2 \end{bmatrix}\begin{bmatrix} \operatorname{diag}\left(\mathbf{m}_1^{(k+1)}\right) \\ \operatorname{diag}\left(\mathbf{m}_2^{(k+1)}\right) \end{bmatrix},$$

where D_1 and D_2 are centred finite difference matrices. We can then formulate an all-at-once approach using the notation

$$L(\mathbf{m}^{(k+1)})\,\mathbf{y}^{(k+1)} = D_t\mathbf{y}^{(k)},$$

where $L(\mathbf{m}^{(k)}) = I + \frac{\tau}{2h} K(\mathbf{m}^{(k)})$, to obtain for a given number N_t time-steps a matrix system of the form

$$\underbrace{\begin{bmatrix} L(\mathbf{m}^{(1)}) & & & & \\ -D_t & L(\mathbf{m}^{(2)}) & & & \\ & -D_t & \ddots & & \\ & & \ddots & \ddots & \\ & & & -D_t & L(\mathbf{m}^{(N_t)}) \end{bmatrix}}_{\mathcal{A}(\mathbf{m})} \underbrace{\begin{bmatrix} \mathbf{y}^{(1)} \\ \mathbf{y}^{(2)} \\ \vdots \\ \vdots \\ \mathbf{y}^{(N_t)} \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} D_t \mathbf{y}^{(0)} \\ \mathbf{0} \\ \vdots \\ \vdots \\ \mathbf{0} \end{bmatrix}}_{\mathbf{d}}$$

representing the discretised PDE constraint. Depending on the boundary conditions (Dirichlet or periodic), the matrices $L(\mathbf{m}^{(k)})$ and D_t need to be slightly modified in rows pertaining to boundary nodes. In this work we apply periodic boundary conditions, in analogy to the work of Benzi, Haber and Taralli [6]. Furthermore, we may approximate the objective function (2) on the discrete level by

$$\boldsymbol{\mathcal{E}}(\mathbf{y}, \mathbf{m}) = \frac{1}{2\gamma} (\mathbf{y}^{(N_t)} - \mathbf{y}_1)^T M (\mathbf{y}^{(N_t)} - \mathbf{y}_1) + \frac{\beta\,\tau}{2} \mathbf{m}^T \mathcal{M} \mathbf{m},$$

where $\mathcal{M} = \text{blkdiag}(W, \ldots, W)$, and W is obtained from the discretisation of the term $\int_\Omega (Q\vec{m}(\vec{x}, t))^2 \, \mathrm{d}\Omega$ (which could simply be a scaled identity operator). Note that, for simplicity, we have not included possible scalings of the individual W matrices in $\mathcal{M}$ as these depend on the choice of discretisation performed in time. We now form the discrete Lagrangian for this problem

$$\boldsymbol{\mathcal{L}}(\mathbf{y}, \mathbf{m}, \mathbf{p}) = \boldsymbol{\mathcal{E}}(\mathbf{y}, \mathbf{m}) + \mathbf{p}^T \mathcal{Q} \left(\mathcal{A}(\mathbf{m})\, \mathbf{y} - \mathbf{d} \right),$$

where $\mathcal{Q}$ is a matrix allowing us to interpret the Lagrange multiplier $\mathbf{p}$ as a grid function. For simplicity we assume that $\mathcal{Q} = \tau h^2 I$, with I the identity matrix of the appropriate dimension. Following [6], the computation of the first-order conditions

$$\boldsymbol{\mathcal{L}}_{\mathbf{y}} = \mathbf{0}, \quad \boldsymbol{\mathcal{L}}_{\mathbf{m}} = \mathbf{0}, \quad \boldsymbol{\mathcal{L}}_{\mathbf{p}} = \mathbf{0},$$

leads to

$$\gamma^{-1} \mathcal{M}_{N_t} (\mathbf{y} - \mathbf{y}_{0,1}) + \mathcal{A}(\mathbf{m})^T \mathcal{Q}\, \mathbf{p} = \mathbf{0}, \tag{4a}$$

$$\beta\,\tau \mathcal{M} \mathbf{m} + \mathcal{J}(\mathbf{y})^T \mathcal{Q}\, \mathbf{p} = \mathbf{0}, \tag{4b}$$

$$\mathcal{Q} \left(\mathcal{A}(\mathbf{m})\, \mathbf{y} - \mathbf{d} \right) = \mathbf{0}, \tag{4c}$$

where $\mathbf{y}_{0,1}$ is a vector containing vectors of zeros for every time-step, apart from the final step which contains the vector $\mathbf{y}_1$. The matrix $\mathcal{M}_{N_t}$ contains zero blocks at every time-step, apart from the final time-step, which gives rise to an identity matrix scaled by h^2, denoted as M. Note that in optical flow applications (relating to our subsequent discussion of the "full observation case")

one is often given an image for every time-step, meaning the matrix $\mathcal{M}_{N_t}$ can be modified to one that does not contain zero diagonal blocks, and the vector $\mathbf{y}_{0,1}$ contains all time instances of these images. Further, $\mathcal{J}(\mathbf{y})$ denotes the block diagonal matrix blkdiag $\big(J(\mathbf{y}^{(1)}), \ldots, J(\mathbf{y}^{(N_t)})\big)$, where

$$\frac{\tau}{2h}\begin{bmatrix} D_1 & D_2 \end{bmatrix}\begin{bmatrix} \operatorname{diag}\left(\mathbf{y}^{(j)}\right) & 0 \\ 0 & \operatorname{diag}\left(\mathbf{y}^{(j)}\right) \end{bmatrix} =: J(\mathbf{y}^{(j)}),$$

at each time-step $j = 1, 2, ..., N_t$. The equations (4) represent a nonlinear system, which we have to treat with a nonlinear optimisation scheme. We follow [6] and use a Gauss–Newton method for the solution of the first-order conditions, which leads to the matrix system

$$\begin{bmatrix} \gamma^{-1}\mathcal{M}_{N_t} & 0 & \mathcal{A}(\mathbf{m})^T\mathcal{Q} \\ 0 & \beta\,\tau\mathcal{M} & \mathcal{J}(\mathbf{y})^T\mathcal{Q} \\ \mathcal{Q}\mathcal{A}(\mathbf{m}) & \mathcal{Q}\mathcal{J}(\mathbf{y}) & 0 \end{bmatrix}\begin{bmatrix} \mathbf{s_y} \\ \mathbf{s_m} \\ \mathbf{s_p} \end{bmatrix} = -\begin{bmatrix} \gamma^{-1}\mathcal{M}_{N_t}(\mathbf{y}-\mathbf{y}_{0,1}) + \mathcal{A}(\mathbf{m})^T\mathcal{Q}\,\mathbf{p} \\ \beta\,\tau\mathcal{M}\mathbf{m} + \mathcal{J}(\mathbf{y})^T\mathcal{Q}\,\mathbf{p} \\ \mathcal{Q}\left(\mathcal{A}(\mathbf{m})\,\mathbf{y} - \mathbf{d}\right) \end{bmatrix} \tag{5}$$

at every step of the nonlinear iteration.

3.2. Optimise-then-discretise

We now highlight that it is also possible to follow the *optimise-then-discretise* approach, where we commence by considering the continuous Lagrangian

$$\mathcal{L}(y, \vec{m}, p) = \mathcal{E}(y, \vec{m}) + \int_0^1 \int_\Omega p\,(y_t + \nabla\cdot(\vec{m}y))\;\mathrm{d}\Omega\,\mathrm{d}t,$$

and then search for the continuous first-order conditions. Note that, for brevity, we have omitted the initial and boundary conditions within this Lagrangian functional; these are also accounted for and reappear in the optimality conditions. Proceeding formally, by considering the Fréchet derivatives of the Lagrangian $\mathcal{L}$ with respect to y, $\vec{m}$ and p, and integrating by parts, we then obtain the conditions

$$-p_t - \vec{m}\cdot\nabla p = 0, \tag{6a}$$

$$\beta\,Q^*Q\vec{m} - y\nabla p = \vec{0}, \tag{6b}$$

$$y_t + \nabla\cdot(\vec{m}y) = 0, \tag{6c}$$

together with the initial condition for y, and the final-time condition

$$\frac{1}{\gamma}(y(\cdot,1) - y_1) = -p(1),$$

corresponding to the adjoint equation. We have now established the continuous first-order conditions for the optimal transport problem. Equations (6) represent a nonlinear system, which are augmented with boundary and initial/final-time

conditions. Equivalently, we can write this as $G(\vec{z})$ with $\vec{z} = [y,\ m_1,\ m_2,\ p]^T$ and solve this nonlinear problem using a Gauss–Newton or Newton's method. The latter is the Lagrange–Newton or sequential quadratic programming (SQP) scheme. At each iteration we need to solve

$$G'(\vec{z}_k)\,\vec{s}_k = -G(\vec{z}_k) =: \vec{b},$$

where $\vec{s}_k = \vec{z} - \vec{z}_k$, until convergence of the method is achieved. We now need to form the derivative of G to solve the Newton problem, and obtain

$$-(s_p)_t - \vec{s}_m \cdot \nabla p - \vec{m} \cdot \nabla s_p = b_1, \tag{7a}$$

$$\beta\, Q^* Q \vec{s}_m - y \nabla s_p - s_y \nabla p = \vec{b}_2, \tag{7b}$$

$$(s_y)_t + \nabla \cdot (\vec{m} s_y) + \nabla \cdot (\vec{s}_m y) = b_3. \tag{7c}$$

