

Learning Agent Interactions from Density Evolution in 3D Regions With Obstacles

Amoolya Tirumalai, Christos N. Mavridis, John S. Baras

Abstract—In this work, we study the inverse problem of identifying complex flocking dynamics in a domain cluttered with obstacles. We get inspiration from animal flocks moving in complex ways with capabilities far beyond what current robots can do. Owing to the difficulty of observing and recovering the trajectories of the agents, we focus on the dynamics of their probability densities, which are governed by partial differential equations (PDEs), namely compressible Euler equations subject to non-local forces. We formulate the inverse problem of learning interactions as a PDE-constrained optimization problem of minimizing the squared Hellinger distance between the histogram of the flock and the distribution associated to our PDEs. The numerical methods used to efficiently solve the PDE-constrained optimization problem are described. Realistic flocking data are simulated using the Boids model of flocking agents, which differs in nature from the reconstruction models used in our PDEs. Our analysis and simulated experiments show that the behavior of cohesive flocks can be recovered accurately with approximate PDE solutions.

I. INTRODUCTION

In multi-agent (MA) systems, decisions depend critically on the interactions between agents (both local and global), the information each agent has on the dynamics and actions of other agents, and the topologies of various networks (or multigraphs) used to abstractly model the MA system (collaboration, information, and communication network) [1]. Learning emerges naturally when addressing inference problems in MA systems associated with learning the dynamics of other agents [2], [3], learning the interactions or coordination laws [3], [4], [5], and learning to collaborate in decision making, inference or attention [6], [7].

A still-open problem in MA systems is how one can discover or characterize the interaction (or coordination) schemes that govern agent dynamics from data [4], [2], [8], [9], [10]. This problem is especially interesting in discovering how animals interact in flocks, swarms, and herds. Animals which move in these large collectives behave in emergent ways that are far more complex and useful than any behaviors achieved thus far by robots. They are also able to perform these maneuvers in very dense formations and within regions cluttered with obstacles. Beyond the interesting conclusions related to the basic science which one gains by discovering the interaction rules of such animals, one can also try to map these behaviors onto robots, especially when structured models are used. There are even more

challenging and important problems where such inspiration and knowledge can be used, including social networks over the Internet, collaborating human-machine MA systems, and defence mechanisms against aerial UAV swarm attacks [5].

There are generally two broad classes of model-based approaches used to solve this problem: microscopic agent models, which have dynamics given by (typically nonlinear) ordinary or stochastic differential equations, and macroscopic models, which are described with partial differential equations [11], [12], [2]. However, one of the main problems in studying natural flocking lies in extracting information from data. Extracting useful and accurate trajectories of the agents has been shown to be very difficult [2], [13]. On the other hand, extracting the empirical distribution of the dynamics (or more precisely, functionals of the empirical distributions, such as histograms) can be less prone to noise, as one extracts this information using operators which exhibit averaging properties. Hence, we focus on the macroscopic approach. The macroscopic models are generally understood as mean-field limits (which can be in either a rigorous or formal sense) of microscopic model dynamics. The area of study which describes passage from microscopic to mesoscopic and then macroscopic models is described in kinetic and hydrodynamic theory, and is a thoroughly studied area of research in applied mathematics [14], [10].

A. Contribution.

Our objective in this work is to discover the interaction (or coordination) dynamics of a large swarm of agents interacting within a workspace cluttered with obstacles, via observations of their density, i.e., their probability distribution in three spatial dimensions, plus time.

We generate 3D particle trajectories using a version of the Boids model [15]. Such models have been used extensively to generate very realistic scenes and system trajectories and associated data. We bound the Boids to a cubical region of space with obstacles formed from cubes, and subject them to specular reflections. From these trajectories, we extract the histograms of the density and momentum profiles. We represent the initial mean-field distribution and momentum profile as a deep neural network for the initialization of the PDE describing the mean-field spatial probability distribution. The time-evolving spatial probability distribution estimates are then used as continuous observations of the density of the swarm. To discover the interaction (or coordination) laws, which we parametrize with a real vector, we pose a PDE-constrained optimization problem to minimize the squared Hellinger distance between the Boid histogram and the

Department of Electrical and Computer Engineering and the Institute for Systems Research, University of Maryland, College Park, MD 20742, USA. Emails: {ast256, mavridis, baras}@umd.edu

This material is based upon work supported partially by ONR grant N00014-17-1-2622 (AT, CM, and JB), and by the Ann G. Wylie Dissertation Fellowship from the University of Maryland (Tirumalai).

hydrodynamic mean-field distribution. We solve this problem numerically and approximately using the Newton-conjugate gradient method. To the authors' knowledge, this is the first instance of a macroscopic MA system identification problem for flocking being solved in three spatial dimensions in the presence of obstacles.

