PDE-CONSTRAINED OPTIMIZATION MODELS AND PSEUDOSPECTRAL METHODS FOR MULTISCALE PARTICLE DYNAMICS

MILDRED ADUAMOAH*, BENJAMIN D. GODDARD†, JOHN W. PEARSON‡, AND JONNA C. RODEN§

Abstract. Very abstract at the moment.

Key words. PDE-constrained optimization; Multiscale particle dynamics; Pseudospectral methods; Numerical continuation

AMS subject classifications. 35Q70, 35Q93, 49J20, 65N35, 82C22

1. Introduction.

Recent work on particle dynamics applications within a PDE-constrained optimization framework [2, 3, 4, 10, 12, 17]

Iterative solvers for mean-field games [1, 6]

New things in this paper: modelling approach linking to statistical mechanics literature; pseudospectral method for control problem; continuation algorithm

Minor: dealing with Neumann/no-flux BCs(??); new test problems; extension to software library

Advantages of our approach: not solved on a torus (unlike many multidimensional problems) which is relatively easy; more accurate (pseudospectral) method vs finite difference methods of previous work; our "trick" allows us to ensure density functions are positive, unlike standard spectral-type methods

Challenges: need good initial guess, especially for Neumann/no-flux condition, which is a common challenge in numerical analysis

2. Background.

2.1. Multiscale particle dynamics.

The dynamics of many systems can be accurately described by interacting particles or agents. Examples of such physical particles range in scale from electrons in atoms and molecules [39], through biological cells in tissues [5], up to planets and stars in galaxies [9]. Other individual-based models include animals undergoing flocking and swarming [46], pedestrians walking [18], or people who interact and thus change their opinions [26].

In principle, such situations can be modelled by differential equations for the 'state' (e.g. position, momentum, opinion) of each individual. However, the challenge here is that physical systems typically have huge numbers of particles (e.g., 10^{25} molecules in a litre of water) and, as such, are beyond the treatment of standard numerical methods, both in terms of storage and processor time. For N particles, typical algorithms scale as N^2 or N^3, which prevents direct computation for more than, say, $\mathcal{O}(10^4)$ particles. It is clear from the separation of computationally realistic

*The School of Mathematics and Maxwell Institute for Mathematical Sciences, The University of Edinburgh, Edinburgh, EH9 3FD, UK (s1783812@sms.ed.ac.uk)

†The School of Mathematics and Maxwell Institute for Mathematical Sciences, The University of Edinburgh, Edinburgh, EH9 3FD, UK (b.goddard@ed.ac.uk)

‡The School of Mathematics and Maxwell Institute for Mathematical Sciences, The University of Edinburgh, Edinburgh, EH9 3FD, UK (j.pearson@ed.ac.uk)

§The School of Mathematics and Maxwell Institute for Mathematical Sciences, The University of Edinburgh, Edinburgh, EH9 3FD, UK (J.C.Roden@sms.ed.ac.uk)

problems to the size of physical systems that this issue cannot be overcome through the sequential improvement of computer hardware.

An additional complication of directly solving the dynamics of such systems, e.g. through Newtonian dynamics, is the sensitive dependence on initial conditions [25]. For many physical systems, it is unreasonable to assume that one knows the exact initial conditions for each particle As such, one is interested not in a particular realisation of the dynamics, but rather in an 'average' behaviour, which is typical for the system.

Both of these challenges suggest that it would be prudent to instead study the dynamics through a statistical mechanics approach, in which one is interested in the macroscopic quantities, rather than individual realisations. However, this approach comes with its own challenges and drawbacks.

The first is that, at least without additional simplifying approximations, the resulting equations are no easier to solve than the underlying particle dynamics. For example, instead of treating the Langevin stochastic differential equation (SDE), which formally scales computationally as N^2, one may treat the corresponding Fokker-Planck (forward Kolmogorov / Smoluchowski) equation, which is a partial differential equation in dN dimensions, where d is the number of degrees of freedom of the one-particle phase space (typically 6 when including momentum, and 3 when only considering the particle positions). A standard approach would then be to discretize each degree of freedom, reducing the PDE to a system of coupled ODEs, which may then, in principle, be solved numerically. The issue here lies with the curse of dimensionality: for M points in each degree of freedom, one requires a total of M^{dN} points. Taking, for the sake of argument, $M = 10$ points and $N = 10$ particles in three dimensions, then the total number of points required is 10^{30}, which is far too many for a reasonable computation.

A common approach to overcome the is to use 'coarse-graining', which reduces the dimensionality of the system, generally at the cost of a loss of accuracy or physical effects, and the introduction of unconstrained approximations [43]. This links to the second challenge, which concerns the multiscale nature of the problem. In many systems of interest, physically crucial effects manifest themselves on scales of the particle size, all the way up to the macroscale. Examples include volume exclusion of hard particles [13], biological cellular alignment [8], and nucleation of clusters and clouds [28]. A standard coarse-graining approach would be to ignore effects such as volume exclusion, and treat the whole system as a bulk, and hence determine quantities such as average densities and orientations [24]. Whilst this is viable in homogeneous systems close to equilibrium, it completely fails to capture heterogeneous systems, symmetry breaking, and many dynamical effects.

An extremely efficient and accurate example of coarse-graining is Dynamic Density Functional Theory (DDFT) [29, 15]. The crucial observation here is that the full N-body information in a system is a functional of the 1-body density, $\rho(\vec{x}, t)$ (i.e. the probability of finding any one particle at a given position at a given time). This is an extension of classical density functional theory (DFT) (see, e.g., the early works [19, 35] and later reviews [44, 45, 27]), which considers the equilibrium case, and is linked to the celebrated quantum version [23]. The main challenge here is that the proof is non-constructive; it is unknown how to map from ρ to the full information in the system. However, in many practical applications, it is ρ itself that is the quantity of interest. Hence it is desirable to derive equations of motion for the 1-body density.

The simplest example is the diffusion equation, which corresponds to Brownian

motion, and concerns non-interacting particles; hence the reduction to the 1-body density is trivial. We are instead concerned with systems in which the particles interact, e.g. through electrostatic forces, volume exclusion, or exchange of information. Typical DDFTs can be thought of as generalised diffusion equations of the form

$$\partial_t \rho(\vec{x}, t) = \nabla \cdot \left(\rho \nabla \frac{\delta \mathcal{F}[\rho]}{\delta \rho} \right) = \nabla \cdot \vec{j}. \tag{2.1}$$

Here $\mathcal{F}$ is the Helmholtz free energy of the system. For the non-interacting case, at equilibrium, it is given by

$$\mathcal{F}_{\mathrm{id}}[\rho] = \int \rho(\vec{x})(\log \rho(\vec{x}) - 1) \, \mathrm{d}\vec{x},$$

from which it follows that $\frac{\delta \mathcal{F}_{\mathrm{id}}[\rho]}{\delta \rho} = \nabla \rho / \rho$, which results in the diffusion equation.

For more general systems, the exact free energy is unknown (except in the special case of hard rods in one dimension [40]). As such, much effort has been devoted to determine accurate approximations of the free energy for a wide range of systems, but particular focus is given to hard spheres [36] and particles with soft interactions [22]; these cases may be combined in a perturbative manner [20]. Here we will focus on a relatively simple DDFT, which closes the equation for ρ by considering that the particles are, in average, uncorrelated. For particles which interact through a pairwise potential V_2, in an external potential field V_1, the (approximate) free energy is modelled by

$$\mathcal{F}[\rho] = \int \rho(\vec{x})(\log \rho(\vec{x}) - 1) \, \mathrm{d}\vec{x} + \int V_1(\vec{x}) \rho(\vec{x}) \, \mathrm{d}\vec{x} + \frac{1}{2} \int \int \rho(\vec{x}) \rho(\vec{x}') V_2(|\vec{x} - \vec{x}'|) \, \mathrm{d}\vec{x} \mathrm{d}\vec{x}'.$$

This is known as the mean field approximation and has been shown to be surprisingly accurate for a range of systems [7] and is known to be exact in the limit of dense systems of particles with soft interactions [30]. We note that this should be considered as a the first stepping stone on a path to treat PDE-constrained optimal control systems for general DDFTs. Such systems are highly challenging, not only due to the non-local, non-linear nature of the PDEs, but also due to the complexity of the free energy functionals. For example, Fundamental Measure Theory (FMT), which describes the interactions of systems of hard particles, requires the computation of weighted densities through convolution integrals, followed by a further integral of a complicated function of these weighted densities [36]. As such, these challenges are postponed to future work.

A final challenge we will address here is the implementation of (spatial) boundary conditions. Most physical systems are constrained in some way, often in a 'box' with impassable walls, such that the number of particles is conserved. For DDFTs, the corresponding boundary condition is $\vec{j} \cdot \vec{n} = 0$ on the boundary, where j is the flux, as in (2.1) and n is the unit normal to the boundary. Whilst this is a standard Neumann boundary condition, we note that the difficulty lies in the form of j; for interacting problems, j is non-local and, as such, so is the corresponding boundary condition. This results in an equation which is challenging to solve numerically (see Section [[REF]]).

2.2. Pseudospectral methods. There are a number of standard methods for solving DDFT-like problems. The two most common are the finite element method (FEM) and pseudospectral methods. Here we focus on the latter, but note that

the algorithm presented below [[REF]] is general and may be easily adapted to other numerical methods. The main challenge in using FEM for DDFT problems lies in their non-locality. Heuristically, the principal benefits of FEM are that it (i) produces large, but sparse matrices, leading to systems which may be efficiently solved, for example through the implementation of standard timestepping schemes and carefully-chosen preconditioners [[REF PRECONDITIONERS]]; and (ii) may be applied to complex domains through standard triangulation/meshing routines. In contrast, for non-local problems such as DDFT the corresponding matrices are not only large, but they are also dense. This prevents the use of standard numerical schemes and significantly increases the computational cost.

Recently, accurate and efficient pseudospectral methods have been developed to tackle these non-local, non-linear DDFTs [31]. Some details of the implementation will be discussed in Section [[REF]]; here we highlight the benefits and challenges. As is widely known [41, 11], pseudospectral methods are extremely accurate for problems with smooth solutions on 'nice' domains; here 'nice' roughly corresponds to domains which may be mapped to the unit square in a simple (e.g. conformal) manner. They are more challenging to apply on complex domains (although spectral elements can be seen as a compromise between FEM and pseudospectral methods), and are also of poor accuracy when the solutions are not smooth (heuristically, the accuracy is order $(1/N)^p$ where the solution is p-times differentiable.

Their use to treat DDFT problems stems from three main observations: (i) at least morally, the diffusion term present in all DDFTs should lead to smoothing of solutions for sufficiently smooth particle interactions; (ii) the matrices involved are always dense and, as such, treating non-local terms does not formally affect the numerical cost; (iii) the implementation of non-local boundary conditions may be treated via standard algebraic-differential equations solver, thus removing the need for bespoke treatments of different boundary conditions.

2.3. PDE-constrained optimization. In this section we introduce a number of PDE-constrained optimization problem structures that we will consider in a multiscale particle dynamics setting. In the following, the terms 'linear control' and 'nonlinear control' refer to the application of the control in the PDE constraint either linearly or nonlinearly. Alternatively, these are known as 'force control', since the control can be applied linearly through the source, 'force' term, and 'flow control', where the nonlinear control is the vector field, 'flow', in the advection term.