We examine the discretisation of this system of equations, starting with the treatment of the term

$$(s_y)_t + \nabla \cdot (\vec{m} s_y)$$

in the forward equation (7c), using the implicit Lax–Friedrichs scheme. This gives

$$\frac{1}{\tau}\left((s_y)_{i,j}^{(k+1)} - \frac{1}{4}\left[(s_y)_{i+1,j}^{(k)} + (s_y)_{i-1,j}^{(k)} + (s_y)_{i,j+1}^{(k)} + (s_y)_{i,j-1}^{(k)}\right]\right)$$
$$+ \frac{1}{2h}\left((m_1 \odot s_y)_{i+1,j}^{(k+1)} - (m_1 \odot s_y)_{i-1,j}^{(k+1)} + (m_2 \odot s_y)_{i,j+1}^{(k+1)} - (m_2 \odot s_y)_{i,j-1}^{(k+1)}\right).$$

Written in the same form as for the discretise-then-optimise approach, the matrices corresponding to the term at each time-step are $\left(I + \frac{\tau}{2h} K(\mathbf{m}^{(k+1)})\right) \mathbf{s_y}^{(k+1)} - D_t \mathbf{s_y}^{(k)}$, for $k = 0, 1, ..., N_t - 1$. The discretisation of $\nabla \cdot (\vec{s}_m y)$ is performed analogously and we obtain

$$\nabla \cdot (\vec{s}_m y) = \frac{\partial (s_{m_1} y)}{\partial x_1} + \frac{\partial (s_{m_2} y)}{\partial x_2},$$

where $\vec{s}_m = [s_{m_1},\ s_{m_2}]^T$, which leads to

$$\frac{1}{2h}\left((s_{m_1} \odot y)_{i+1,j}^{(k+1)} - (s_{m_1} \odot y)_{i-1,j}^{(k+1)} + (s_{m_2} \odot y)_{i,j+1}^{(k+1)} - (s_{m_2} \odot y)_{i,j-1}^{(k+1)}\right),$$

and when taking into account the multiplication by the time-step τ results in

$$\frac{\tau}{2h}\left((s_{m_1} \odot y)_{i+1,j}^{(k+1)} - (s_{m_1} \odot y)_{i-1,j}^{(k+1)} + (s_{m_2} \odot y)_{i,j+1}^{(k+1)} - (s_{m_2} \odot y)_{i,j-1}^{(k+1)}\right).$$

We write this in matrix form as $\mathcal{J}(\mathbf{y})\,\mathbf{s_m}$. Consider now the discretisation of the terms arising from the continuous gradient equation (7b). For the term $-s_y \nabla p = \left[-s_y \frac{\partial p}{\partial x_1},\ -s_y \frac{\partial p}{\partial x_2}\right]^T$, we will obtain entries of the form

$$-(s_y)_{i,j}^{(k+1)} \frac{1}{2h} \begin{bmatrix} p_{i+1,j}^{(k+1)} - p_{i-1,j}^{(k+1)} \\ p_{i,j+1}^{(k+1)} - p_{i,j-1}^{(k+1)} \end{bmatrix}$$

at the $(k+1)$-st time-step, which can be written in matrix form (when multiplied by the time-step τ) as

$$\frac{\tau}{2h}\begin{bmatrix} \operatorname{diag}\left(D_1^T\mathbf{p}^{(k+1)}\right) \\ \operatorname{diag}\left(D_2^T\mathbf{p}^{(k+1)}\right) \end{bmatrix}\mathbf{s}_{\mathbf{y}}^{(k+1)}.$$

In an analogous way, we may discretise the term $-y\nabla s_p = \left[-y\frac{\partial s_p}{\partial x_1},\ -y\frac{\partial s_p}{\partial x_2}\right]^T$ by

$$-y_{i,j}^{(k+1)}\frac{1}{2h}\begin{bmatrix} (s_p)_{i+1,j}^{(k+1)} - (s_p)_{i-1,j}^{(k+1)} \\ (s_p)_{i,j+1}^{(k+1)} - (s_p)_{i,j-1}^{(k+1)} \end{bmatrix},$$

which in block matrix form will lead to terms of the form

$$\frac{\tau}{2h}\begin{bmatrix} \operatorname{diag}\left(\mathbf{y}^{(k+1)}\right) & 0 \\ 0 & \operatorname{diag}\left(\mathbf{y}^{(k+1)}\right) \end{bmatrix}\begin{bmatrix} D_1^T \\ D_2^T \end{bmatrix}\mathbf{s}_{\mathbf{p}}^{(k+1)},$$

abbreviated by $\mathcal{J}(\mathbf{y})^T\mathbf{s}_{\mathbf{p}}$. Finally, let us analyse the terms within the adjoint equation (7a). The term

$$-\vec{s}_m\cdot\nabla p = -s_{m_1}\frac{\partial p}{\partial x_1} - s_{m_2}\frac{\partial p}{\partial x_2},$$

is approximated at time t_{k+1} by

$$-(s_{m_1})_{i,j}^{(k+1)}\frac{1}{2h}\left(p_{i+1,j}^{(k+1)} - p_{i-1,j}^{(k+1)}\right) - (s_{m_2})_{i,j}^{(k+1)}\frac{1}{2h}\left(p_{i,j+1}^{(k+1)} - p_{i,j-1}^{(k+1)}\right),$$

or in matrix form, when considering the time-step τ,

$$\frac{\tau}{2h}\begin{bmatrix} \operatorname{diag}\left(D_1^T\mathbf{p}^{(k+1)}\right) & \operatorname{diag}\left(D_2^T\mathbf{p}^{(k+1)}\right) \end{bmatrix}\mathbf{s}_{\mathbf{m}}^{(k+1)} =: \mathcal{G}(\mathbf{p})\,\mathbf{s}_{\mathbf{m}}^{(k+1)}.$$

By now it is clear that the collection of all the previously discretised expressions results in a linear system of comparable structure to the matrix (5) obtained from the discretise-then-optimise, Gauss–Newton approach. The last ingredient needed is a discretised version of the adjoint operator, i.e.,

$$-(s_p)_t - \vec{m}\cdot\nabla s_p = -(s_p)_t - m_1\frac{\partial s_p}{\partial x_1} - m_2\frac{\partial s_p}{\partial x_2}.$$

An implicit Lax–Friedrichs scheme again uses forward averaged differences in time and centred differences in space, leading to equations of the form

$$\begin{aligned} &-\frac{1}{\tau}\left((s_p)_{i,j}^{(k+1)} - \frac{1}{4}\left[(s_p)_{i+1,j}^{(k)} + (s_p)_{i-1,j}^{(k)} + (s_p)_{i,j+1}^{(k)} + (s_p)_{i,j-1}^{(k)}\right]\right) \\ &\quad -\frac{1}{2h}\left((m_1)_{i,j}\left((s_p)_{i+1,j}^{(k+1)} - (s_p)_{i-1,j}^{(k+1)}\right) + (m_2)_{i,j}\left((s_p)_{i,j+1}^{(k+1)} - (s_p)_{i,j-1}^{(k+1)}\right)\right), \end{aligned}$$

which in turn may be summarised by matrices $\left(I + \frac{\tau}{2h}L(\mathbf{m}^{(k+1)})\right)\mathbf{s}_{\mathbf{p}}^{(k+1)} - D_t\mathbf{s}_{\mathbf{p}}^{(k)}$. These may be assembled for all time-steps into a high-dimensional

linear system $\mathcal{B}(\mathbf{m})$ in an analogous manner to $\mathcal{A}(\mathbf{m})$. We have now discretised the PDE-constrained optimisation problem using the optimise-then-discretise approach, obtaining a matrix system of the form

$$\begin{bmatrix} \gamma^{-1}\mathcal{M}_{N_t} & \mathcal{G}(\mathbf{p}) & \mathcal{B}(\mathbf{m}) \\ \mathcal{G}(\mathbf{p})^T & \beta\,\tau\mathcal{M} & \mathcal{J}(\mathbf{y})^T \\ \mathcal{A}(\mathbf{m}) & \mathcal{J}(\mathbf{y}) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{s_y} \\ \mathbf{s_m} \\ \mathbf{s_p} \end{bmatrix} = \mathbf{b}. \tag{8}$$

Note that we have not yet established that the discretisation of the adjoint equation above leads to the desired (symmetric) form with $\mathcal{A}(\mathbf{m})^T = \mathcal{B}(\mathbf{m})$. For both matrices the diagonal blocks are of key interest, and we discuss these particular blocks now. For $\mathcal{A}(\mathbf{m})^T$, we obtain for the crucial diagonal blocks that

$$\begin{aligned} \left(K(\mathbf{m}^{(k+1)})\right)^T \mathbf{s}_\mathbf{p}^{(k+1)} &= \begin{bmatrix} \operatorname{diag}\left(\mathbf{m}_1^{(k+1)}\right) & \operatorname{diag}\left(\mathbf{m}_2^{(k+1)}\right) \end{bmatrix} \begin{bmatrix} D_1^T \mathbf{s}_\mathbf{p}^{(k+1)} \\ D_2^T \mathbf{s}_\mathbf{p}^{(k+1)} \end{bmatrix} \\ &= \operatorname{diag}\left(\mathbf{m}_1^{(k+1)}\right) D_1^T \mathbf{s}_\mathbf{p}^{(k+1)} + \operatorname{diag}\left(\mathbf{m}_2^{(k+1)}\right) D_2^T \mathbf{s}_\mathbf{p}^{(k+1)}, \end{aligned}$$

whereupon applying $D_1^T = -D_1$ and $D_2^T = -D_2$ clearly leads[2] to the desired form within $\mathcal{B}(\mathbf{m})$.