– ***Advection–diffusion nonlinear control problem:*** We commence with the following minimization problem involving L^2-norm terms within the entire space-time interval Ω, constrained by a nonlinear time-dependent advection–diffusion equation:

$$\min_{\rho,\vec{w}} \quad \frac{1}{2}\int_0^T \int_\Omega (\rho - \widehat{\rho})^2 \, \mathrm{d}x\mathrm{d}t + \frac{\beta}{2}\int_0^T \int_\Omega \|\vec{w}\|^2 \, \mathrm{d}x\mathrm{d}t \tag{2.2}$$

$$\begin{aligned} \text{s.t.} \quad \partial_t \rho - \nabla^2 \rho + \nabla \cdot (\rho \vec{w}) - \gamma \nabla_r \cdot \left(\int_\Omega \rho(r)\rho(r')\vec{K}(|r-r'|) \, \mathrm{d}r' \right) & \\ = \nabla \cdot (\rho \nabla V_{\text{ext}}) + f \quad & \text{on } \Omega \times (0,T), \\ \rho = \rho_0(\vec{x}) \quad & \text{at } t = 0, \end{aligned} \tag{2.3}$$

where $\Omega \subset \mathbb{R}^d$, $d \in \{1,2,3\}$, is some given domain with boundary $\partial\Omega$, and T is a prescribed "final time" at which the process is considered. The scalar function ρ and the vector-valued function $\vec{w}$ are the *state* and *control variables*, respectively, $\beta > 0$ is

a given *regularization parameter*, and $\widehat{\rho}(\vec{x},t)$, $V_{\text{ext}}(\vec{x},t)$, $f(\vec{x},t)$, $\rho_0(\vec{x})$ are prescribed functions corresponding to the *desired state*, *external potential*, PDE source term, and initial condition, respectively. We highlight that frequently $f(\vec{x},t)=0$, which results in conservation of mass. Additionally, there is a nonlocal integral term, in order to model interactions between individual particles, where $\vec{K}$ denotes some vector function acting on the distance between particles. When the parameter γ is set to zero, the equation reduces to a standard nonlinear advection-diffusion control problem. When γ is positive, the integral term models repulsive interactions (check with sign of K!), when γ is negative, attractive interactions. Typical values of γ are in the range of $[-1,1]$, and can be interpreted as the interaction strength of the particles.

We consider two possibilities for the boundary conditions imposed on ρ, specifically the Dirichlet boundary condition:

$$\rho = c \quad \text{on } \partial\Omega \times (0,T), \tag{2.4}$$

where $c \in \mathbb{R}$ is often zero, and that of the boundary condition:

$$\frac{\partial \rho}{\partial n} - \rho \vec{w}\cdot\vec{n} + \rho \frac{\partial V_{\text{ext}}}{\partial n} + \gamma \int_\Omega \rho(r)\rho(r') \frac{\partial \vec{K}(|r-r'|)}{\partial n}\, \mathrm{d}r' = 0 \quad \text{on } \partial\Omega \times (0,T), \tag{2.5}$$

where $\frac{\partial}{\partial n}$ denotes the derivative with respect to the normal $\vec{n}$. This is a no-flux boundary condition if $f=0$.

– ***Advection–diffusion linear control problem:*** We consider the following problem with L^2-norm terms within the entire space-time interval Ω, constrained by a nonlinear time-dependent advection–diffusion equation with linear components in a scalar control variable:

$$\min_{\rho,\vec{w}} \quad \frac{1}{2}\int_0^T \int_\Omega (\rho - \widehat{\rho})^2 \,\mathrm{d}x\mathrm{d}t + \frac{\beta}{2}\int_0^T \int_\Omega w^2 \,\mathrm{d}x\mathrm{d}t \tag{2.6}$$

$$\begin{aligned} \text{s.t.} \quad \partial_t \rho - \nabla^2 \rho - \gamma \nabla_r \cdot \left(\int_\Omega \rho(r)\rho(r') \vec{K}(|r-r'|)\, \mathrm{d}r' \right) & \\ = \nabla\cdot(\rho \nabla V_{\text{ext}}) + w + f \quad & \text{on } \Omega \times (0,T), \\ \rho = \rho_0(\vec{x}) \quad & \text{at } t = 0, \end{aligned} \tag{2.7}$$

along with the Dirichlet boundary condition (2.4) or the 'Neumann-type' boundary condition:

$$\frac{\partial \rho}{\partial n} + \rho \frac{\partial V_{\text{ext}}}{\partial n} + \gamma \int_\Omega \rho(r)\rho(r') \frac{\partial \vec{K}(|r-r'|)}{\partial n}\, \mathrm{d}r' = 0 \quad \text{on } \partial\Omega \times (0,T). \tag{2.8}$$

The parameter γ in the interaction term is interpreted as in the nonlinear problem above, and can be set to zero, to recover a standard linear advection-diffusion control problem.

[[check sign of integral term in constraint is correct - A: I think so, is in line with code.]]

3. Particle dynamics models and first-order optimality conditions. In order to obtain first-order optimality conditions for the models (2.2), and (2.6), we apply an *optimize-then-discretize method*, meaning we derive appropriate conditions

on the continuous level and then consider suitable discretization strategies. The alternative to this approach is the *discretize-then-optimize* method, however we select the former in order to obtain numerical solutions that are more faithful to the continuous first-order optimality conditions. We highlight that an area of active interest in the PDE-constrained optimization community is to construct discretization schemes such that the two approaches coincide (see [16] for a fundamental example of a problem for which different results are obtained using either method).

– ***Nonlinear control with Dirichlet boundary condition:*** We first consider the advection–diffusion constrained optimization problem (2.2) with the Dirichlet boundary condition (2.4). The interaction term is excluded ($\gamma = 0$), for readability. This leads to the continuous Lagrangian:

$$\begin{aligned}\mathcal{L}(\rho, \vec{w}, q_1, q_2) = \frac{1}{2}\int_0^T \int_\Omega (\rho - \widehat{\rho})^2 \,\mathrm{d}x\mathrm{d}t + \frac{\beta}{2}\int_0^T \int_\Omega \|\vec{w}\|^2 \,\mathrm{d}x\mathrm{d}t \\ - \int_0^T \int_\Omega \left(\partial_t \rho - \nabla^2 \rho + \nabla\cdot(\rho\vec{w}) - \nabla\cdot(\rho\nabla V_{\mathrm{ext}}) - f\right) q_1 \,\mathrm{d}x\mathrm{d}t \\ - \int_0^T \int_{\partial\Omega} \rho q_2 \,\mathrm{d}s\mathrm{d}t,\end{aligned} \tag{3.1}$$

where q_1 and q_2 correspond to the portions of the *adjoint variable* q arising in the interior of the spatial domain Ω and its boundary $\partial\Omega$, respectively.

To obtain first-order optimality conditions, we first follow standard working for deriving the *adjoint equation* for time-dependent PDE-constrained optimization, see [42, Chapter 3] for instance. We obtain that the derivative of $\mathcal{L}$ in the direction ρ must satisfy $D_\rho \mathcal{L}(\bar{\rho}, \bar{w}, q_1, q_2)\rho = 0$ for all ρ such that $\rho(\vec{x}, 0) = 0$. Now, from (3.1),

$$\begin{aligned}D_\rho \mathcal{L}(\bar{\rho}, \bar{w}, q_1, q_2)\rho = \int_0^T \int_\Omega (\bar{\rho} - \widehat{\rho})\rho \,\mathrm{d}x\mathrm{d}t \\ - \int_0^T \int_\Omega \left(\partial_t \rho - \nabla^2 \rho + \nabla\cdot(\rho\bar{w}) - \nabla\cdot(\rho\nabla V_{\mathrm{ext}})\right) q_1 \,\mathrm{d}x\mathrm{d}t \\ - \int_0^T \int_{\partial\Omega} \rho q_2 \,\mathrm{d}s\mathrm{d}t,\end{aligned}$$

whereupon upon integrating by parts and applying Green's formula, any sufficiently

smooth ρ such that $\rho(\vec{x},0)=0$ satisfies

$$\begin{aligned}
0 = & -\int_0^T\int_\Omega(-\partial_t q_1-\nabla^2 q_1-\bar{w}\cdot\nabla q_1+\nabla V_{\text{ext}}\cdot\nabla q_1+\widehat{\rho}-\bar{\rho})\rho\ \mathrm{d}x\mathrm{d}t \\
& +\int_0^T\int_\Omega\left[\nabla\cdot(q_1\nabla\rho)-\nabla\cdot(\rho\nabla q_1)-\nabla\cdot(\rho q_1\bar{w})+\nabla\cdot(\rho q_1\nabla V_{\text{ext}})\right]\mathrm{d}x\mathrm{d}t \\
& +\int_\Omega q(\vec{x},T)\rho(\vec{x},T)\ \mathrm{d}x-\int_0^T\int_{\partial\Omega} q_2\rho\ \mathrm{d}s\mathrm{d}t \\
= & -\int_0^T\int_\Omega(-\partial_t q_1-\nabla^2 q_1-\bar{w}\cdot\nabla q_1+\nabla V_{\text{ext}}\cdot\nabla q_1+\widehat{\rho}-\bar{\rho})\rho\ \mathrm{d}x\mathrm{d}t \\
& +\int_\Omega q(\vec{x},T)\rho(\vec{x},T)\ \mathrm{d}x+\int_0^T\int_{\partial\Omega} q_1\frac{\partial\rho}{\partial n}\ \mathrm{d}s\mathrm{d}t \\
& +\int_0^T\int_{\partial\Omega}\left[-\frac{\partial q_1}{\partial n}-q_1\bar{w}\cdot\vec{n}+q_1\frac{\partial V_{\text{ext}}}{\partial n}-q_2\right]\rho\ \mathrm{d}s\mathrm{d}t.
\end{aligned}\tag{3.2}$$

Noting first that (3.2) must hold for all $\rho\in C_0^\infty(\Omega\times(0,T))$ (i.e., where $\rho(\vec{x},T)$, $\rho(\vec{x},0)$, ρ vanish on Ω, and $\frac{\partial\rho}{\partial n}$ vanishes on $\partial\Omega$), and observing that $C_0^\infty(\Omega\times(0,T))$ is dense on $L^2(\Omega\times(0,T))$, we obtain the adjoint PDE:

$$-\partial_t q_1-\nabla^2 q_1-\vec{w}\cdot\nabla q_1+\nabla V_{\text{ext}}\cdot\nabla q_1=\rho-\widehat{\rho}\quad\text{on }\Omega\times(0,T).$$

Removing the restriction that $\rho(\vec{x},T)$ vanishes on Ω, and arguing similarly, leads to the adjoint boundary condition $q_1(\vec{x},T)=0$. From here, we may similarly remove the condition that $\frac{\partial\rho}{\partial n}$ vanishes on $\partial\Omega$ to conclude that $q_1=0$ on $\partial\Omega\times(0,T)$. Setting the final integral term in (3.2) to zero then gives the relation between q_1 and q_2. Putting all the pieces together, and relabelling the q_1 as the *adjoint variable* q, we obtain the complete adjoint problem:

$$\begin{aligned}
-\partial_t q-\nabla^2 q-\vec{w}\cdot\nabla q+\nabla V_{\text{ext}}\cdot\nabla q=\rho-\widehat{\rho}\quad & \text{on }\Omega\times(0,T), \\
q=0\qquad & \text{at } t=T, \\
q=0\qquad & \text{on }\partial\Omega\times(0,T).
\end{aligned}\tag{3.3}$$

Searching for the stationary point upon differentiation with respect to each component of $\vec{w}$, using similar working as above, gives:

$$D_{w_i}\mathcal{L}(\bar{\rho},\bar{w},q_1,q_2)w_i=\beta\int_0^T\int_\Omega\bar{w}_i w_i\ \mathrm{d}x\mathrm{d}t-\int_0^T\int_\Omega\frac{\partial}{\partial x_i}(\bar{\rho}w_i)q_1\ \mathrm{d}x\mathrm{d}t.$$

Therefore, using integration by parts,

$$0=\beta\int_0^T\int_\Omega\bar{w}_i w_i\ \mathrm{d}x\mathrm{d}t+\int_0^T\int_\Omega\bar{\rho}\frac{\partial q_1}{\partial x_i}w_i\ \mathrm{d}x\mathrm{d}t-\int_0^T\int_\Omega\frac{\partial}{\partial x_i}(\bar{\rho}q_1 w_i)\ \mathrm{d}x\mathrm{d}t,$$

whereupon considering the derivatives with respect to the all entries of $\vec{w}$, and applying Green's formula, leads to the *gradient equation*:

$$\beta\vec{w}+\rho\nabla q=\vec{0}.\tag{3.4}$$

To summarize, the complete first-order optimality system for the problem (2.2) with the Dirichlet boundary condition $\rho=0$ includes the PDE constraint itself (often

referred to as the *state equation*), the adjoint problem (3.3), and the gradient equation (3.4).