We emphasise that the matrix system (8) was obtained using a full Newton method, as opposed to the analysis for the discretise-then-optimise method for which the Gauss–Newton approach of [6] was applied. The main consequence of this change in the outer iteration is the appearance of the $\mathcal{G}(\mathbf{p})$ and $\mathcal{G}(\mathbf{p})^T$ blocks in (8). We also point out that in (5) the scaling matrix $\mathcal{Q}$ was used for the Lagrange multiplier, following [6]. Such a scaling could also be used to make the system (8) resemble the discretise-then-optimise approach more closely.

3.3. Alternative problem formulation

Whereas we focus for the most part on the optimal transport problem given in (2)–(3), we also wish to briefly discuss an alternative formulation given by the minimisation of

$$\mathcal{E}(y, \vec{m}) = \frac{1}{2\gamma} \int_\Omega (y(\vec{x}, 1) - y_1)^2 \, \mathrm{d}\Omega + \frac{\beta}{2} \int_0^1 \int_\Omega (Q\vec{m}(\vec{x}, t))^2 \, \mathrm{d}\Omega \, \mathrm{d}t$$

subject to the advection transport equation (cf. [16])

$$y_t + \vec{m} \cdot \nabla y = 0, \tag{9}$$

along with appropriate boundary and initial conditions. Let us briefly compare (9) with (3). The divergence theorem implies $\frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega y \, \mathrm{d}\Omega = -\int_\Gamma (\vec{m} y) \cdot \vec{n} \, \mathrm{d}\Gamma$, and

[2] Due to the stucture of the differential operators corresponding to D_1 and D_2, we can form these as skew-symmetric matrices by setting to zero the entries corresponding to boundary conditions, and transferring this information to the right-hand side of the matrix system.

thus (3) will be mass preserving in the presence of homogeneous Dirichlet or periodic boundary conditions. By constrast, mass may be produced or removed in (9) unless $\nabla \cdot \vec{m} = 0$ holds.

Discretising the objective function as before results in the following functional on the discrete level:

$$\boldsymbol{\mathcal{E}}(\mathbf{y}, \mathbf{m}) = \frac{1}{2\gamma}(\mathbf{y}^{(N_t)} - \mathbf{y}_1)^T M(\mathbf{y}^{(N_t)} - \mathbf{y}_1) + \frac{\beta\,\tau}{2}\mathbf{m}^T \mathcal{M}\mathbf{m}. \tag{10}$$

The discretisation of the transport equation (9) via an implicit Lax–Friedrichs scheme [3] gives

$$\begin{aligned}\frac{1}{\tau}\left(y_{i,j}^{(k+1)} - \frac{1}{4}\left[y_{i+1,j}^{(k)} + y_{i-1,j}^{(k)} + y_{i,j+1}^{(k)} + y_{i,j-1}^{(k)}\right]\right)\\ + \frac{1}{2h}\left((m_1)_{i,j}\left(y_{i+1,j}^{(k+1)} - y_{i-1,j}^{(k+1)}\right) + (m_2)_{i,j}\left(y_{i,j+1}^{(k+1)} - y_{i,j-1}^{(k+1)}\right)\right) = 0,\end{aligned}$$

which can be written in matrix form as

$$\left(I + \frac{\tau}{2h}\widetilde{K}(\mathbf{m}^{(k+1)})\right)\mathbf{y}^{(k+1)} = D_t\mathbf{y}^{(k)},$$

where

$$\widetilde{K}(\mathbf{m}^{(k+1)}) = \begin{bmatrix} \operatorname{diag}\left(\mathbf{m}_1^{(k+1)}\right) & \operatorname{diag}\left(\mathbf{m}_2^{(k+1)}\right) \end{bmatrix}\begin{bmatrix} D_1 \\ D_2 \end{bmatrix}.$$

Therefore, in block form, the system of equations for the forward problem at all time-steps reads (with $\widetilde{L}(\mathbf{m}^{(k)}) = I + \frac{\tau}{2h}\widetilde{K}(\mathbf{m}^{(k)})$):

$$\underbrace{\begin{bmatrix} \widetilde{L}(\mathbf{m}^{(1)}) & & & & \\ -D_t & \widetilde{L}(\mathbf{m}^{(2)}) & & & \\ & -D_t & \ddots & & \\ & & \ddots & \ddots & \\ & & & -D_t & \widetilde{L}(\mathbf{m}^{(N_t)}) \end{bmatrix}}_{\widetilde{\mathcal{A}}(\mathbf{m})} \underbrace{\begin{bmatrix} \mathbf{y}^{(1)} \\ \mathbf{y}^{(2)} \\ \vdots \\ \vdots \\ \mathbf{y}^{(N_t)} \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} D_t\mathbf{y}^{(0)} \\ \mathbf{0} \\ \vdots \\ \vdots \\ \mathbf{0} \end{bmatrix}}_{\mathbf{d}}. \tag{11}$$

Using the standard Lagrangian approach for differentiating the objective function (10) subject to the constraints (11), we obtain the first-order conditions, neglecting the scaling matrix $\mathcal{Q}$:

$$\begin{aligned}\boldsymbol{\mathcal{L}}_{\mathbf{y}} &= \gamma^{-1}\mathcal{M}_{N_t}(\mathbf{y} - \mathbf{y}_{0,1}) + \widetilde{\mathcal{A}}(\mathbf{m})^T\mathbf{p} = \mathbf{0},\\ \boldsymbol{\mathcal{L}}_{\mathbf{m}} &= \beta\,\tau\mathcal{M}\mathbf{m} + \widetilde{\mathcal{G}}(\mathbf{y})\,\mathbf{p} = \mathbf{0},\\ \boldsymbol{\mathcal{L}}_{\mathbf{p}} &= \widetilde{\mathcal{A}}(\mathbf{m})\,\mathbf{y} - \mathbf{d} = \mathbf{0},\end{aligned}$$

where

$$\widetilde{G}^{(j)} = \frac{\tau}{2h}\begin{bmatrix} \operatorname{diag}(\mathbf{y}_1^{(j)}) \\ \operatorname{diag}(\mathbf{y}_2^{(j)}) \end{bmatrix} \text{ with } \mathbf{y}_i^{(j)} = D_i y^{(j)} \text{ and } \widetilde{\mathcal{G}} = \operatorname{blkdiag}\left(\widetilde{G}^{(1)}, \ldots, \widetilde{G}^{(N_t)}\right).$$

As for the previous problem formulation in Section 3.1 we may then write down a Gauss–Newton scheme for this problem, governed by the matrix

$$\begin{bmatrix} \gamma^{-1}\mathcal{M}_{N_t} & 0 & \widetilde{\mathcal{A}}(\mathbf{m})^T \\ 0 & \beta\,\tau\mathcal{M} & \widetilde{\mathcal{G}}(\mathbf{y}) \\ \widetilde{\mathcal{A}}(\mathbf{m}) & \widetilde{\mathcal{G}}(\mathbf{y})^T & 0 \end{bmatrix}. \tag{12}$$

3.4. Modified cost functional

We now briefly discuss the changes to the optimality conditions and matrix systems if the cost functional (1) is instead considered. In this case, when the discretise-then-optimise method is applied, the discrete approximation of $\mathcal{E}$ is given by

$$\boldsymbol{\mathcal{E}}(\mathbf{y},\mathbf{m}) = \frac{1}{2\gamma}(\mathbf{y}^{(N_t)}-\mathbf{y}_1)^T M(\mathbf{y}^{(N_t)}-\mathbf{y}_1) + \frac{\delta\,\tau}{2}(\mathbf{y}-\bar{\mathbf{y}})^T\bar{\mathcal{M}}(\mathbf{y}-\bar{\mathbf{y}}) + \frac{\beta\,\tau}{2}\mathbf{m}^T\mathcal{M}\mathbf{m},$$

where $\bar{\mathbf{y}}$ contains the discrete values of the desired state $\bar{y}$ at each time-step, and $\bar{\mathcal{M}}$ is a block diagonal matrix corresponding to a scaled identity operator applied at each time-step. The equations $\boldsymbol{\mathcal{L}}_{\mathbf{m}} = \boldsymbol{\mathcal{L}}_{\mathbf{p}} = \mathbf{0}$, as given by (4b)–(4c), will then hold as before. By contrast, the equation $\boldsymbol{\mathcal{L}}_{\mathbf{y}} = \mathbf{0}$ becomes

$$\gamma^{-1}\mathcal{M}_{N_t}(\mathbf{y}-\mathbf{y}_{0,1}) + \delta\,\tau\bar{\mathcal{M}}(\mathbf{y}-\bar{\mathbf{y}}) + \mathcal{A}(\mathbf{m})^T\mathcal{Q}\,\mathbf{p} = \mathbf{0},$$

and the Gauss–Newton system (5) is therefore modified to the form

$$\begin{bmatrix} \gamma^{-1}\mathcal{M}_{N_t} + \delta\,\tau\bar{\mathcal{M}} & 0 & \mathcal{A}(\mathbf{m})^T\mathcal{Q} \\ 0 & \beta\,\tau\mathcal{M} & \mathcal{J}(\mathbf{y})^T\mathcal{Q} \\ \mathcal{Q}\mathcal{A}(\mathbf{m}) & \mathcal{Q}\mathcal{J}(\mathbf{y}) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{s}_{\mathbf{y}} \\ \mathbf{s}_{\mathbf{m}} \\ \mathbf{s}_{\mathbf{p}} \end{bmatrix}$$
$$= -\begin{bmatrix} \gamma^{-1}\mathcal{M}_{N_t}(\mathbf{y}-\mathbf{y}_{0,1}) + \delta\,\tau\bar{\mathcal{M}}(\mathbf{y}-\bar{\mathbf{y}}) + \mathcal{A}(\mathbf{m})^T\mathcal{Q}\,\mathbf{p} \\ \beta\,\tau\mathcal{M}\mathbf{m} + \mathcal{J}(\mathbf{y})^T\mathcal{Q}\,\mathbf{p} \\ \mathcal{Q}\,(\mathcal{A}(\mathbf{m})\,\mathbf{y} - \mathbf{d}) \end{bmatrix}.$$