– ***Nonlinear control with 'no-flux type' boundary condition:*** To provide an illustration of how the same working may be applied to problem (2.2) with the no-flux boundary condition (2.5), and $\gamma = 0$, we briefly consider the Lagrangian given by:

$$\begin{aligned}\mathcal{L}(\rho, \vec{w}, q_1, q_2) = \frac{1}{2}\int_0^T \int_\Omega (\rho - \widehat{\rho})^2 \, \mathrm{d}x\mathrm{d}t + \frac{\beta}{2}\int_0^T \int_\Omega \|\vec{w}\|^2 \, \mathrm{d}x\mathrm{d}t \\ - \int_0^T \int_\Omega \left(\partial_t \rho - \nabla^2 \rho + \nabla \cdot (\rho \vec{w}) - \nabla \cdot (\rho \nabla V_{\text{ext}}) - f\right) q_1 \, \mathrm{d}x\mathrm{d}t \\ - \int_0^T \int_{\partial\Omega} \left(\frac{\partial \rho}{\partial n} - \rho \vec{w} \cdot \vec{n} + \rho \frac{\partial V_{\text{ext}}}{\partial n}\right) q_2 \, \mathrm{d}s\mathrm{d}t.\end{aligned}$$

Solving $D_\rho \mathcal{L}(\bar{\rho}, \bar{w}, q_1, q_2)\rho = 0$ for all ρ such that $\rho(\vec{x}, 0) = 0$ gives that:

$$\begin{aligned}0 = & -\int_0^T \int_\Omega (-\partial_t q_1 - \nabla^2 q_1 - \bar{w} \cdot \nabla q_1 + \nabla V_{\text{ext}} \cdot \nabla q_1 + \widehat{\rho} - \bar{\rho})\rho \, \mathrm{d}x\mathrm{d}t \\ & + \int_\Omega q(\vec{x}, T)\rho(\vec{x}, T) \, \mathrm{d}x + \int_0^T \int_{\partial\Omega} (q_1 - q_2)\frac{\partial \rho}{\partial n} \, \mathrm{d}s\mathrm{d}t \\ & - \int_0^T \int_{\partial\Omega} \left[\frac{\partial q_1}{\partial n} + (q_1 - q_2)\left(\bar{w} \cdot \vec{n} - \frac{\partial V_{\text{ext}}}{\partial n}\right)\right] \rho \, \mathrm{d}s\mathrm{d}t.\end{aligned}$$

Applying the same reasoning as above then leads to the adjoint problem:

$$\begin{aligned} -\partial_t q - \nabla^2 q - \vec{w} \cdot \nabla q + \nabla V_{\text{ext}} \cdot \nabla q &= \rho - \widehat{\rho} \quad \text{on } \Omega \times (0, T), \\ q &= 0 \qquad \text{at } t = T, \\ \frac{\partial q}{\partial n} &= 0 \qquad \text{on } \partial\Omega \times (0, T), \end{aligned} \tag{3.5}$$

along with the state equation as in (2.2), and the gradient equation (3.4).

– ***Linear control with Dirichlet boundary condition:*** We next consider the problem (2.6) with the Dirichlet boundary condition (2.4), and $\gamma = 0$. This leads to the continuous Lagrangian:

$$\begin{aligned}\mathcal{L}(\rho, w, q_1, q_2) = \frac{1}{2}\int_0^T \int_\Omega (\rho - \widehat{\rho})^2 \, \mathrm{d}x\mathrm{d}t + \frac{\beta}{2}\int_0^T \int_\Omega w^2 \, \mathrm{d}x\mathrm{d}t \\ - \int_0^T \int_\Omega \left(\partial_t \rho - \nabla^2 \rho - \nabla \cdot (\rho \nabla V_{\text{ext}}) - w - f\right) q_1 \, \mathrm{d}x\mathrm{d}t \\ - \int_0^T \int_{\partial\Omega} \rho q_2 \, \mathrm{d}s\mathrm{d}t.\end{aligned} \tag{3.6}$$

Solving $D_\rho \mathcal{L}(\bar{\rho}, \bar{w}, q_1, q_2)\rho = 0$ for all ρ such that $\rho(\vec{x}, 0) = 0$ gives that:

$$\begin{aligned} = & -\int_0^T \int_\Omega (-\partial_t q_1 - \nabla^2 q_1 + \nabla V_{\text{ext}} \cdot \nabla q_1 + \widehat{\rho} - \bar{\rho})\rho \, \mathrm{d}x\mathrm{d}t \\ & + \int_\Omega q(\vec{x}, T)\rho(\vec{x}, T) \, \mathrm{d}x + \int_0^T \int_{\partial\Omega} q_1 \frac{\partial \rho}{\partial n} \, \mathrm{d}s\mathrm{d}t \\ & + \int_0^T \int_{\partial\Omega} \left[-\frac{\partial q_1}{\partial n} + q_1 \frac{\partial V_{\text{ext}}}{\partial n} - q_2 \right] \rho \, \mathrm{d}s\mathrm{d}t. \end{aligned} \tag{3.7}$$

Applying the same reasoning as above then leads to the adjoint problem:

$$\begin{aligned} -\partial_t q - \nabla^2 q + \nabla V_{\text{ext}} \cdot \nabla q = \rho - \widehat{\rho} \quad & \text{on } \Omega \times (0, T), \\ q = 0 \quad & \text{at } t = T, \\ q = 0 \quad & \text{on } \partial\Omega \times (0, T). \end{aligned} \tag{3.8}$$

Searching for the stationary point upon differentiation with respect to w, using similar working as above, gives:

$$D_w \mathcal{L}(\bar{\rho}, \bar{w}, q_1, q_2)w = \beta \int_0^T \int_\Omega \bar{w} w \, \mathrm{d}x\mathrm{d}t + \int_0^T \int_\Omega \bar{w} q_1 \, \mathrm{d}x\mathrm{d}t,$$

leading to the *gradient equation*:

$$\beta w + q = 0. \tag{3.9}$$

To summarize, the complete first-order optimality system for the problem (2.6) with the Dirichlet boundary condition $\rho = c$, where $c \in \mathbb{R}$, includes the PDE constraint itself (often referred to as the *state equation*), the adjoint problem (3.8), and the gradient equation (3.9).

– ***Linear control with 'no-flux type' boundary condition:*** To provide an illustration of how the same working may be applied to problem (2.6) with the no-flux boundary condition (2.8), and $\gamma = 0$, we briefly consider the Lagrangian given by:

$$\begin{aligned} \mathcal{L}(\rho, \vec{w}, q_1, q_2) = & \frac{1}{2} \int_0^T \int_\Omega (\rho - \widehat{\rho})^2 \, \mathrm{d}x\mathrm{d}t + \frac{\beta}{2} \int_0^T \int_\Omega w^2 \, \mathrm{d}x\mathrm{d}t \\ & - \int_0^T \int_\Omega \left(\partial_t \rho - \nabla^2 \rho - \nabla \cdot (\rho \nabla V_{\text{ext}}) - w - f \right) q_1 \, \mathrm{d}x\mathrm{d}t \\ & - \int_0^T \int_{\partial\Omega} \left(\frac{\partial \rho}{\partial n} + \rho \frac{\partial V_{\text{ext}}}{\partial n} \right) q_2 \, \mathrm{d}s\mathrm{d}t. \end{aligned}$$

Applying the same reasoning as above then leads to the adjoint problem:

$$\begin{aligned} -\partial_t q - \nabla^2 q + \nabla V_{\text{ext}} \cdot \nabla q = \rho - \widehat{\rho} \quad & \text{on } \Omega \times (0, T), \\ q = 0 \quad & \text{at } t = T, \\ \frac{\partial q}{\partial n} = 0 \quad & \text{on } \partial\Omega \times (0, T), \end{aligned}$$

along with the state equation as in (2.6), and the gradient equation (3.9).

– ***Additional nonlocal integral term:*** Finally, applying this methodology to the problem (2.2), with $\gamma \neq 0$, results in obtaining the state equation:

$$\begin{aligned} \partial_t \rho - \nabla^2 \rho + \nabla \cdot (\rho \vec{w}) - \gamma \nabla_r \cdot \left(\int_\Omega \rho(r)\rho(r')\vec{K}(|r-r'|)\, \mathrm{d}r' \right) & \\ = \nabla \cdot (\rho \nabla V_{\text{ext}}) \quad \text{on } \Omega \times (0,T), & \\ \rho = \rho_0(\vec{x}) \qquad\qquad \text{at } t = 0, & \end{aligned}$$

with boundary condition (2.4) or (2.5). The adjoint equation is:

$$\begin{aligned} -\partial_t q - \nabla^2 q - \vec{w} \cdot \nabla q + \nabla V_{\text{ext}} \cdot \nabla q + \left(\gamma \int_\Omega \rho(r')\vec{K}(|r-r'|)\, \mathrm{d}r' \right) \cdot \nabla_r q(r) & \\ + \gamma \int_\Omega \left(\rho(r')\vec{K}(|r-r'|) \cdot \nabla_{r'} q(r') \right) \mathrm{d}r' = \rho - \widehat{\rho} \quad \text{on } \Omega \times (0,T), & \\ q = 0 \qquad\qquad \text{at } t = T. & \end{aligned}$$

The boundary condition for the adjoint equation, corresponding to (2.4) in the state equation is:

$$q = 0 \qquad \text{on } \partial\Omega \times (0,T),$$

regardless of the value of c in (2.4). The boundary condition for the adjoint corresponding to boundary condition (2.5) is:

$$\frac{\partial q}{\partial n} = 0 \qquad \text{on } \partial\Omega \times (0,T).$$

Note that the boundary conditions for the adjoint equations remain unchanged when adding an interaction term, compare to (3.3) and (3.5). This applies to the linear case as well and is therefore ommited here. Finally, the gradient equation is:

$$\beta \vec{w} + \rho \nabla q = \vec{0}.$$

[[check against Dante's papers]]

4. Numerical method for optimization model.

4.1. Pseudospectral method for the forward problem. As described in Section 2.2, we solve the forward problem using Chebyshev pseudospectral methods, in particular implemented in Matlab using 2DChebClass [31, 21] The principal novelties of the method concern the computation of convolution integrals and the implementation of spatial boundary conditions; the boundary conditions in time will be discussed in the following section.

As described in [31], the convolution integrals are computed in real space, in contrast to many implementations in which they are computed via Fourier transforms. In this latter case, the domain must be padded to ensure that all functions are (approximately) periodic, with suitably large padding regions to ensure that the edges of the physical domain do not interact with each other. The principal advantage of this method is that it is computationally cheap, with requiring only fast Fourier transforms and multiplication of functions. The disadvantages are that, as above, one needs to pad the domain (which effectively wastes computational power in computing a lot of zeros), and the boundary conditions can be challenging to impose on such

an extended grid. The spatial method can be implemented by a single matrix–vector multiplication, with the matrix precomputed for all time steps and is implemented on the physical domain, allowing efficient implementation of the boundary conditions. In short, for a convolution of functions f and ρ,

$$I_f(\rho) = \int f(\vec{x} - \vec{y})\rho(\vec{y}) \, \mathrm{d}\vec{y}$$

we compute a matrix M_f, such that for the discretized vector $\boldsymbol{\rho}$, we may approximate the convolution integral via the matrix multiplication $M_f\boldsymbol{\rho} \approx I_f(\rho)$. Note that the matrix M_f is independent of ρ, and as such may be precomputed. In fact, it may be computed in parallel for each computational grid point.

As is standard, after discretization, in this case through the use of (mapped) Chebyshev pseudospectral points, the forward PDE(s) are converted into a system of ODEs. For example, the diffusion equation becomes

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\rho} = D_2\boldsymbol{\rho}, \qquad + \text{ IC and BC}, \tag{4.1}$$

where $\boldsymbol{\rho}$ is a vector of values of the solution at each of the Chebyshev points, and D_2 is the Chebyshev second order differentiation matrix. In the interior of the domain, this can be solved using essentially any ODE solver; the challenge lies in imposing the correct spatial boundary conditions. One standard approach is to modify the matrix on the right hand side of (4.1) so that the boundary conditions are automatically satisfied. This is known as 'boundary-boardering' [11]. For simple boundary conditions, such as homogenenous Dirichlet or Neumann, such an approach is relatively straightforward. For example, for homogeneous Dirichlet, assuming that the initial conditions satisfy the boundary conditions, it is sufficient to set the first and last rows and columns of $D2$ to zero. For homogeneous Neumann, there is a similar approach (see [41]), which becomes more involved with more complex right hand sides of the PDE. Another approach is to restrict the computation to interpolants (solutions) which satisfy the boundary conditions; we do not discuss this here as it is highly non-trivial for the non-linear, non-local problems that we are interested in.

Here we take a more general approach. The imposition of spatial boundary conditions can be seen as extending the discretized system of ODEs to a system of differential-algebraic equations, where the discretized PDE is solved on the interior of the domain, and the boundary conditions correspond to algebraic equations. There are various numerical methods to solve such differential-algebraic equations, e.g. [37]. The main advantage here is that the numerical method does not have to be explicitly adapted when one changes the boundary conditions; one simply has to specify different algebraic constraints that correspond to the boundary conditions. In fact, the 2DChebClass code automatically identifies the boundary of various geometries, allowing a simple implementation of this approach.