Similarly, for the matrix system (8) arising from the optimise-then-discretise approach, the top left entry $\gamma^{-1}\mathcal{M}_{N_t}$ must be replaced by $\gamma^{-1}\mathcal{M}_{N_t} + \delta\,\tau\bar{\mathcal{M}}$ if the modified cost functional (1) is used.

4. Preconditioning

The most important step within our algorithm, in order to minimise the computational work required, is to accurately and efficiently solve large and

sparse linear systems. To illustrate our methodology, we focus our description on the Gauss–Newton matrix derived from the discretise-then-optimise case for problem (2)–(3); see (5):

$$\begin{bmatrix} \gamma^{-1}\mathcal{M}_{N_t} & 0 & \mathcal{A}(\mathbf{m})^T\mathcal{Q} \\ 0 & \beta\tau\mathcal{M} & \mathcal{J}(\mathbf{y})^T\mathcal{Q} \\ \mathcal{Q}\mathcal{A}(\mathbf{m}) & \mathcal{Q}\mathcal{J}(\mathbf{y}) & 0 \end{bmatrix}. \tag{13}$$

We approach the solution of such linear systems by exploiting the *saddle point form* of the matrices involved. It is well known that saddle point matrices of the form

$$\begin{bmatrix} A & B^T \\ C & 0 \end{bmatrix} \tag{14}$$

can be effectively approximated by the block diagonal or block triangular preconditioners

$$P_D = \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix}, \qquad P_T = \begin{bmatrix} A & 0 \\ B & -S \end{bmatrix},$$

where $S := CA^{-1}B^T$ denotes the (negative) *Schur complement* of the matrix system, provided that A and S are non-singular. It can be shown (see [23, 24]) that preconditioning the saddle point system with P_D or P_T results in the convergence of a Krylov subspace method in 3 or 2 iterations, respectively. It can also be shown (see [25]) that a similar block triangular preconditioner may also be applied if the $(2,2)$ block of (14) is non-zero. This convergence is independent of all parameters in our problem set-up, such as the regularisation parameters, the mesh parameter, or the time-step.

Of course, the so-called '*ideal preconditioners*' P_D and P_T would not be applied in practice, as the computational cost of inverting A and S would be almost as great as (or maybe greater than) inverting the entire system. We therefore wish to consider variants of these preconditioners, where the $(1,1)$ block and Schur complement are replaced with suitable (cheap) approximations. For the matrix (13), we see that

$$A = \begin{bmatrix} \gamma^{-1}\mathcal{M}_{N_t} & 0 \\ 0 & \beta\tau\mathcal{M} \end{bmatrix}, \qquad B = C = \begin{bmatrix} \mathcal{Q}\mathcal{A}(\mathbf{m}) & \mathcal{Q}\mathcal{J}(\mathbf{y}) \end{bmatrix}.$$

Below, we present two preconditioners which we discover to be effective for this system. In the case of "full observations", i.e., $\delta > 0$ and with the desired state given for every time-step, as in some optical flow problems, the matrix $\mathcal{M}_{N_t}$ is block diagonal and invertible, which makes preconditioning more straightforward. We therefore focus on the case where $\mathcal{M}_{N_t}$ is highly singular[3] and only

[3]By this we mean that a very high proportion of the eigenvalues of the matrix are equal to zero.

comment on the more straightforward case. The goal for these approximations is to carry over the parameter robustness to the inexact preconditioners that we apply in practice.

4.1. First preconditioner

The first preconditioner we introduce is based on the block diagonal structure P_D, but with the $(1,1)$ block and Schur complement replaced with suitable approximations. For the $(1,1)$ block, we write that

$$A = \begin{bmatrix} \gamma^{-1}\mathcal{M}_{N_t} & 0 \\ 0 & \beta\,\tau\mathcal{M} \end{bmatrix} \approx \begin{bmatrix} \widehat{\mathcal{M}}_{N_t} & 0 \\ 0 & \beta\,\tau\mathcal{M} \end{bmatrix} =: \widehat{A},$$

where $\widehat{\mathcal{M}}_{N_t}$ approximates the highly singular matrix $\gamma^{-1}\mathcal{M}_{N_t}$. As suggested by Benzi, Haber and Taralli in [6], all zero diagonal entries in the $(1,1)$ block are replaced with μ within the preconditioner, where this parameter reflects the mean ratio of diagonal entries between the first and second terms of the Schur complement.

Since the $(1,1)$ block is highly singular, we define the Schur complement of the "perturbed" system as

$$S = \mathcal{Q}\mathcal{A}(\mathbf{m})\widehat{\mathcal{M}}_{N_t}^{-1}\mathcal{A}(\mathbf{m})^T\mathcal{Q} + \frac{1}{\beta\,\tau}\,\mathcal{Q}\mathcal{J}(\mathbf{y})\mathcal{M}^{-1}\mathcal{J}(\mathbf{y})^T\mathcal{Q}.$$

The approach we use to approximate this matrix follows the matching strategy introduced in [19, 20, 26, 27], where we approximate S by

$$\widehat{S}_1 = \big(\mathcal{Q}\mathcal{A}(\mathbf{m}) + \mathcal{M}_1\big)\widehat{\mathcal{M}}_{N_t}^{-1}\big(\mathcal{A}(\mathbf{m})^T\mathcal{Q} + \mathcal{M}_2\big), \tag{15}$$

with the aim that the term $\mathcal{M}_1\widehat{\mathcal{M}}_{N_t}^{-1}\mathcal{M}_2$ accurately captures the second term of the exact Schur complement, that is:

$$\mathcal{M}_1\widehat{\mathcal{M}}_{N_t}^{-1}\mathcal{M}_2 \approx \frac{1}{\beta\,\tau}\,\mathcal{Q}\mathcal{J}(\mathbf{y})\mathcal{M}^{-1}\mathcal{J}(\mathbf{y})^T\mathcal{Q}.$$

A possible selection of the matrices $\mathcal{M}_1$, $\mathcal{M}_2$ is as follows:

$$\mathcal{M}_1 = \mathcal{M}_2^T = \frac{1}{\sqrt{\beta\,\tau}}\,\mathcal{Q}\mathcal{J}(\mathbf{y})\mathcal{M}^{-1/2}\widehat{\mathcal{M}}_{N_t}^{1/2}, \tag{16}$$

whereupon $\widehat{S}_1$ is invertible provided $\mathcal{A}(\mathbf{m})^T\mathcal{Q} + \mathcal{M}_2$ is, which as $\mathcal{Q}$, $\widehat{\mathcal{M}}_{N_t}$ are scaled identity operators unrelated to the PDE operators $\mathcal{A}(\mathbf{m})$, $\mathcal{J}(\mathbf{y})$, will be the case almost surely. A further saving in the required computational cost may be achieved by replacing these matrices with the diagonal approximations:

$$\mathcal{M}_1 = \mathcal{M}_2^T = \frac{1}{\sqrt{\beta\,\tau}}\,\mathcal{Q}\mathcal{J}(\mathbf{y})\left[\mathrm{diag}\big(\mathcal{M}\big)\right]^{-1/2}\left[\mathrm{diag}\big(\widehat{\mathcal{M}}_{N_t}\big)\right]^{1/2},$$

which leads to a preconditioner that may be applied cheaply in practice.

Whereas the effectiveness of this Schur complement approximation will inevitably depend to some extent on the numerical behaviour of the solution at each Newton step, the following observation may be readily made (based on the methodology of [20]), guaranteeing the robustness of the smallest eigenvalue of the preconditioned Schur complement in an ideal setting:

Lemma 1. *The eigenvalues λ of $\widehat{S}_1^{-1}S$ satisfy $\lambda \geq \frac{1}{2}$, where $\widehat{S}_1$ is as defined by* (15)*, and $\mathcal{M}_1$, $\mathcal{M}_2$ are given in* (16)*.*

PROOF. Due to the symmetry and positive definiteness of S and $\widehat{S}_1$, which is observed due to $\widehat{\mathcal{M}}_{N_t}$ being symmetric positive definite by construction, the eigenvalues may be bounded by the Rayleigh quotient

$$R := \frac{\mathbf{v}^T S\mathbf{v}}{\mathbf{v}^T\widehat{S}_1\mathbf{v}} = \frac{\mathbf{v}^T\left(\mathcal{Q}\mathcal{A}(\mathbf{m})\widehat{\mathcal{M}}_{N_t}^{-1}\mathcal{A}(\mathbf{m})^T\mathcal{Q} + \beta^{-1}\tau^{-1}\mathcal{Q}\mathcal{J}(\mathbf{y})\mathcal{M}^{-1}\mathcal{J}(\mathbf{y})^T\mathcal{Q}\right)\mathbf{v}}{\mathbf{v}^T\big(\mathcal{Q}\mathcal{A}(\mathbf{m}) + \mathcal{M}_1\big)\widehat{\mathcal{M}}_{N_t}^{-1}\big(\mathcal{A}(\mathbf{m})^T\mathcal{Q} + \mathcal{M}_2\big)\mathbf{v}}.$$

We now observe that we may write

$$R = \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})}, \quad \text{where } \mathbf{a} = \widehat{\mathcal{M}}_{N_t}^{-1/2}\mathcal{A}(\mathbf{m})^T\mathcal{Q}\mathbf{v},\ \mathbf{b} = \frac{1}{\sqrt{\beta\,\tau}}\,\mathcal{M}^{-1/2}\mathcal{J}(\mathbf{y})^T\mathcal{Q}\mathbf{v}.$$

Simple manipulation therefore tells us that

$$\frac{1}{2}(\mathbf{a}-\mathbf{b})^T(\mathbf{a}-\mathbf{b}) \geq 0 \quad \Leftrightarrow \quad \mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b} \geq \frac{1}{2}(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b}) \quad \Leftrightarrow \quad R \geq \frac{1}{2},$$

which leads to the result. □

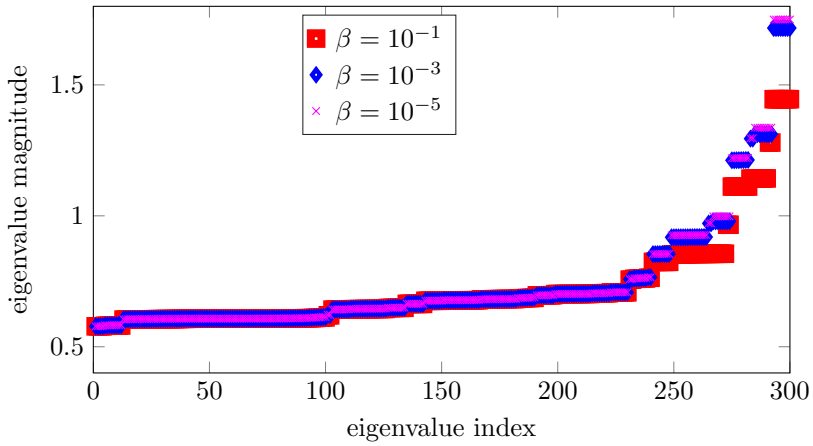
We highlight that, although the lower bound for the eigenvalues of $\widehat{S}_1^{-1}S$ can be analysed in detail, the magnitude of the largest eigenvalue will depend on the precise behaviour of the Newton iterates, which we cannot control in general. To provide an illustration of the overall eigenvalue distribution, we present in Fig. 2 the eigenvalues for a particular test problem, for a range of problem sizes and values of β. The eigenvalues are found to become more clustered for finer grids, with the magnitude of the largest eigenvalues fairly robust to changes in regularisation parameter.

Applying our approximations of the $(1,1)$ block and Schur complement leads to a preconditioner of the form

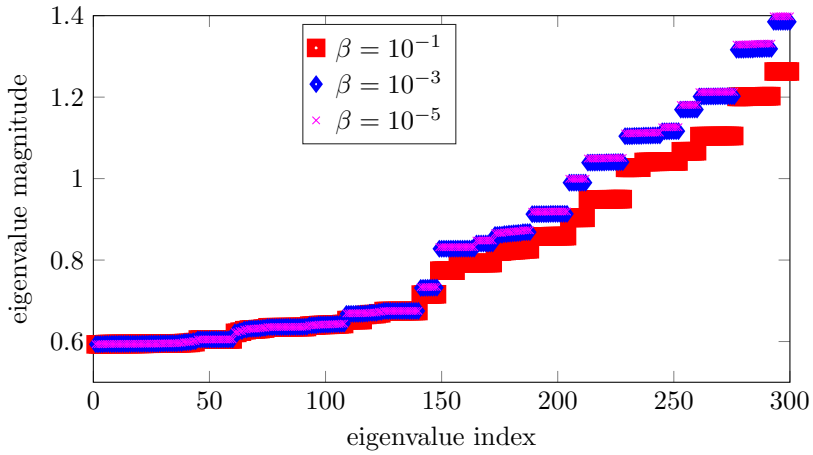
$$P_1 = \begin{bmatrix} \widehat{\mathcal{M}}_{N_t} & 0 & 0 \\ 0 & \beta\,\tau\mathcal{M} & 0 \\ 0 & 0 & \widehat{S}_1 \end{bmatrix}, \tag{17}$$

which we can then apply within a Krylov subspace method.

We highlight that similar ideas may be applied to the matrix system (8) arising from the optimise-then-discretise setting. In more detail, one may apply



(a) $n_x = 8$



(b) $n_x = 16$

Figure 2: Eigenvalues of $\widehat{S}_1^{-1}S$ for an imaging test problem, where finite difference nodes are equally distributed in each spatial direction. We vary the number of spatial degrees of freedom from $n_x = 8$ (left) to $n_x = 16$ (right). We then vary the regularisation parameter β, indicated by the three different colours. The horizontal axis gives the index of the largest 300 eigenvalues, and the vertical axis their corresponding magnitudes.

preconditioners of the form

$$\begin{bmatrix} \widehat{\mathcal{M}}_{N_t} & 0 & 0 \\ 0 & \beta\tau\mathcal{M} & 0 \\ 0 & 0 & \widehat{S}_{1,\mathrm{OTD}} \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \widehat{\mathcal{M}}_{N_t} & 0 & 0 \\ \mathcal{G}(\mathbf{p})^T & \beta\tau\mathcal{M} & 0 \\ \mathcal{A}(\mathbf{m}) & \mathcal{J}(\mathbf{y}) & -\widehat{S}_{1,\mathrm{OTD}} \end{bmatrix}.$$

Here, $\widehat{S}_{1,\mathrm{OTD}}$ can be chosen to approximate the Schur complement of the matrix

system obtained by setting $\mathcal{G}(\mathbf{y}) = 0$, for example. That is,

$$\begin{aligned}\widetilde{S} &:= \mathcal{A}(\mathbf{m})\widehat{\mathcal{M}}_{N_t}^{-1}\mathcal{B}(\mathbf{m}) + \frac{1}{\beta\,\tau}\,\mathcal{J}(\mathbf{y})\mathcal{M}^{-1}\mathcal{J}(\mathbf{y})^T \\ &\approx \left(\mathcal{A}(\mathbf{m}) + \frac{1}{\sqrt{\beta\,\tau}}\,\mathcal{M}_{1,\mathrm{OTD}}\right)\widehat{\mathcal{M}}_{N_t}^{-1}\left(\mathcal{B}(\mathbf{m}) + \frac{1}{\sqrt{\beta\,\tau}}\,\mathcal{M}_{2,\mathrm{OTD}}\right) =: \widehat{S}_{1,\mathrm{OTD}},\end{aligned}$$

with $\mathcal{M}_{1,\mathrm{OTD}}$, $\mathcal{M}_{2,\mathrm{OTD}}$ chosen such that

$$\mathcal{M}_{1,\mathrm{OTD}}\widehat{\mathcal{M}}_{N_t}^{-1}\mathcal{M}_{2,\mathrm{OTD}} \approx \mathcal{J}(\mathbf{y})\mathcal{M}^{-1}\mathcal{J}(\mathbf{y})^T.$$

Note. The derivation for the preconditioner (17) has been based on the cost functional (2), whereupon the highly singular matrix $\gamma^{-1}\mathcal{M}_{N_t}$ is approximated by $\widehat{\mathcal{M}}_{N_t}$. If instead the cost functional (1) is considered (with $\delta > 0$), the corresponding matrix in the $(1,1)$ block of (13) is $\mathcal{M}_{1,1} := \gamma^{-1}\mathcal{M}_{N_t} + \delta\,\tau\bar{\mathcal{M}}$, which is now invertible. Therefore, when deriving an analogous preconditioner P_1 for this problem setup, the matrix $\widehat{\mathcal{M}}_{N_t}$ must be replaced with $\mathcal{M}_{1,1}$ on all occasions.