4.2. Pseudospectral method for the adjoint equation. For the optimization problem, we have a pair of coupled PDEs, the forward PDE with an initial time condition and the adjoint equation with a final time condition. Hence, one cannot use a standard time-stepping scheme, due to the negative Laplacian having the same sign as the time derivative in the state equation, and the opposite sign in the adjoint equation.(+++ Note that above it is written differently: both terms on LHS, therefore opposite true, may be confusing+++) This must be accounted for to ensure stable

solutions. One approach is to apply a backward Euler method for the time derivative in the state equation, with the adjoint operator applied to the adjoint equation, whereupon a huge-scale coupled system of equations is obtained from matrices arising at each time-step. These may be tackled using a preconditioned iterative method, following the methodology in [32, 33, 38], for instance. As above, as well as boundary conditions in time, there are also boundary conditions in space. In order to utilise our efficient and accurate forward solver, we reverse time in the adjoint problem, resulting in a set of equations with initial conditions, so that the pseudospectral method can be applied, as described in Section 4.1. However, the forward and adjoint equations are coupled non-locally in time; the adjoint equation requires the value of the state variable at later times, so the two equations cannot be solved simultaneously. This difficulty is addressed using the fixed point algorithm presented in the next section.

4.3. Optimization Solver. The remaining challenges are to (i) determine the time discretization for the optimality system; (ii) choose a suitable optimization scheme. For (i), we again choose a Chebyshev pseudospectral scheme (1D in time), which, assuming that the solutions are smooth in time, leads to exponentially accurate interpolation. In [[REF]], we compare this approach to a standard finite difference scheme. As mentioned in Section 4.1, the system of ODEs and the algebraic-differential equations, resulting from applying the pseudospectral method, can be solved with a standard ODE solver. In this paper, the Matlab inbuilt ODE solver `ode15s` is used. For (ii), we note that the choice of optimization solver depends strongly on the nature of the solution, and the amount of information available. For ease of implementation, and to remove the need to, e.g. analytically compute the Jacobian, we use a fixed point method. However, we note that this approach is highly modular and it is straightforward to replace this solver with any other optimization routine. In order to solve an optimality system of the form derived in Section 3, we adapt a fixed point method, also known as sweeping algorithm in the literature, for example in [3], to solve the system of equations iteratively.

During each iteration, denoted by i, the following steps are computed:
1. The solution to the forward equation is computed, using a guess for the control $\vec{w}_i$ as input variable, alongside the initial condition $\rho(\vec{x}, 0)$.
2. The adjoint equation is solved, using $\vec{w}_i$, and the solution $\boldsymbol{\rho}_i$ from the forward solve, as inputs. This circumvents the issue resulting from the non-local coupling in time, mentioned in Section 4.2, since $\boldsymbol{\rho}_i$ contains the solution for all times $t \in [0, T]$. The final time condition $q(\vec{x}, T)$ must be provided and time is reversed in the adjoint equation, as explained in Section 4.2. The result $\boldsymbol{q}_{i,T-t}$ is transformed back to $\boldsymbol{q}_{i,t}$, to proceed.
3. The gradient equation is solved, given the solutions $\boldsymbol{\rho}_i$ and $\boldsymbol{q}_i$. This results in a new control value $\vec{w}_g$.
4. Finally, the update $\vec{w}_{i+1}$ is a linear combination of the current guess $\vec{w}_i$, and $\vec{w}_g$, the value obtained from solving the gradient equation:

$$\vec{w}_{i+1} = (1 - \lambda)\vec{w}_i + \lambda\vec{w}_g.$$

This updating method is called a mixing scheme, with mixing rate λ, and is known to stabilise fixed point methods, ++ add references ++. The guess for the control is updated from $\vec{w}_i$ to $\vec{w}_{i+1}$ and the steps described above are repeated until the method converges. +++ Notation of $\boldsymbol{\rho}_i$ and $\boldsymbol{q}_i$ needs to be decided and made consistent. bold for indicating that it's discretized as in Section 4.1. +++

Typical values of λ, which provide stable convergence, lie between 0.1 and 0.001. Throughout this paper, $\lambda = 0.01$, unless stated otherwise. This mixing scheme is equivalent to the updating scheme presented in [14]. Note that, while the solutions $\boldsymbol{\rho}_i$ and $\boldsymbol{q}_i$ change in each iteration, the initial condition $\rho(\vec{x}, 0)$ and final time condition $q(\vec{x}, T)$ remain unchanged throughout the process. Therefore, the only variable inducing a change in the solution is $\vec{w}_i$.

The convergence of the scheme is measured by computing the error between $\vec{w}_i$ and $\vec{w}_g$. If this error is lower than a set tolerance, the optimality system is self-consistent. This implies that the solution triplet $(\bar{\rho}, \bar{w}, q)$ (++ check notation here - consistent with Section 3? ++) solves the optimality system, and is therefore an optimal solution to the PDE-constrained optimization problem of interest. All errors in Section 4.4 and Section 5 are calculated as follows, where y is the variable of interest and y_R is the reference value that y is compared to. For example, when measuring convergence of the fixed point scheme, $y = \vec{w}_g$ and $y_R = \vec{w}_i$. Alternatively, when investigating a known test problem, as considered in Section 4.4, y is a numerical solution and y_R is an exact solution.
The relative L^2 error in the spatial direction is:

$$\mathcal{E}_{Rel}(t) = \frac{||y(x,t) - y_R(x,t)||_{L^2(\Omega)}}{||y_R(x,t) + 10^{-10}||_{L^2(\Omega)}},$$

where the small additional term on the denominator prevents division by zero. Furthermore, the absolute L^2 error is:

$$\mathcal{E}_{Abs}(t) = ||y(x,t) - y_R(x,t)||_{L^2(\Omega)},$$

Then an L_∞ error in time is taken of the minimum of $\mathcal{E}_{Rel}$ and $\mathcal{E}_{Abs}$:

$$\mathcal{E} = \max_{t\in[0,T]} \left[\min\left(\mathcal{E}_{Rel}(t), \mathcal{E}_{Abs}(t)\right)\right].$$

The minimum between absolute and relative spatial error is taken to avoid choosing an erogenously large relative error, caused by division of one small term by another. As a benchmark, we compared the fixed point scheme to Matlab's inbuilt `fsolve` function. It uses the trust-region-dogleg algorithm, see [34], to solve the optimality systems of interest. While it is very robust, it is also much slower in comparison to the fixed point method, which is working reliably for the types of problems considered above.

(++ include shortcomings of the method? (e.g. advection dominated problems, good initial guess for BVP solvers) ++)

4.4. Validation. Specific test problems are given in Appendix A. Further tests to validate the method are presented in Appendix B.

5. Numerical experiments. In order to solve the optimal control problem (2.2), or (2.6), some inputs must be provided. The desired state $\widehat{\rho}$, the PDE source term f, and the external potential V_{ext} must be given. Furthermore, an initial condition for ρ, the final time condition for q and an initial guess for the control $\vec{w}$ have to be be specified. The interaction kernel (++ terminology? ++) is of the form:

$$\vec{K} = \nabla V_2, \qquad V_2 = e^{-x^2}.$$

Three interaction strengths are considered. Firstly, the problem is solved without an interaction term present ($\gamma = 0$). Then the considered problem is solved with an order one attractive interaction term ($\gamma = -1$) and an order one repulsive interaction term ($\gamma = 1$), respectively. Initially, the control $\vec{w}$ is set to zero. It is then investigated how the control changes from this baseline, influenced by the different interaction strengths. This is considered for different values of the regularization parameter β and it is expected that the control will increase with decreasing β, since the cost functional in problems (2.2) and (2.6) allows for a larger control with smaller β. In the following examples, the domain considered is $\Omega \times [0, T] = [-1, 1] \times [0, 1]$. The number of spatial points is $N = 40$, and the number of time points is $n = 41$ (++will be changed++). The tolerances in the ODE solver are set to 10^{-8} and the tolerance for the convergence of the optimization algorithm is 10^{-4}. The mixing parameter λ is 0.01, unless stated otherwise.

5.1. Nonlinear control problems with an additional nonlocal integral term. Examples of solving Problem (2.2), with 'no-flux type' boundary conditions (2.5) and Dirichlet boundary conditions (2.4) are given in this section.

5.1.1. Neumann boundary conditions, Example 1. The chosen inputs for this example are:

$$\begin{aligned} \widehat{\rho} &= \frac{1}{2}(1-t) + t\left(\frac{1}{2}\sin(\pi(y-2)/2) + \frac{1}{2}\right) \\ \rho_0 &= \frac{1}{2} \\ q_T &= 0 \\ \vec{w} &= 0 \\ f &= 0 \\ V_{ext} &= 0 \end{aligned}$$

Table 5.1 displays the results for this example. The value of the cost functional for the uncontrolled case (J_{uc}) is compared with the controlled case (J_c) for different values of β and for each of the interaction strengths. It can be observed that in all cases J_c will be lower or equal value to J_{uc}. The lowest values of J_c will be observed for the smallest β value considered. At large values of β, applying control is heavily penalised and the optimal control approaches zero, which coincides with the uncontrolled case. Furthermore, Table 5.1 displays the number of iterations for each of these examples. The desired state $\widehat{\rho}$, and ρ_{uc} for $\gamma = 1$ and $\gamma = -1$, with $\beta = 10^{-3}$ are shown in Figure 5.1. The desired state $\hat{\rho}$ and ρ_{uc} are independent of β. However, ρ_{uc} changes considerably with the choice of interaction strength γ. The optimal states ρ for $\gamma = 1, 0, -1$ and corresponding optimal controls, with $\beta = 10^{-3}$ are shown in Figure 5.2. It can be observed that in the case of $\beta = 10^{-3}$, the optimal state ρ is very similar to $\hat{\rho}$, regardless of the choice of interaction. For larger values of β, the optimal ρ displays qualitative differences, depending on the interaction term.

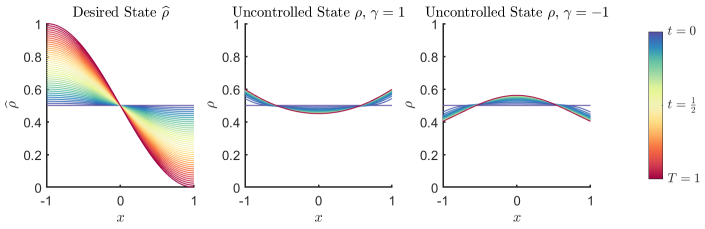

FIG. 5.1. *Example 1, desired state* $\widehat{\rho}$ *and* ρ_{uc} *at* $\gamma = 1$ *and* $\gamma = -1$, $\beta = 10^{-3}$

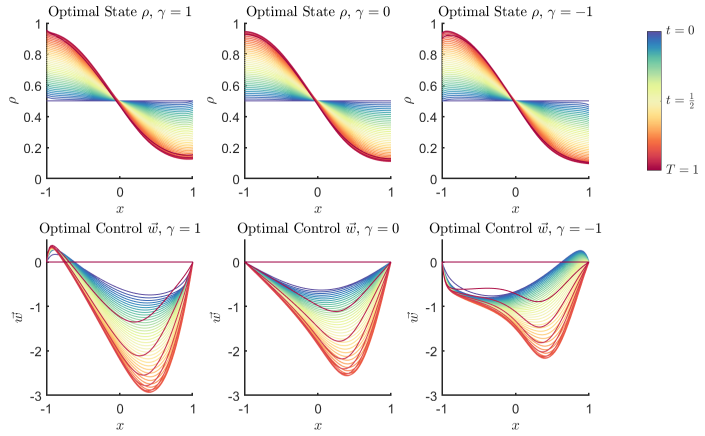

FIG. 5.2. *Example 1, optimal solutions* ρ *for* $\gamma = 1, 0, -1$ *and corresponding optimal controls* $\vec{w}$, $\beta = 10^{-3}$.

5.1.2. Neumann boundary conditions, Example 2. The chosen inputs for Example 2 are:

$$\begin{aligned}
\widehat{\rho} &= \left(\frac{1}{2}\cos(\pi y) + \frac{1}{2}\right)(1-t) + t\left(-\frac{1}{2}\cos(2\pi y) + \frac{1}{2}\right) \\
\rho_0 &= \frac{1}{2}\cos(\pi y) + \frac{1}{2} \\
q_T &= 0 \\
\vec{w} &= 0 \\
f &= 0 \\
V_{ext} &= 0
\end{aligned}$$

In Table 5.2 the results for Example 2 are displayed. These are mostly comparable with the results for Example 1. In all three configurations of the interaction term, the control is focussed on transporting the mass from the middle onto two piles centred at $x = -0.5$ and $x = 0.5$. In Figure 5.3, the desired state $\widehat{\rho}$, the optimal state ρ and the optimal control $\vec{w}$ are displayed for $\beta = 10^{-3}$, and compared to Example 3 below.

β / γ	10^{-3}	10^{-1}	10	10^{3}
-1	$J_{uc} = 0.0438$ $J_c = 0.0011$ Iter. $= 667$	$J_{uc} = 0.0438$ $J_c = 0.0270$ Iter. $= 649$	$J_{uc} = 0.0438$ $J_c = 0.0435$ Iter. $= 468$	$J_{uc} = 0.0438$ $J_c = 0.0438$ Iter. $= 13$
0	$J_{uc} = 0.0417$ $J_c = 0.0014$ Iter. $= 671$	$J_{uc} = 0.0417$ $J_c = 0.0283$ Iter. $= 656$	$J_{uc} = 0.0417$ $J_c = 0.0415$ Iter. $= 434$	$J_{uc} = 0.0417$ $J_c = 0.0417$ Iter. $= 1$
1	$J_{uc} = 0.0434$ $J_c = 0.0020$ Iter. $= 674$	$J_{uc} = 0.0434$ $J_c = 0.0324$ Iter. $= 686$	$J_{uc} = 0.0434$ $J_c = 0.0432$ Iter. $= 411$	$J_{uc} = 0.0434$ $J_c = 0.0434$ Iter. $= 1$

TABLE 5.1

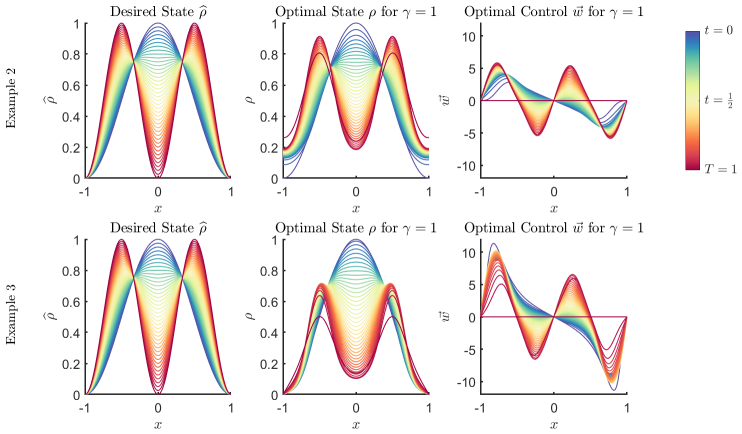

FIG. 5.3. *Example 2/ Example 3, desired state $\widehat{\rho}$, optimal state ρ and corresponding optimal control $\vec{w}$, $\beta = 10^{-3}$, $\gamma = 1$.*

5.1.3. Dirichlet boundary conditions, Example 3. The inputs for this example are:

$$\begin{aligned}
\widehat{\rho} &= \left(\frac{1}{2}\cos(\pi y) + \frac{1}{2}\right)(1-t) + t\left(-\frac{1}{2}\cos(2\pi y) + \frac{1}{2}\right)\\
\rho_0 &= \frac{1}{2}\cos(\pi y) + \frac{1}{2}\\
q_T &= 0\\
\vec{w} &= 0\\
f &= 0\\
V_{ext} &= 0
\end{aligned}$$

Table 5.3 presents the results for this example for a range of β values and different interaction strengths. The observations are in line with those in Example 1 and 2. In particular, $\widehat{\rho}$ and ρ_0 coincide with those of the Neumann problem in Example 2. The comparison between results can be seen in Figure 5.3. Both the optimal state ρ and the optimal control are qualitatively different when considering Dirichlet boundary conditions over Neumann conditions.

β / γ	10^{-3}	10^{-1}	10	10^{3}
-1	$J_{uc} = 0.0536$ $J_c = 0.0096$ Iter. $= 724$	$J_{uc} = 0.0536$ $J_c = 0.0493$ Iter. $= 769$	$J_{uc} = 0.0536$ $J_c = 0.0535$ Iter. $= 379$	$J_{uc} = 0.0536$ $J_c = 0.0536$ Iter. $= 1$
0	$J_{uc} = 0.0669$ $J_c = 0.0109$ Iter. $= 726$	$J_{uc} = 0.0669$ $J_c = 0.0603$ Iter. $= 770$	$J_{uc} = 0.0669$ $J_c = 0.0668$ Iter. $= 390$	$J_{uc} = 0.0669$ $J_c = 0.0669$ Iter. $= 1$
1	$J_{uc} = 0.0839$ $J_c = 0.0125$ Iter. $= 728$	$J_{uc} = 0.0839$ $J_c = 0.0749$ Iter. $= 772$	$J_{uc} = 0.0839$ $J_c = 0.0838$ Iter. $= 396$	$J_{uc} = 0.0839$ $J_c = 0.0839$ Iter. $= 1$

TABLE 5.2

β / γ	10^{-3}	10^{-1}	10	10^{3}
-1	$J_{uc} = 0.0510$ $J_c = 0.0026$ Iter. $= 690$	$J_{uc} = 0.0510$ $J_c = 0.0365$ Iter. $= 696$	$J_{uc} = 0.0510$ $J_c = 0.0508$ Iter. $= 696$	$J_{uc} = 0.0510$ $J_c = 0.0510$ Iter. $= 696$
0	$J_{uc} = 0.0417$ $J_c = 0.0027$ Iter. $= 696$	$J_{uc} = 0.0417$ $J_c = 0.0343$ Iter. $= 742$	$J_{uc} = 0.0417$ $J_c = 0.0416$ Iter. $= 409$	$J_{uc} = 0.0417$ $J_c = 0.0417$ Iter. $= 1$
1	$J_{uc} = 0.0452$ $J_c = 0.0030$ Iter. $= 703$	$J_{uc} = 0.0452$ $J_c = 0.0388$ Iter. $= 779$	$J_{uc} = 0.0452$ $J_c = 0.0452$ Iter. $= 397$	$J_{uc} = 0.0452$ $J_c = 0.0452$ Iter. $= 1$

TABLE 5.3
Update table! Is for different example

+++ Two additional nonlinear control problems below, may be deleted? +++

5.1.4. Neumann boundary conditions, Symmetric Example 1. Consider the following symmetric setup:

$$\begin{aligned} \widehat{\rho} &= \frac{1}{2}(1-t) + t\frac{1}{4}(\cos(\pi y) + 2) \\ \rho_0 &= \frac{1}{2} \\ q_T &= 0 \\ \vec{w} &= 0 \\ f &= 0 \\ V_{ext} &= 0 \end{aligned}$$

Table 5.4 summarizes the results for this example.The attractive interaction term causes ρ to move towards the centre of the domain. Since $\widehat{\rho}$ is also centred in the domain, J_{uc} is small for $\gamma = -1$ in comparison to the problems with $\gamma = 0$ and $\gamma = 1$. This example illustrates that the particle interaction term can have a significant impact on the optimization problem considered.

β / γ	10^{-3}	10^{-1}	10	10^{3}
-1	$J_{uc} = 0.0041$ $J_c = 0.0002$ Iter. $= 607$	$J_{uc} = 0.0041$ $J_c = 0.0033$ Iter. $= 637$	$J_{uc} = 0.0041$ $J_c = 0.0040$ Iter. $= 311$	$J_{uc} = 0.0041$ $J_c = 0.0041$ Iter. $= 1$
0	$J_{uc} = 0.0104$ $J_c = 0.0005$ Iter. $= 635$	$J_{uc} = 0.0104$ $J_c = 0.0086$ Iter. $= 671$	$J_{uc} = 0.0104$ $J_c = 0.0104$ Iter. $= 340$	$J_{uc} = 0.0104$ $J_c = 0.0104$ Iter. $= 1$
1	$J_{uc} = 0.0195$ $J_c = 0.0011$ Iter. $= 656$	$J_{uc} = 0.0195$ $J_c = 0.0164$ Iter. $= 696$	$J_{uc} = 0.0195$ $J_c = 0.0195$ Iter. $= 356$	$J_{uc} = 0.0195$ $J_c = 0.0195$ Iter. $= 1$

TABLE 5.4

β / γ	10^{-3}	10^{-1}	10	10^{3}
-1	$J_{uc} = 0.0209$ $J_c = 0.0009$ Iter. $= 646$	$J_{uc} = 0.0209$ $J_c = 0.0168$ Iter. $= 691$	$J_{uc} = 0.0209$ $J_c = 0.0209$ Iter. $= 379$	$J_{uc} = 0.0209$ $J_c = 0.0209$ Iter. $= 1$
0	$J_{uc} = 0.0104$ $J_c = 0.0005$ Iter. $= 635$	$J_{uc} = 0.0104$ $J_c = 0.0086$ Iter. $= 671$	$J_{uc} = 0.0104$ $J_c = 0.0104$ Iter. $= 340$	$J_{uc} = 0.0104$ $J_c = 0.0104$ Iter. $= 1$
1	$J_{uc} = 0.0047$ $J_c = 0.00034$ Iter. $= 623$	$J_{uc} = 0.0047$ $J_c = 0.0040$ Iter. $= 651$	$J_{uc} = 0.0047$ $J_c = 0.0047$ Iter. $= 297$	$J_{uc} = 0.0047$ $J_c = 0.0047$ Iter. $= 1$

TABLE 5.5

5.1.5. Neumann boundary conditions, Symmetric Example 2. Consider the following symmetric setup, which is the opposite of the first symmetric example:

$$\begin{aligned} \widehat{\rho} &= \frac{1}{2}(1-t) + t\frac{1}{4}(-\cos(\pi y) + 2) \\ \rho_0 &= \frac{1}{2} \\ q_T &= 0 \\ \vec{w} &= 0 \\ f &= 0 \\ V_{ext} &= 0 \end{aligned}$$

This example can be compared to the Symmetric Example 1. Here, the desired state is having ρ clustered at both boundaries, which is similar to the effect of the repulsive interaction term $\gamma = 1$. Therefore, for this choice of interaction term, the value of the cost functional J_{uc} is smaller than the one for $\gamma = 0$ and $\gamma = -1$. This is the opposite to the observation made in the Symmetric Example 1, which is to be expected, given the two choices of desired state.

5.2. Linear control problems with an additional nonlocal integral term. In this section, examples of solving Problem (2.6) with both 'no-flux type' boundary conditions (2.8) and Dirichlet boundary conditions (2.4).

5.2.1. Dirichlet boundary conditions, Example 4. ++ If kept maybe change $\widehat{\rho}$ and ρ_0? Same as Examples 2 and 3 / or compare to 2 and 3? ++ The inputs for this example are:

$$\begin{aligned}\widehat{\rho} &= (1-t)\left(\frac{1}{2}\cos(\pi y)+\frac{1}{2}\right)+t\left(-\frac{1}{2}\cos(\pi y)+\frac{1}{2}\right)\\ \rho_0 &= \frac{1}{2}\cos(\pi y)+\frac{1}{2}\\ q_T &= 0\\ w &= 0\\ f &= 0\\ V_{ext} &= 0\end{aligned}$$

In Table 5.6 the results for Example 4 for a range of parameter values can be found. The results are qualitatively similar to the previous examples, the only difference is that the control is applied linearly in this example.