4.2. Second preconditioner

We now derive a second block preconditioner, based largely on results in [28]. We commence by considering the following permutation of the matrix to be solved:

$$\Pi\begin{bmatrix} \gamma^{-1}\mathcal{M}_{N_t} & 0 & \mathcal{A}(\mathbf{m})^T\mathcal{Q} \\ 0 & \beta\,\tau\mathcal{M} & \mathcal{J}(\mathbf{y})^T\mathcal{Q} \\ \mathcal{Q}\mathcal{A}(\mathbf{m}) & \mathcal{Q}\mathcal{J}(\mathbf{y}) & 0 \end{bmatrix} = \begin{bmatrix} \mathcal{Q}\mathcal{A}(\mathbf{m}) & \mathcal{Q}\mathcal{J}(\mathbf{y}) & 0 \\ 0 & \beta\,\tau\mathcal{M} & \mathcal{J}(\mathbf{y})^T\mathcal{Q} \\ \gamma^{-1}\mathcal{M}_{N_t} & 0 & \mathcal{A}(\mathbf{m})^T\mathcal{Q} \end{bmatrix}, \tag{18}$$

where the permutation matrix is given by

$$\Pi := \begin{bmatrix} 0 & 0 & I \\ 0 & I & 0 \\ I & 0 & 0 \end{bmatrix}.$$

The matrix (18) is now of general block structure (14), with

$$A = \begin{bmatrix} \mathcal{Q}\mathcal{A}(\mathbf{m}) & \mathcal{Q}\mathcal{J}(\mathbf{y}) \\ 0 & \beta\,\tau\mathcal{M} \end{bmatrix}, \quad B = \begin{bmatrix} 0 & \mathcal{Q}\mathcal{J}(\mathbf{y}) \end{bmatrix}, \quad C = \begin{bmatrix} \gamma^{-1}\mathcal{M}_{N_t} & 0 \end{bmatrix},$$

and a non-zero $(2,2)$ block given by $\mathcal{A}(\mathbf{m})^T\mathcal{Q}$.

We may then consider the right preconditioner

$$\widetilde{P} = \begin{bmatrix} \mathcal{Q}\mathcal{A}(\mathbf{m}) & \mathcal{Q}\mathcal{J}(\mathbf{y}) & 0 \\ 0 & \beta\,\tau\mathcal{M} & 0 \\ \gamma^{-1}\mathcal{M}_{N_t} & 0 & -\widehat{S}_2 \end{bmatrix},$$

with its inverse $\widetilde{P}^{-1}$ given by

$$\begin{bmatrix} \mathcal{A}(\mathbf{m})^{-1}\mathcal{Q}^{-1} & -\beta^{-1}\tau^{-1}\mathcal{A}(\mathbf{m})^{-1}\mathcal{J}(\mathbf{y})\mathcal{M}^{-1} & 0 \\ 0 & \beta^{-1}\tau^{-1}\mathcal{M}^{-1} & 0 \\ \gamma^{-1}\widehat{S}_2^{-1}\mathcal{M}_{N_t}\mathcal{A}(\mathbf{m})^{-1}\mathcal{Q}^{-1} & -\gamma^{-1}\beta^{-1}\tau^{-1}\widehat{S}_2^{-1}\mathcal{M}_{N_t}\mathcal{A}(\mathbf{m})^{-1}\mathcal{J}(\mathbf{y})\mathcal{M}^{-1} & -\widehat{S}_2^{-1} \end{bmatrix}.$$

The matrix $\widehat{S}_2$ is designed to approximate the Schur complement of the *permuted matrix system*, that is

$$\widehat{S}_2 \approx S = \mathcal{A}(\mathbf{m})^T\mathcal{Q} + \frac{1}{\beta\,\tau\,\gamma}\,\mathcal{M}_{N_t}\mathcal{A}(\mathbf{m})^{-1}\mathcal{J}(\mathbf{y})\mathcal{M}^{-1}\mathcal{J}(\mathbf{y})^T\mathcal{Q}.$$

Let us now reapply the permutation to the preconditioned system (that is to say we propose a preconditioner P_2 for the matrix (13) such that $P_2^{-1} = \widetilde{P}^{-1}\Pi$), and therefore obtain that P_2^{-1} is given by

$$\begin{bmatrix} 0 & -\beta^{-1}\tau^{-1}\mathcal{A}(\mathbf{m})^{-1}\mathcal{J}(\mathbf{y})\mathcal{M}^{-1} & \mathcal{A}(\mathbf{m})^{-1}\mathcal{Q}^{-1} \\ 0 & \beta^{-1}\tau^{-1}\mathcal{M}^{-1} & 0 \\ -\widehat{S}_2^{-1} & -\gamma^{-1}\beta^{-1}\tau^{-1}\widehat{S}_2^{-1}\mathcal{M}_{N_t}\mathcal{A}(\mathbf{m})^{-1}\mathcal{J}(\mathbf{y})\mathcal{M}^{-1} & \gamma^{-1}\widehat{S}_2^{-1}\mathcal{M}_{N_t}\mathcal{A}(\mathbf{m})^{-1}\mathcal{Q}^{-1} \end{bmatrix}. \tag{19}$$

Applying the preconditioner is in fact more straightforward than it currently appears. To compute a vector $\mathbf{v} = P_2^{-1}\mathbf{w}$, where $\mathbf{v} := \left[\mathbf{v}_1^T,\ \mathbf{v}_2^T,\ \mathbf{v}_3^T\right]^T$, $\mathbf{w} := \left[\mathbf{w}_1^T,\ \mathbf{w}_2^T,\ \mathbf{w}_3^T\right]^T$, we first see from the second block of P_2^{-1} that

$$\beta^{-1}\tau^{-1}\mathcal{M}^{-1}\mathbf{w}_2 = \mathbf{v}_2.$$

The first equation derived from (19) then gives that

$$\begin{aligned} & -\beta^{-1}\tau^{-1}\mathcal{A}(\mathbf{m})^{-1}\mathcal{J}(\mathbf{y})\mathcal{M}^{-1}\mathbf{w}_2 + \mathcal{A}(\mathbf{m})^{-1}\mathcal{Q}^{-1}\mathbf{w}_3 = \mathbf{v}_1 \\ \Rightarrow \quad & -\mathcal{A}(\mathbf{m})^{-1}\mathcal{J}(\mathbf{y})\mathbf{v}_2 + \mathcal{A}(\mathbf{m})^{-1}\mathcal{Q}^{-1}\mathbf{w}_3 = \mathcal{A}(\mathbf{m})^{-1}\big(\mathcal{Q}^{-1}\mathbf{w}_3 - \mathcal{J}(\mathbf{y})\mathbf{v}_2\big) = \mathbf{v}_1, \end{aligned}$$

and using this we can write the last equation in (19) as

$$\begin{aligned} & -\widehat{S}_2^{-1}\mathbf{w}_1 - \gamma^{-1}\widehat{S}_2^{-1}\mathcal{M}_{N_t}\big(\beta^{-1}\tau^{-1}\mathcal{A}(\mathbf{m})^{-1}\mathcal{J}(\mathbf{y})\mathcal{M}^{-1}\mathbf{w}_2 - \mathcal{A}(\mathbf{m})^{-1}\mathcal{Q}^{-1}\mathbf{w}_3\big) \\ & \qquad\qquad = \mathbf{v}_3 \\ \Rightarrow \quad & -\widehat{S}_2^{-1}\mathbf{w}_1 + \gamma^{-1}\widehat{S}_2^{-1}\mathcal{M}_{N_t}\big(\mathcal{A}(\mathbf{m})^{-1}\mathcal{Q}^{-1}\mathbf{w}_3 - \mathcal{A}(\mathbf{m})^{-1}\mathcal{J}(\mathbf{y})\mathbf{v}_2\big) \\ & \qquad\qquad = \widehat{S}_2^{-1}\big(\gamma^{-1}\mathcal{M}_{N_t}\mathbf{v}_1 - \mathbf{w}_1\big) = \mathbf{v}_3. \end{aligned}$$

Therefore, in order to solve a system with the preconditioner P_2, we need to solve for the matrix $\mathcal{M}$, which is certainly invertible, as well as the matrix $\mathcal{Q}\mathcal{A}(\mathbf{m})$. What remains is the construction of the approximation $\widehat{S}_2$ of the Schur complement. In more detail, we suggest the use of a similar matching strategy as above, to write

$$\widehat{S}_2 = \big(\mathcal{A}(\mathbf{m})^T\mathcal{Q} + \mathcal{M}_l\big)\mathcal{A}(\mathbf{m})^{-1}\mathcal{Q}^{-1}\big(\mathcal{Q}\mathcal{A}(\mathbf{m}) + \mathcal{M}_r\big),$$

where

$$\mathcal{M}_l \mathcal{A}(\mathbf{m})^{-1} \mathcal{Q}^{-1} \mathcal{M}_r \approx \frac{1}{\beta\,\tau\,\gamma}\, \mathcal{M}_{N_t} \mathcal{A}(\mathbf{m})^{-1} \mathcal{J}(\mathbf{y}) \mathcal{M}^{-1} \mathcal{J}(\mathbf{y})^T \mathcal{Q}.$$

Such an approximation may be achieved if, for example,

$$\mathcal{M}_l = \frac{1}{\gamma}\, \mathcal{M}_{N_t},$$
$$\mathcal{M}_r = \frac{1}{\beta\,\tau}\, \mathcal{Q}\mathcal{J}(\mathbf{y}) \mathcal{M}^{-1} \mathcal{J}(\mathbf{y})^T \mathcal{Q} \quad \text{or} \quad \frac{1}{\beta\,\tau}\, \mathcal{Q}\mathcal{J}(\mathbf{y}) \left[\text{diag}(\mathcal{M})\right]^{-1} \mathcal{J}(\mathbf{y})^T \mathcal{Q},$$

and we thus build such approximations into our preconditioner P_2. We highlight that at no stage in applying P_2^{-1} does one have to apply a representation of the inverse of the often highly singular matrix $\mathcal{M}_{N_t}$, which is a key advantage of the preconditioner P_2 over P_1.