β / γ	10^{-3}	10^{-1}	10	10^{3}
-1	$J_{uc}=0.1417$ $J_c=0.0203$ Iter. $=787$	$J_{uc}=0.1417$ $J_c=0.0903$ Iter. $=740$	$J_{uc}=0.1417$ $J_c=0.1407$ Iter. $=503$	$J_{uc}=0.1417$ $J_c=0.1417$ Iter. $=49$
0	$J_{uc}=0.1545$ $J_c=0.0200$ Iter. $=791$	$J_{uc}=0.1545$ $J_c=0.1015$ Iter. $=740$	$J_{uc}=0.1545$ $J_c=0.1536$ Iter. $=510$	$J_{uc}=0.1545$ $J_c=0.1545$ Iter. $=56$
1	$J_{uc}=0.1661$ $J_c=0.0204$ Iter. $=795$	$J_{uc}=0.1661$ $J_c=0.1135$ Iter. $=741$	$J_{uc}=0.1661$ $J_c=0.1652$ Iter. $=515$	$J_{uc}=0.1661$ $J_c=0.1661$ Iter. $=61$

TABLE 5.6

5.2.2. Neumann boundary conditions, Example 5. The inputs for this example are:

$$\begin{aligned}\widehat{\rho} &= \frac{1}{2}(1-t)+t\frac{1}{2}(-\cos(\pi y)+1)\\ \rho_0 &= \frac{1}{2}\\ q_T &= 0\\ w &= 0\\ f &= 0\\ V_{ext} &= 0\end{aligned}$$

Table 5.7 shows the results for Example 5. Note that for this example, when $\beta = 10^{-3}$, the mixing parameter λ had to be set to 0.001 (why? explanation needed?). Again, the only qualitative difference to interpreting the results is that the control is applied linearly.

6. Concluding remarks. Many extensions: "final-time case", different cost functionals, applications to opinion dynamics, flocking, swarming, robotics, ...

β / γ	10^{-3}	10^{-1}	10	10^{3}
-1	$J_{uc} = 0.0606$ $J_c = 0.0060$ Iter. $= 7311$	$J_{uc} = 0.0606$ $J_c = 0.0554$ Iter. $= 771$	$J_{uc} = 0.0606$ $J_c = 0.0606$ Iter. $= 389$	$J_{uc} = 0.0606$ $J_c = 0.0606$ Iter. $= 1$
0	$J_{uc} = 0.0417$ $J_c = 0.0045$ Iter. $= 7227$	$J_{uc} = 0.0417$ $J_c = 0.0383$ Iter. $= 759$	$J_{uc} = 0.0417$ $J_c = 0.0416$ Iter. $= 364$	$J_{uc} = 0.0417$ $J_c = 0.0417$ Iter. $= 1$
1	$J_{uc} = 0.0286$ $J_c = 0.0036$ Iter. $= 7205$	$J_{uc} = 0.0286$ $J_c = 0.0265$ Iter. $= 746$	$J_{uc} = 0.0286$ $J_c = 0.0285$ Iter. $= 341$	$J_{uc} = 0.0286$ $J_c = 0.0286$ Iter. $= 1$

TABLE 5.7

Acknowledgements. MA and JR are supported by The Maxwell Institute Graduate School in Analysis and its Applications, a Centre for Doctoral Training funded by the UK Engineering and Physical Sciences Research Council (EPSRC grant EP/L016508/01), the Scottish Funding Council, Heriot-Watt University, and The University of Edinburgh. BDG gratefully acknowledges support from the EPSRC grant EP/L025159/1. JWP gratefully acknowledges support from the EPSRC Fellowship EP/M018857/2, the EPSRC grant EP/S027785/1, and a Fellowship from The Alan Turing Institute.

Appendix A. Test problems with analytic solutions. +++ Need to double check the solutions, just in case! +++ In this appendix we present two test problems, posed on a two-dimensional domain $\Omega = (-1, 1)^d$, where d is the dimension of the problem. Computing the errors between the given exact solution and the result of solving the uncontrolled PDE and the solution to the PDE-constrained optimization problem demonstrates the correct implementation of the algorithms at hand. The error (Err_{uc}) between the exact solution for the state variable, ρ_{Ex}, and the uncontrolled state, ρ_{uc}, as well as the error (Err_c) between ρ_{Ex} and the controlled state, ρ_c, are measured in a relative L^2 norm in space and in the L_∞ norm in time, as introduced in Section 4.3.

– ***Test Problem 1:*** The following triplet $(\rho, \vec{w}, q)$ form a solution to the first-order continuous optimality conditions derived for the advection–diffusion control problem (2.2) with the zero Dirichlet boundary condition (2.4), with $d = 2$ and $\gamma = 0$:

$$
\begin{aligned}
\rho &= 2\beta^{1/2} e^t \cos\left(\frac{\pi x_1}{2}\right) \cos\left(\frac{\pi x_2}{2}\right), \\
\vec{w} &= \frac{\pi}{2} e^t (e^T - e^t) \left[\sin(\pi x_1)\cos^2\left(\frac{\pi x_2}{2}\right),\ \cos^2\left(\frac{\pi x_1}{2}\right)\sin(\pi x_2)\right]^T, \\
q &= \beta^{1/2}(e^T - e^t)\cos\left(\frac{\pi x_1}{2}\right)\cos\left(\frac{\pi x_2}{2}\right),
\end{aligned}
$$

β / $N_1 \times N_2$, n	10^{-3}	10^{-1}	10	10^3
20	$Err_{uc} = 2.1694 \times 10^{-7}$ $Err_c = 0$	$Err_{uc} = 1.0138 \times 10^{-8}$ $Err_c = 0$	$Err_{uc} = 5.532 \times 10^{-9}$ $Err_c = 0$	$Err_{uc} = 5.514 \times 10^{-9}$ $Err_c = 0$
30	$Err_{uc} = 1.9745 \times 10^{-7}$ $Err_c = 0$	$Err_{uc} = 9.920 \times 10^{-9}$ $Err_c = 0$	$Err_{uc} = 5.759 \times 10^{-9}$ $Err_c = 0$	$Err_{uc} = 5.683 \times 10^{-9}$ $Err_c = 0$
40	$Err_{uc} = 2.1367 \times 10^{-7}$ $Err_c = 0$	$Err_{uc} = 1.0051 \times 10^{-8}$ $Err_c = 0$	$Err_{uc} = 5.643 \times 10^{-9}$ $Err_c = 0$	$Err_{uc} = 5.611 \times 10^{-9}$ $Err_c = 0$

TABLE A.1
Update this table.

where $\boldsymbol{x} = [x_1, x_2]^T$, and

$$\begin{aligned}
\widehat{\rho} &= \beta^{1/2} e^t \cos\left(\frac{\pi x_1}{2}\right)\cos\left(\frac{\pi x_1}{2}\right) - \beta^{1/2}\frac{\pi^2}{2}(e^T - e^t)\cos\left(\frac{\pi x_1}{2}\right)\cos\left(\frac{\pi x_1}{2}\right) \\
&\quad - \beta^{1/2}\frac{\pi^2}{2} e^t (e^T - e^t)^2 \left(\cos\left(\frac{\pi x_1}{2}\right)\cos\left(\frac{\pi x_2}{2}\right)\right)\left[\sin^2\left(\frac{\pi x_1}{2}\right)\cos^2\left(\frac{\pi x_2}{2}\right) + \cos^2\left(\frac{\pi x_1}{2}\right)\sin^2\left(\frac{\pi x_2}{2}\right)\right], \\
V_{\text{ext}} &= 0, \\
f &= \beta^{1/2}\left(2 + \pi^2\right) e^t \cos\left(\frac{\pi x_1}{2}\right)\cos\left(\frac{\pi x_1}{2}\right) \\
&\quad + 2\beta^{1/2}\pi^2 e^{2t}(e^T - e^t) \\
&\quad \left[\cos^3\left(\frac{\pi x_1}{2}\right)\cos^3\left(\frac{\pi x_2}{2}\right) - \cos^3\left(\frac{\pi x_1}{2}\right)\sin^2\left(\frac{\pi x_2}{2}\right)\cos\left(\frac{\pi x_2}{2}\right) - \cos^3\left(\frac{\pi x_2}{2}\right)\sin^2\left(\frac{\pi x_1}{2}\right)\cos\left(\frac{\pi x_1}{2}\right)\right], \\
\rho_0 &= 2\beta^{1/2}\cos\left(\frac{\pi x_1}{2}\right)\cos\left(\frac{\pi x_2}{2}\right),
\end{aligned}$$

+++ fix equations over size...+++ Table A.1 displays the results for Test Problem 1. It can be observed that the error for the uncontrolled problem (Err_{uc}), and the error for the optimal control problem (Err_c), coincide. This is because the same forward problem is solved in the first step of the fixed point algorithm as in solving the forward problem, when provided with the exact solution. ++ Maybe change these tables to show the errors for q and w as well. ++

– ***Test Problem 2:*** Similarly, the following $(\rho, \vec{w}, q)$ solve the problem (2.2) with the boundary condition (2.5), with $d = 1$, and $\gamma = 0$:

$$\begin{aligned}
\rho &= 2c^{1/2}\beta^{1/2} e^t \left(\cos(\pi x) + 2\right), \\
\vec{w} &= 2\pi c e^t (e^T - e^t)\sin(\pi x)\left(\cos(\pi x) + 2\right), \\
q &= c^{1/2}\beta^{1/2}(e^T - e^t)\left(\cos(\pi x)\right),
\end{aligned}$$

where $c > 0$ is an arbitary constant, and

$$\begin{aligned}
\widehat{\rho} &= -c^{1/2}\beta^{1/2}\pi^2\left(e^T - e^t\right)\cos(\pi x) + c^{1/2}\beta^{1/2} e^t\left(\cos(\pi x) + 2\right) - 2\pi^2 c^{3/2}\beta^{1/2} e^t (e^T - e^t)^2 \sin^2(\pi x)\cos(\pi x), \\
V_{\text{ext}} &= 0, \\
f &= 2(1 + \pi^2)c^{1/2}\beta^{1/2} e^t \cos(\pi x) + 4\pi^2 c^{3/2}\beta^{1/2} e^{2t}(e^T - e^T)\left(\cos(\pi x)\right)\left(\cos(\pi x) + 2\right)\left[\cos(\pi x)\left(\cos(\pi x) + 2\right) - 2\sin^2(\pi x)\right] \\
\rho_0 &= 2c^{1/2}\beta^{1/2}\cos(\pi x),
\end{aligned}$$

+++ Maybe delete the 2? Not done in the current version of my code – so also not in Appendix B +++ Table A.2 shows the results for the error in both ρ_{uc} and ρ_c,

β N_1, N_2, n	10^{-3}	10^{-1}	10	10^3
10	$Err_{uc} = 0.0361$ $Err_c = 1.628 \times 10^{-17}$	$Err_{uc} = 0.1727$ $Err_c = 1.5116 \times 10^{-16}$	$Err_{uc} = 0.1727$ $Err_c = 1.0992 \times 10^{-16}$	$Err_{uc} = 0.1727$ $Err_c = 1.0265 \times 10^{-16}$
20	$Err_{uc} = 2.0020 \times 10^{-7}$ $Err_c = 2.266 \times 10^{-17}$	1.5986×10^{-6} $Err_c = 1.1093 \times 10^{-16}$	$Err_{uc} = 1.6059 \times 10^{-6}$ $Err_c = 1.4317 \times 10^{-16}$	$Err_{uc} = 1.6059 \times 10^{-6}$ $Err_c = 1.1007 \times 10^{-16}$
30	$Err_{uc} = 1.2515 \times 10^{-7}$ $Err_c = 2.086 \times 10^{-17}$	$Err_{uc} = 5.4219 \times 10^{-8}$ $Err_c = 1.0673 \times 10^{-16}$	$Err_{uc} = 2.9363 \times 10^{-8}$ $Err_c = 1.0697 \times 10^{-16}$	$Err_{uc} = 2.9363 \times 10^{-8}$ $Err_c = 1.0779 \times 10^{-16}$
40	$Err_{uc} = 1.2529 \times 10^{-7}$ $Err_c = 1.837 \times 10^{-17}$	$Err_{uc} = 5.4266 \times 10^{-8}$ $Err_c = 1.0200 \times 10^{-16}$	$Err_{uc} = 2.9599 \times 10^{-8}$ $Err_c = 1.0400 \times 10^{-16}$	$Err_{uc} = 2.9599 \times 10^{-8}$ $Err_c = 1.0319 \times 10^{-16}$

TABLE A.2
Update this table!

the Fixed point algorithm gives the same errors as Err_{uc}. It can be concluded that 10 spatial and time points are not sufficient for solving the uncontrolled PDE, and 20 points are still not quite enough. However, 30 points are enough to get results that are close to the ODE tolerance 10^{-8}.

– ***Test Problem 3:*** Similarly, the following $(\rho, \vec{w}, q)$ solve the problem (2.2) with the boundary condition (2.5), with $d = 2$, and $\gamma = 0$:

$$\rho = 2e^t \cos(\pi x_1)\cos(\pi x_2),$$
$$\vec{w} = \frac{2\pi}{\beta} e^t (e^T - e^t) \left[\sin(\pi x_1)\cos(\pi x_1)\cos^2(\pi x_2),\ \cos^2(\pi x_1)\sin(\pi x_2)\cos(\pi x_2)\right]^T,$$
$$q = (e^T - e^t)\cos(\pi x_1)\cos(\pi x_2),$$

where

$$\begin{aligned}\widehat{\rho} = &\left[-2\pi^2 e^T + (1 + 2\pi^2)e^t\right]\cos(\pi x_1)\cos(\pi x_2) \\ &- \frac{2\pi^2}{\beta} e^t (e^T - e^t)^2 \left[\sin^2(\pi x_1)\cos(\pi x_1)\cos^3(\pi x_2) + \cos^3(\pi x_1)\sin^2(\pi x_2)\cos(\pi x_2)\right],\end{aligned}$$
$$V_{\text{ext}} = 0,$$
$$\begin{aligned}f = &(2 + 4\pi^2)e^t \cos(\pi x_1)\cos(\pi x_2) + \frac{8\pi^2}{\beta} e^{2t}(e^T - e^T)\cos^3(\pi x_1)\cos^3(\pi x_2) \\ &- \frac{8\pi^2}{\beta} e^{2t}(e^T - e^T)\sin^2(\pi x_1)\cos(\pi x_1)\cos^3(\pi x_2) \\ &- \frac{8\pi^2}{\beta} e^{2t}(e^T - e^T)\cos^3(\pi x_1)\sin^2(\pi x_2)\cos(\pi x_2),\end{aligned}$$
$$\rho_0 = 2\cos(\pi x_1)\cos(\pi x_2),$$

Table A.3 displays the error in ρ_{uc} and ρ_c for different choices of spatial and time points as well as for different β values. It is shown that the choice of 20 points in space and time are not sufficient to reach the accuracy of the ODE solver of 10^{-8}, while 30 points are sufficient for most problems. However, the error measures of the ODE solver and used in this paper may differ and therefore may not be as straightforward to compare. It can also be noted that the case with $\beta = 10^{-3}$ results in less accuracy than other choices of β. When computing this with $N_1 = N_2 = n = 50$, the error

β / $N_1 \times N_2$, n	10^{-3}	10^{-1}	10	10^3
20	$Err_{uc} = 1.3792 \times 10^{-6}$ $Err_c = 1.3792 \times 10^{-6}$	$Err_{uc} = 1.3419 \times 10^{-6}$ $Err_c = 1.3419 \times 10^{-6}$	$Err_{uc} = 1.3413 \times 10^{-6}$ $Err_c = 1.3413 \times 10^{-6}$	$Err_{uc} = 1.3412 \times 10^{-6}$ $Err_c = 1.3412 \times 10^{-6}$
30	$Err_{uc} = 1.9316 \times 10^{-7}$ $Err_c = 1.9316 \times 10^{-7}$	$Err_{uc} = 5.062 \times 10^{-9}$ $Err_c = 5.062 \times 10^{-9}$	$Err_{uc} = 2.739 \times 10^{-9}$ $Err_c = 2.739 \times 10^{-9}$	$Err_{uc} = 2.739 \times 10^{-9}$ $Err_c = 2.739 \times 10^{-9}$
41	$Err_{uc} = 1.8241 \times 10^{-7}$ $Err_c = 1.8241 \times 10^{-7}$	$Err_{uc} = 5.196 \times 10^{-9}$ $Err_c = 5.196 \times 10^{-9}$	$Err_{uc} = 2.563 \times 10^{-9}$ $Err_c = 2.563 \times 10^{-9}$	$Err_{uc} = 2.530 \times 10^{-9}$ $Err_c = 2.530 \times 10^{-9}$

TABLE A.3

is reduced to 9.4011×10^{-8}, which is a small improvement. These results can be compared to the one dimensional case in Test Problem 2 and Table A.2. The findings for the one dimensional and the two dimensional case are very similar.

– ***Test Problem 4:*** The following triplet $(\rho, \vec{w}, q)$ solve the problem (2.6) with the zero Dirichlet boundary condition (2.4), and $\gamma = 0$:

$$\rho = 2e^t \prod_{i=1}^{d} \cos\left(\frac{\pi x_i}{2}\right),$$
$$w = -\frac{1}{\beta}(e^T - e^t) \prod_{i=1}^{d} \cos\left(\frac{\pi x_i}{2}\right) \quad [q = -\beta w],$$

where $\vec{x} = [x_1, ..., x_d]^T$, and

$$\widehat{\rho} = \left[-\frac{d\pi^2}{4}e^T + \left(1 + \frac{d\pi^2}{4}\right)e^t\right] \prod_{i=1}^{d} \cos\left(\frac{\pi x_i}{2}\right)$$
$$- \frac{\pi^2}{4}(e^T - e^t) \sum_{i=1}^{d} \sin^2\left(\frac{\pi x_i}{2}\right) \prod_{j=1, j\neq i}^{d} \cos^2\left(\frac{\pi x_j}{2}\right),$$
$$V_{\text{ext}} = \prod_{i=1}^{d} \cos\left(\frac{\pi x_i}{2}\right),$$
$$f = \left[\frac{1}{\beta}e^T + \left(2 + \frac{d\pi^2}{2} - \frac{1}{\beta}\right)e^t\right] \prod_{i=1}^{d} \cos\left(\frac{\pi x_i}{2}\right) + \frac{d\pi^2}{2}e^t \prod_{i=1}^{d} \cos^2\left(\frac{\pi x_i}{2}\right)$$
$$- \frac{\pi^2}{2}e^t \sum_{i=1}^{d} \sin^2\left(\frac{\pi x_i}{2}\right) \prod_{j=1, j\neq i}^{d} \cos^2\left(\frac{\pi x_j}{2}\right),$$
$$\rho_0 = 2 \prod_{i=1}^{d} \cos\left(\frac{\pi x_i}{2}\right),$$

Table A.4 shows the results for $d = 1$ and the controlled problem solved by the fixed point method. In this problem, 20 points seem to be enough to get accurate results.

– ***Test Problem 5:*** The following triplet $(\rho, \vec{w}, q)$ solve the problem (2.6) with

β / $N_1 \times N_2$, n	10^{-3}	10^{-1}	10	10^3
20	$Err_{uc} = 1.5572 \times 10^{-8}$ $Err_c = 1.0843 \times 10^{-16}$	$Err_{uc} = 1.5572 \times 10^{-8}$ $Err_c = 1.0843 \times 10^{-16}$	$Err_{uc} = 1.5572 \times 10^{-8}$ $Err_c = 1.0843 \times 10^{-16}$	$Err_{uc} = 1.5572 \times 10^{-8}$ $Err_c = 1.0843 \times 10^{-16}$
30	$Err_{uc} = 1.5756 \times 10^{-8}$ $Err_c = 9.186 \times 10^{-17}$	$Err_{uc} = 1.5756 \times 10^{-8}$ 9.186×10^{-17}	$Err_{uc} = 1.5756 \times 10^{-8}$ $Err_c 9.186 \times 10^{-17}$	$Err_{uc} = 1.5756 \times 10^{-8}$ $Err_c = 9.186 \times 10^{-17}$
40	$Err_{uc} = 1.5591 \times 10^{-8}$ $Err_c = 1.1053 \times 10^{-16}$	$Err_{uc} = 1.5591 \times 10^{-8}$ $Err_c = 1.1053 \times 10^{-16}$	$Err_{uc} = 1.5591 \times 10^{-8}$ $Err_c = 1.1053 \times 10^{-16}$	$Err_{uc} = 1.5591 \times 10^{-8}$ $Err_c = 1.1053 \times 10^{-16}$

TABLE A.4
Update table!

the boundary condition (2.8), and $\gamma = 0$:

$$\rho = 2e^t \prod_{i=1}^{d} \cos(\pi x_i),$$
$$w = -\frac{1}{\beta}(e^T - e^t) \prod_{i=1}^{d} \cos(\pi x_i) \quad [q = -\beta w],$$

where

$$\widehat{\rho} = \left[-d\pi^2 e^T + (1 + d\pi^2)e^t\right] \prod_{i=1}^{d} \cos(\pi x_i) - \pi^2(e^T - e^t) \sum_{i=1}^{d} \sin^2(\pi x_i) \prod_{j=1, j\neq i}^{d} \cos^2(\pi x_j),$$
$$V_{\text{ext}} = \prod_{i=1}^{d} \cos(\pi x_i),$$
$$f = \left[\frac{1}{\beta}e^T + \left(2 + 2d\pi^2 - \frac{1}{\beta}\right)e^t\right] \prod_{i=1}^{d} \cos(\pi x_i) + 2d\pi^2 e^t \prod_{i=1}^{d} \cos^2(\pi x_i) - 2\pi^2 e^t \sum_{i=1}^{d} \sin^2(\pi x_i) \prod_{j=1, j\neq i}^{d} \cos^2(\pi x_j),$$
$$\rho_0 = 2 \prod_{i=1}^{d} \cos(\pi x_i),$$

Table A.5 shows the results for $d = 1$ and the controlled problem. It can be observed that in this case 20 points are sufficient. This could be explained by the fact that Problem (2.6) is easier to solve than Problem (2.5).

[[we should check these, possibly using Maple]]

Appendix B. Perturbing test problems (working title).

Since it is necessary to provide an initial guess for the control $\vec{w}$ to start the optimization routine, as detailed in Section 4.3, one method of validation is to perturb the exact solution for $\vec{w}$ and use this as an initial guess in the optimization solver. In the first iteration, this results in a different solution for ρ and q than the exact solution. The optimization method converges to the exact solution, which is the/an

β / $N_1 \times N_2$, n	10^{-3}	10^{-1}	10	10^3
20	$Err_{uc} = 1.5605 \times 10^{-8}$ $Err_c = 1.0395 \times 10^{-16}$	$Err_{uc} = 1.5605 \times 10^{-8}$ $Err_c = 1.0395 \times 10^{-16}$	$Err_{uc} = 1.5605 \times 10^{-8}$ $Err_c = 1.0395 \times 10^{-16}$	$Err_{uc} = 1.5605 \times 10^{-8}$ 1.0395×10^{-16}
30	$Err_{uc} = 1.5366 \times 10^{-8}$ $Err_c = 1.1296 \times 10^{-16}$	$Err_{uc} = 1.5366 \times 10^{-8}$ $Err_c = 1.1296 \times 10^{-16}$	$Err_{uc} = 1.5366 \times 10^{-8}$ $Err_c = 1.1296 \times 10^{-16}$	$Err_{uc} = 1.5366 \times 10^{-8}$ 1.1296×10^{-16}
40	$Err_{uc} = 9.298 \times 10^{-9}$ $Err_c = 8.992 \times 10^{-17}$	$Err_{uc} = 1.2436 \times 10^{-8}$ $Err_c = 8.992 \times 10^{-17}$	$Err_{uc} = 1.2584 \times 10^{-8}$ $Err_c = 8.992 \times 10^{-17}$	$Err_{uc} = 1.2584 \times 10^{-8}$ $Err_c = 8.992 \times 10^{-17}$

TABLE A.5

optimal solution. We consider Test Problem 2 (maybe tag this) in Appendix A, which is an exact solution for Problem (2.2), with boundary conditions (2.5), and $\gamma = 0$. The following two perturbation functions are considered. The first perturbation is in time only and is defined as:

$$\begin{aligned} g(t) &= \frac{1}{2} f(t - t_0, a) \times f(t - t_0, -a) \\ &= \frac{1}{2} \frac{e^{-a/(t-t_0)}}{e^{-a/(t-t_0)} + e^{-a/(1-t-t_0)}} \times \frac{e^{a/(t-t_0)}}{e^{a/(t-t_0)} + e^{a/(1-t-t_0)}}. \end{aligned}$$

The normalised version is then

$$\tilde{g}(t) = \frac{g(t)}{\max |g(t)|},$$

so that $\max \tilde{g}(t) = 1$. A similar perturbation can be done in space, taking into account the difference in length of spatial and time domain:

$$\begin{aligned} h(x) &= \frac{1}{2} f(x - x_0, 2a) \times f(x - x_0, -2a) \\ &= \frac{1}{2} \frac{e^{-2a/(x-x_0)}}{e^{-2a/(x-x_0)} + e^{-2a/(1-x-x_0)}} \times \frac{e^{2a/(x-x_0)}}{e^{2a/(x-x_0)} + e^{2a/(1-x-x_0)}}. \end{aligned}$$