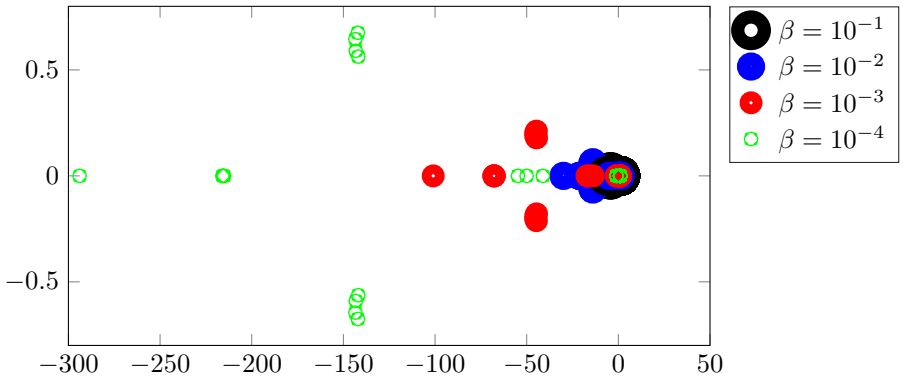
In Fig. 3 we illustrate the distribution of the eigenvalues of the preconditioned matrix with respect to changes in the mesh parameter and the regularisation parameter β. It can be seen that there is some dependence on the regularisation parameter, while it seems that the mesh parameter does not influence the distribution of the eigenvalues dramatically. While this is not a theoretical justification, it indicates that the method proposed is a promising candidate for the case of partial observations of the variable y.

We highlight that our methodology for constructing preconditioners of the form P_1 and P_2 may be readily tailored to the matrix system (12) arising from the alternative problem formulation discussed in Section 3.3.

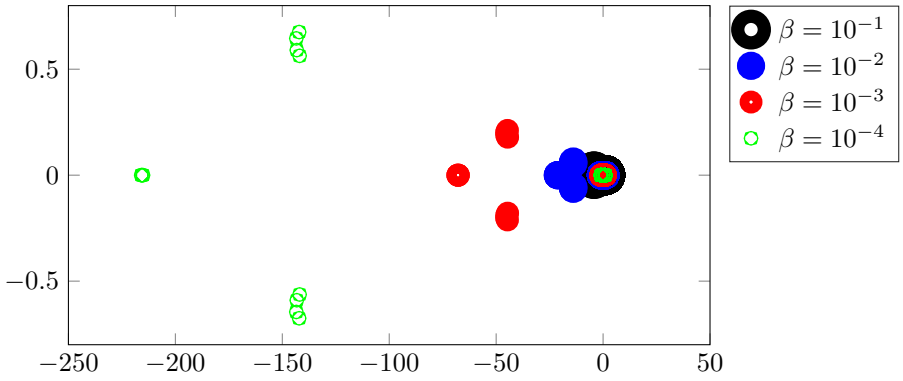
5. Numerical results

We now present the results of a number of numerical experiments, making use of the finite difference discretisation outlined in Section 3. All experiments are implemented in MATLAB R2017b.

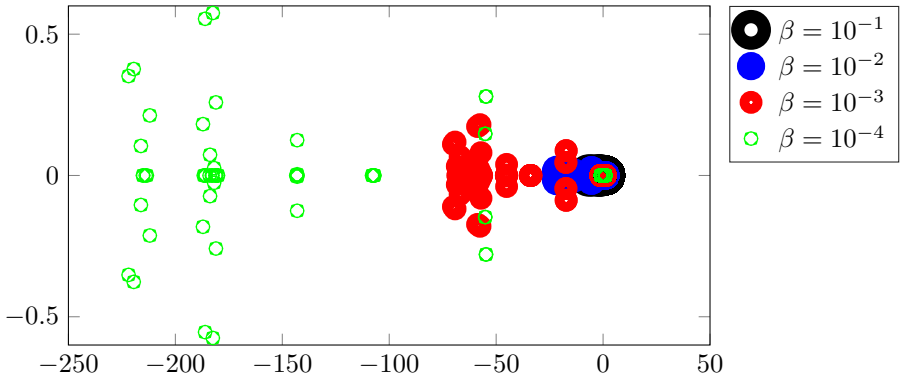
For our first test we consider the minimisation of (1) or (2) subject to the PDE constraint (3), and employ the Gauss–Newton scheme in the finite difference setting. The parameters and operators are chosen to be $\gamma = 1$ and $Q = \text{blkdiag}(I, I)$. The regularisation parameter β is typically varied in our experiments. We examine the performance of the preconditioners presented in Sections 4.1 and 4.2, and recall that the $(1,1)$ block of the system matrix governing (5) is highly singular if the cost functional (2) is considered. We use the same discretisation level in time as we use for the spatial domain for the finite difference approach, and compare 5 spatial mesh levels for the synthetic data. For the image data though we use ten time-steps and the same number of intermediate images, i.e., one image for every time-step. The image data is as depicted in Fig. 1. We here use 100×100 pixel grayscale images, where the pixel values are scaled to be between zero and one. We apply the GMRES method [29] as the iterative scheme for the Gauss–Newton system. This method is stopped when a tolerance (we use 10^{-6}) for the relative residual norm of the



(a) $n_x = 2$



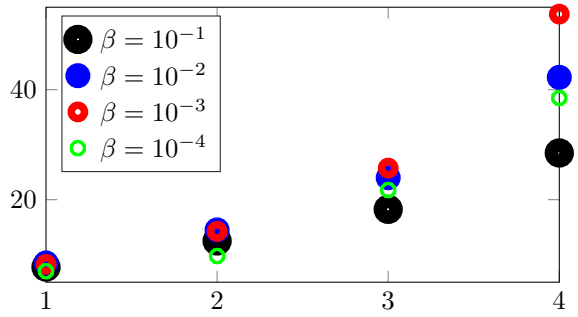
(b) $n_x = 4$



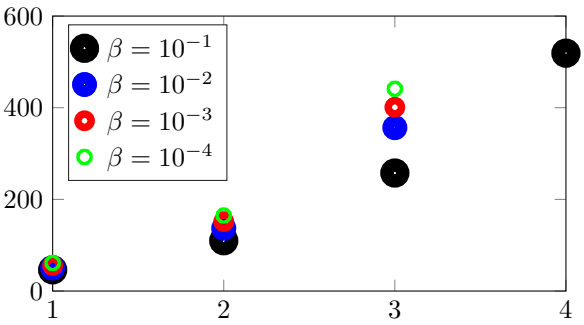
(c) $n_x = 8$

Figure 3: Eigenvalues of $P_2^{-1}\mathcal{W}$, with $\mathcal{W}$ the saddle point matrix (13). The real parts are displayed along the horizontal axis, with the imaginary parts on the vertical axis. We provide results for three different mesh parameters $h = \frac{1}{n_x}$.

linear system (5) is reached, starting from an all-zero initial guess. Note that in MATLAB's implementation of GMRES, the residual is measured in the Euclidean norm. The outer Gauss–Newton scheme is stopped once the relative Euclidean distance between consecutive iterates falls below 10^{-4}. In Fig. 4 we illustrate the performance of the preconditioner P_1 when applied to the full observation problem (i.e., the cost functional (1)), with $\delta = 1$. We give results for a variety of mesh parameters and regularisation parameters. In Fig. 5, we illustrate the



(a) Preconditioned

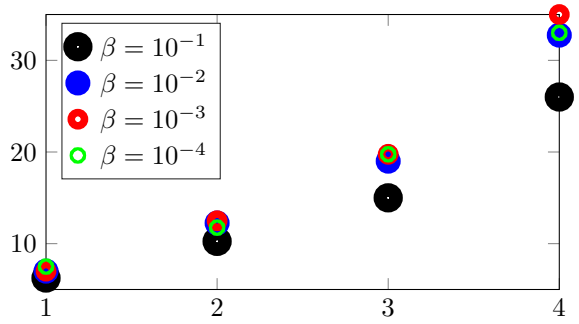


(b) Unpreconditioned

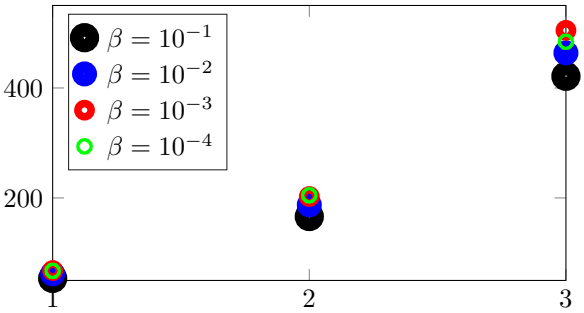
Figure 4: Average iteration numbers (vertical axis) per Gauss–Newton step of the GMRES method with preconditioner P_1, for several mesh parameters (horizontal axis) and different choices of the regularisation parameter β.

performance of the preconditioner P_2 for the partial observation case (i.e., the cost functional (2)), when applied with a variety of mesh parameters and regularisation parameters. We compare this approach to the unpreconditioned case, which is only competitive for very coarse meshes. We observe, for both full observations and partial observations, that the Gauss–Newton scheme converges robustly in 4 or 5 iterations.