Again, the normalised version is:

$$\tilde{h}(x) = \frac{h(x)}{\max |h(x)|}.$$

These perturbation functions are chosen such that the perturbation is smooth and respects the initial condition for ρ as well as the final time condition for q, by not changing the first or final time point, see Figure B.1. The considered perturbations are applied to the exact control $\vec{w}_{ex}$ as follows:

$$\begin{aligned} \vec{w}_{pert1} &= \vec{w}_{ex}(1 + \epsilon \tilde{g}(t)) \\ \vec{w}_{pert2} &= \vec{w}_{ex}(1 + \epsilon \tilde{g}(t)\tilde{h}(x)), \end{aligned}$$

where $a = 0.7$, $x_0 = t_0 = -0.01$ and $\epsilon = 0.1$ or $\epsilon = 0.5$. The chosen number of points is $N = 40$ and $n = 41$, the ODE tolerances are 10^{-8} and Optimization tolerance is 10^{-4}. The mixing rate for the optimization solver is $\lambda = 0.01$. The results presented in Table B.1 show the initial error in $\vec{w}$, and the final errors in $\vec{w}$, ρ and q, measured in the norm presented in Section 4.3, with respect to the exact solution. It can be

	β	Initial Error $\vec{w}$	Final $\vec{w}$	Error ρ	q
$0.1\tilde{g}(t)$	10^{-3}	0.1	6.3190×10^{-5}	1.1648×10^{-5}	2.8425×10^{-5}
	10^{-1}	0.1	6.1449×10^{-5}	7.9587×10^{-5}	2.8405×10^{-4}
	10^{1}	0.1	6.1304×10^{-5}	7.9576×10^{-5}	5.8824×10^{-4}
$0.5\tilde{g}(t)$	10^{-3}	0.5	2.7263×10^{-4}	2.9182×10^{-5}	3.2882×10^{-5}
	10^{-1}	0.5	2.7064×10^{-4}	2.7098×10^{-4}	3.2894×10^{-4}
	10^{1}	0.5	2.7079×10^{-4}	2.7096×10^{-4}	6.8139×10^{-4}
$0.1\tilde{h}(x)$	10^{-3}	0.0855	6.1809×10^{-5}	1.1842×10^{-5}	2.7355×10^{-5}
	10^{-1}	0.0855	6.0187×10^{-5}	8.0212×10^{-5}	2.7321×10^{-4}
	10^{1}	0.0855	6.0047×10^{-5}	8.0200×10^{-5}	5.7592×10^{-4}
$0.5\tilde{h}(x)$	10^{-3}	0.4276	2.5041×10^{-4}	2.7429×10^{-5}	3.1303×10^{-5}
	10^{-1}	0.4276	2.4833×10^{-4}	2.5436×10^{-4}	3.1332×10^{-4}
	10^{1}	0.4276	2.4846×10^{-4}	2.5435×10^{-4}	6.4922×10^{-4}

TABLE B.1

seen that most of the errors are within the optimization tolerance, such as most final errors for $\vec{w}$ and ρ. Since in the optimization routine q is obtained using the final values of $\vec{w}$ and ρ it can be observed that the error in q is generally higher than for the other two variables. Overall, these results show that the method works reliably.

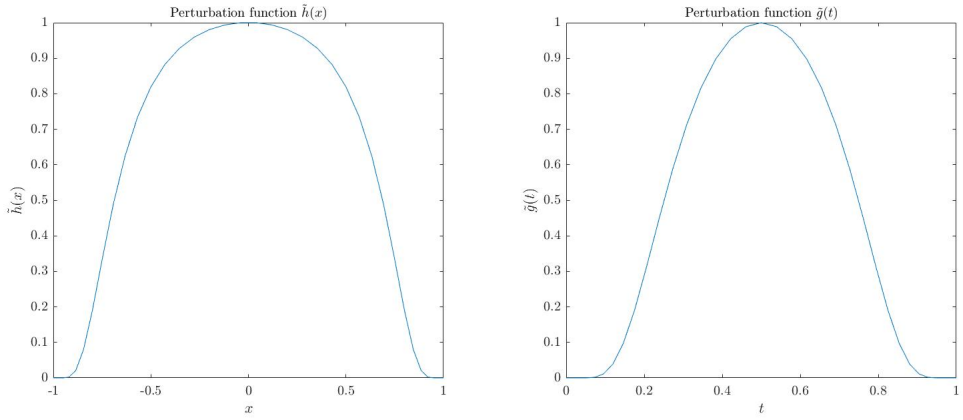

FIG. B.1. *Perturbation functions* $\tilde{h}(x)$ *and* $\tilde{g}(t)$

REFERENCES

[1] Y. Achdou and V. Perez. Iterative strategies for solving linearized discrete mean field games systems. *Netw. Heterog. Media*, 7(2):197–217, 2012.

[2] G. Albi, M. Bongini, E. Cristiani, and D. Kalise. Invisible control of self-organizing agents leaving unknown environments. *SIAM J. Appl. Math.*, 76(4):1683–1710, 2016.

[3] G. Albi, Y.-P. Choi, M. Fornasier, and D. Kalise. Mean field control hierarchy. *Appl. Math. Opt.*, 76(1):93–135, 2017.

[4] G. Albi and D. Kalise. (Sub)Optimal feedback control of mean field multi-population dynamics. *IFAC-PapersOnLine*, 51(3):86–91, 2018.

[5] Wolfgang Alt, Mark Chaplain, Michael Griebel, and Jürgen Lenz. *Polymer and cell dynamics: Multiscale modelling and numerical simulations*. Birkhäuser, 2012.
[6] R. Andreev. Preconditioning the augmented Lagrangian method for instationary mean field games with diffusion. *SIAM J. Sci. Comput.*, 39(6):A2763–A2783, 2017.
[7] Andrew J Archer, Blesson Chacko, and Robert Evans. The standard mean-field treatment of inter-particle attraction in classical dft is better than one might expect. *The Journal of chemical physics*, 147(3):034501, 2017.
[8] Hug Aubin, Jason W Nichol, Ché B Hutson, Hojae Bae, Alisha L Sieminski, Donald M Cropek, Payam Akhyari, and Ali Khademhosseini. Directed 3d cell alignment and elongation in microengineered hydrogels. *Biomaterials*, 31(27):6941–6951, 2010.
[9] James Binney and Scott Tremaine. *Galactic dynamics*, volume 20. Princeton university press, 2011.
[10] M. Bongini and G. Buttazo. Optimal control problems in transport dynamics. *Math. Models Methods Appl. Sci.*, 27(3):427–451, 2017.
[11] John P Boyd. *Chebyshev and Fourier spectral methods*. Courier Corporation, 2001.
[12] L. M. Briceño Aras, D. Kalise, and F. J. Silva. Proximal methods for stationary mean field games with local couplings. *SIAM J. Control Opt.*, 56(2):801–836, 2018.
[13] Maria Bruna and S Jonathan Chapman. Excluded-volume effects in the diffusion of hard spheres. *Physical Review E*, 85(1):011103, 2012.
[14] M. Burger, M. Di Francesco, P. A. Markowich, and M.-T. Wolfram. Mean field games with nonlinear mobilities in pedestrian dynamics. *Discrete and Continuous Dynamical Systems - Series B*, 19(5):1311 –1333, 2014.
[15] Garnet Kin-Lic Chan and Reimar Finken. Time-dependent density functional theory of classical fluids. *Physical review letters*, 94(18):183001, 2005.
[16] S. S. Collis and M. Heinkenschloss. Analysis of the streamline upwind/Petrov Galerkin method applied to the solution of optimal control problems. Technical Report TR02–01, Department of Computational and Applied Mathematics, Rice University, 2002.
[17] E. Cristiani and D. Peri. Robust design optimization for egressing pedestrians in unknown environments. *Appl. Math. Model.*, 72:553–568, 2019.
[18] Emiliano Cristiani, Benedetto Piccoli, and Andrea Tosin. *Multiscale modeling of pedestrian dynamics*, volume 12. Springer, 2014.
[19] R. Evans. The nature of the liquid-vapour interface and other topics in the statistical mechanics of non-uniform, classical fluids. *Adv. Phys.*, 28(2):143, 1979.
[20] Robert Evans. Density functionals in the theory of nonuniform fluids. *Fundamentals of inhomogeneous fluids*, 1:85–176, 1992.
[21] B. D. Goddard, A. Nold, and S. Kalliadasis. 2DChebClass [Software]. http://dx.doi.org/10.7488/ds/1991, 2017.
[22] Jean-Pierre Hansen and Ian Ranald McDonald. *Theory of simple liquids: with applications to soft matter*. Academic Press, 2013.
[23] Pierre Hohenberg and Walter Kohn. Inhomogeneous electron gas. *Physical review*, 136(3B):B864, 1964.
[24] LD Landau and EM Lifshitz. Statistical physics, part 1, vol. 5. *Course of theoretical physics*, 3, 1994.
[25] Ben Leimkuhler and Charles Matthews. *Molecular Dynamics*. Springer, 2016.
[26] Jan Lorenz. Continuous opinion dynamics under bounded confidence: A survey. *International Journal of Modern Physics C*, 18(12):1819–1838, 2007.
[27] J. F. Lutsko. Recent developments in classical density functional theory. *Adv. Chem. Phys.*, 144:1, 2010.
[28] James F Lutsko. A dynamical theory of nucleation for colloids and macromolecules. *The Journal of chemical physics*, 136(3):034509, 2012.
[29] U. M. B. Marconi and P. Tarazona. Dynamic density functional theory of fluids. *The Journal of Chemical Physics*, 110(16):8032–8044, 1999.
[30] Joachim Messer and Herbert Spohn. Statistical mechanics of the isothermal lane-emden equation. *Journal of Statistical Physics*, 29(3):561–578, 1982.
[31] Andreas Nold, Benjamin D Goddard, Peter Yatsyshin, Nikos Savva, and Serafim Kalliadasis. Pseudospectral methods for density functional theory in bounded and unbounded domains. *Journal of Computational Physics*, 334:639–664, 2017.
[32] J. W. Pearson and M. Stoll. Fast iterative solution of reaction–diffusion control problems arising from chemical processes. *SIAM J. Sci. Comput.*, 35(5):B987–B1009, 2013.
[33] J. W. Pearson, M. Stoll, and A. J. Wathen. Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems. *SIAM J. Matrix Anal. Appl.*, 33(4):1126–1152, 2012.

[34] M.J.D Powell. A fortran subroutine for solving systems of nonlinear algebraic equations. *Numerical Methods for Nonlinear Algebraic Equations*, Ch.7, 1970.
[35] E. Roman and W. Dieterich. Classical fluid in a periodic potential and the density-functional approach. *Phys. Rev. A*, 32(6):3726, 1985.
[36] Roland Roth. Fundamental measure theory for hard-sphere mixtures: a review. *Journal of Physics: Condensed Matter*, 22(6):063102, 2010.
[37] Lawrence F Shampine, Mark W Reichelt, and Jacek A Kierzenka. Solving index-1 daes in matlab and simulink. *SIAM review*, 41(3):538–552, 1999.
[38] M. Stoll and A. Wathen. All-at-once solution of time-dependent PDE-constrained optimization problems. Technical Report Technical Report NA-10-13, University of Oxford, 2010.
[39] Attila Szabo and Neil S Ostlund. *Modern quantum chemistry: introduction to advanced electronic structure theory*. Courier Corporation, 2012.
[40] Pedro Tarazona, José A Cuesta, and Yuri Martínez-Ratón. Density functional theories of hard particle systems. In *Theory and Simulation of Hard-Sphere Fluids and Related Systems*, pages 247–341. Springer, 2008.
[41] Lloyd N Trefethen. *Spectral methods in MATLAB*, volume 10. Siam, 2000.
[42] F. Tröltzsch. *Optimal Control of Partial Differential Equations: Theory, Methods, and Applications*. American Mathematical Society, 2010.
[43] Gregory A Voth. *Coarse-graining of condensed phase and biomolecular systems*. CRC press, 2008.
[44] J.-Z. Wu. Density functional theory for chemical engineering: From capillarity to soft materials. *AIChE J.*, 52:1169–1193, 2006.
[45] J.-Z. Wu and Z.-D. Li. Density-functional theory for complex fluids. *Annu. Rev. Phys. Chem.*, 58:85–112, 2007.
[46] Christian A Yates, Ruth E Baker, Radek Erban, and Philip K Maini. Refining self-propelled particle models for collective behaviour. 2010.