We also report results for an optical flow problem of the form in [11, 30], and take the objective function to be given by (1) with varying δ, $\beta = 10^{-1}$, and $\gamma = 10$. The tolerances are set to be 10^{-3} and 10^{-2} for the linear and nonlinear solver, respectively. We here assume that $\bar{y}(\vec{x}, 1) = y_1$, and when discretised in time



(a) Preconditioned



(b) Unpreconditioned

Figure 5: Average iteration numbers (vertical axis) per Gauss–Newton step of the GMRES method with preconditioner P_2, for several mesh parameters (horizontal axis) and different choices of the regularisation parameter β. When applying the unpreconditioned method, we terminate the computation for the finest mesh level as the computation time was too high.

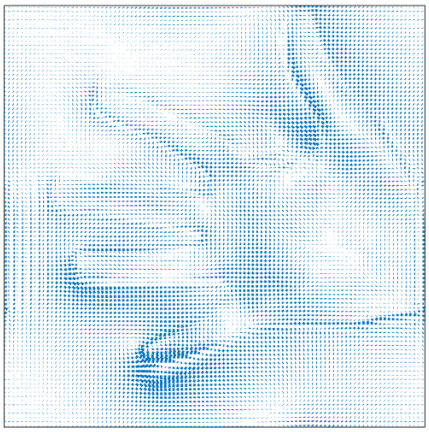
$\bar{y}(\vec{x}, t_i) = y_i$ corresponds to a given image. As intermediate values for the desired state we chose the intermediate images from [31]. It is clear that this setup is covered by our previous discussion, and the matrix representing contributions of the state variable to the objective function is given by $\mathcal{M}_{1,1} = \gamma^{-1}\mathcal{M}_{N_t} + \delta\,\tau\bar{\mathcal{M}}$. One can readily apply the preconditioning techniques introduced in Section 4, and we consider the implementation of the preconditioner P_1. We show the results for our methodology in Fig. 6.

We observe robustness with respect to the matrix dimension and the parameters involved in the problem setup, both in terms of the Gauss–Newton iterations required, and the number of iterations of the preconditioned iterative method. This indicates the effectiveness of our preconditioning strategy for a range of problem statements.

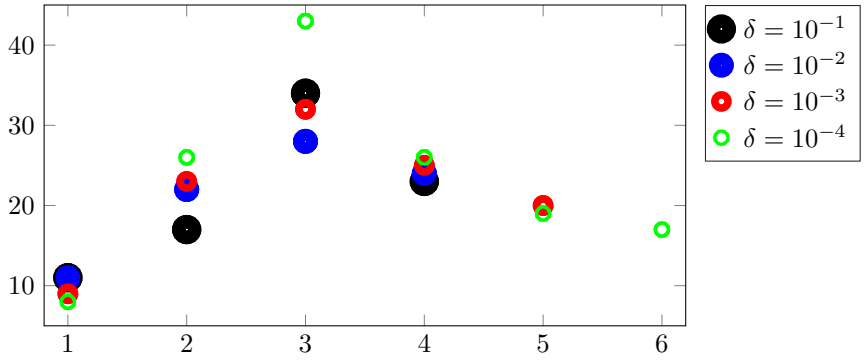
6. Conclusion

In this paper, we have presented numerical methods for the solution of optimal transport problems arising in image metamorphosis. We have discussed

(a) Computed state at time-step 5



(b) Computed control at time-step 5



(c) Varying δ

Figure 6: Results for the preconditioned iterative method with preconditioner P_1, applied to the optical flow problem. We show an instance of the control and the state, as well as the number of GMRES iterations for each Gauss–Newton step, for varying regularisation parameter δ.

the application of Newton and Gauss–Newton methods, using finite difference schemes for the discretisation of the optimality conditions. We presented fast and effective preconditioners which may be applied within Krylov subspace methods to solve the resulting matrix systems, with a focus on the large dimensions of the matrices when many time-steps are taken to solve the systems of PDEs. Encouraging numerical results indicate the potency of the solvers presented.

Acknowledgements. The authors would like to thank two anonymous referees for their helpful comments and suggestions. JWP gratefully acknowledges support from the Engineering and Physical Sciences (EPSRC) Fellowship EP/M018857/2.

References

[1] C. Villani, Optimal Transport: Old and New, Vol. 338, Springer Science & Business Media, 2008.

[2] D. Kuzmin, A Guide to Numerical Methods for Transport Equations, Friedrich-Alexander-Universität Erlangen–Nürnberg, `http://www.mathematik.uni-dortmund.de/~kuzmin/Transport.pdf`, 2010.

[3] R. J. LeVeque, Numerical Methods for Conservation Laws. 2nd ed., Basel: Birkhäuser, 1992.

[4] K. Ito, K. Kunisch, Lagrange Multiplier Approach to Variational Problems and Applications, Vol. 15 of Advances in Design and Control, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2008.

[5] F. Tröltzsch, Optimal Control of Partial Differential Equations: Theory, Methods and Applications, American Mathematical Society, 2010.

[6] M. Benzi, E. Haber, L. Taralli, A preconditioning technique for a class of PDE-constrained optimization problems, Advances in Computational Mathematics 35 (2011) 149–173.

[7] L. V. Kantorovich, On a problem of Monge, Journal of Mathematical Sciences 133 (4) (2006) 1383–1383.

[8] L. V. Kantorovitch, On the translocation of masses, Management Science 5 (1) (1958) 1–4.

[9] J.-D. Benamou, Y. Brenier, A computational fluid mechanics solution to the Monge–Kantorovich mass transfer problem, Numerische Mathematik 84 (3) (2000) 375–393.

[10] R. Andreev, O. Scherzer, W. Zulehner, Simultaneous optical flow and source estimation: Space–time discretization and preconditioning, Applied Numerical Mathematics 96 (2015) 72–81.

[11] A. Borzi, K. Ito, K. Kunisch, Optimal control formulation for determining optical flow, SIAM Journal on Scientific Computing 24 (3) (2003) 818–847.

[12] A. Bruhn, J. Weickert, C. Schnörr, Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods, International Journal of Computer Vision 61 (3) (2005) 211–231.

[13] E. Haber, J. Modersitzki, A multilevel method for image registration, SIAM Journal on Scientific Computing 27 (5) (2006) 1594–1607.

[14] B. K. P. Horn, B. G. Schunck, Determining optical flow, Artificial Intelligence 17 (1–3) (1981) 185–203.

[15] A. Mang, G. Biros, An inexact Newton–Krylov algorithm for constrained diffeomorphic image registration, SIAM Journal on Imaging Sciences 8 (2) (2015) 1030–1069.

[16] A. Mang, L. Ruthotto, A Lagrangian Gauss–Newton–Krylov solver for mass- and intensity-preserving diffeomorphic image registration, SIAM Journal on Scientific Computing 39 (5) (2017) B860–B885.

[17] E. Haber, U. M. Ascher, D. Oldenburg, On optimization techniques for solving nonlinear inverse problems, Inverse Problems 16 (5) (2000) 1263–1280.

[18] O. Axelsson, S. Farouq, M. Neytcheva, Comparison of preconditioned Krylov subspace iteration methods for PDE-constrained optimization problems, Numerical Algorithms 73 (3) (2016) 631–663.

[19] J. W. Pearson, M. Stoll, A. J. Wathen, Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems, SIAM Journal on Matrix Analysis and Applications 33 (4) (2012) 1126–1152.

[20] J. W. Pearson, A. J. Wathen, Fast iterative solvers for convection–diffusion control problems, Electronic Transactions on Numerical Analysis 40 (2013) 294–310.

[21] W. Zulehner, Non-standard norms and robust estimates for saddle point problems, SIAM Journal on Matrix Analysis and Applications 32 (2011) 536–560.

[22] J. Nocedal, S. J. Wright, Numerical Optimization, 2nd ed., Springer Series in Operations Research and Financial Engineering, Springer, New York, 2006.

[23] Y. A. Kuznetsov, Efficient iterative solvers for elliptic finite element problems on nonmatching grids, Russian Journal of Numerical Analysis and Mathematical Modelling 10 (1995) 187–211.

[24] M. F. Murphy, G. H. Golub, A. J. Wathen, A note on preconditioning for indefinite linear systems, SIAM Journal on Scientific Computing 21 (6) (2000) 1969–1972.

[25] I. C. F. Ipsen, A note on preconditioning non-symmetric matrices, SIAM Journal on Scientific Computing 23 (3) (2001) 1050–1051.

[26] J. W. Pearson, M. Stoll, Fast iterative solution of reaction–diffusion control problems arising from chemical processes, SIAM Journal on Scientific Computing 35 (2013) B987–B1009.

[27] J. W. Pearson, A. J. Wathen, A new approximation of the Schur complement in preconditioners for PDE-constrained optimization, Numerical Linear Algebra with Applications 19 (2012) 816–829.

[28] P. Benner, S. Dolgov, A. Onwunta, M. Stoll, Low-rank solvers for unsteady Stokes–Brinkman optimal control problem with random data, Computer Methods in Applied Mechanics and Engineering 304 (2016) 26–54.

[29] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on Scientific Computing, 7 (3) (1986) 856–869.

[30] A. Borzi, K. Ito, K. Kunisch, An optimal control approach to optical flow computation, International Journal for Numerical Methods in Fluids 40 (1–2) (2002) 231–240.

[31] M. J. Black, A. D. Jepson, Eigentracking: Robust matching and tracking of articulated objects using a view-based representation, International Journal of Computer Vision 26 (1) (1998) 63–84.