B. Related Work

In our previous work, we discussed the tradeoffs associated to particle and density-based approaches to discovering swarm interaction laws [4], and explored this problem in two spatial dimensions, plus time [2]. Accurate and efficient solvers for the hydrodynamic Cucker-Smale model were developed which employed the Kurganov-Tadmor finite volume method [16] to compute the hyperbolic part of the PDEs, and spectral solvers which computed the nonlocal part. In [17], we rigorously studied the hydrodynamic model that will be adopted and modified slightly in this work. We established existence and weak-strong uniqueness for this model in smooth regions. Our work was influenced by [10], where the authors followed a similar approach modeling the interactions with respect to a fractional differential system of equations. More recently, the same problem is studied from the perspective of physics-informed neural networks [8].

C. Notation

Throughout this work, we use ‘...’ to indicate line continuation for long equations or expressions. We use the following notations for the inner product and outer products for real vectors $v, w \in \mathbb{R}^3$:

$$v^\top w \equiv \langle v, w \rangle_{\mathbb{R}^3}, v w^\top \equiv v \otimes w,$$

where v^\top is the transposition of v . We use $l : \mathbf{L}(S) \rightarrow \mathbb{R}_0^+$ to denote the standard Lebesgue measure, $\mathbf{L}(S)$ being the σ -algebra of Lebesgue-measurable sets contained in S . If S is bounded and smooth(-enough), $\mathbf{n} : \partial S \rightarrow \partial B(0, 1)$ is the outward-pointing surface normal vector on ∂S . We use $\mathbb{I}_A : \mathbb{R}^3 \rightarrow \{0, 1\}$ to denote the characteristic (indicator) function of a set A . For some countable set B , $|B|$ is the cardinality of that set. Given a sequence of N positions, $\mathbf{S} := \{x^i\}_{i=1}^N \subset \mathbb{R}^3$, the neighborhood of a point $x^i \in \mathbf{S}$ is:

$$N^i(\epsilon) := \{x^j \in \mathbf{S} : \|x^i - x^j\|_{\mathbb{R}^3} \leq \epsilon, x^i \neq x^j\}.$$

For a set S , $\mathcal{B}(S)$ is the Borel σ -algebra on S . Throughout, $(\mathbf{x}, \mathbf{v}) := (x^1, \dots, x^N, v^1, \dots, v^N)$.

II. MODEL FORMULATION

In this section, we describe the physical area where we assume agents are free to move, the model we use to generate synthetic data, and the model which we use to perform system identification.

A. Workspace

We assume that all agents are contained in a bounded polyhedral region $D \subset \mathbb{R}^3$. We assume that we have:

$$D_m := \{x \in \mathbb{R}^3 : \|x - \xi_m\|_\infty \leq R_m\}, \xi_1 = \mathbf{0}_{\mathbb{R}^3}$$

and $D_m \cap D_n = \emptyset, 2 \leq m, n \leq M, R_1 > R_m, 2 \leq m \leq M, \bigcup_{m=2}^M D_m \subset D_1$. Then, we define:

$$D := D_1 \setminus \bigcup_{m=2}^M D_m.$$

B. Boids

We generate flocking data according to a version of the Boids model [15]. Consider agent position-velocity pairs

$$\{(x^{i,\top}, v^{i,\top})(\cdot)\}_{i=1}^N, (x^i, v^i) : [0, T] \rightarrow D \times \mathbb{R}^3,$$

subject to ODEs:

$$\begin{aligned} \frac{d}{dt} x^i(t) &= v^i(t); \\ \frac{d}{dt} v^i(t) &= \sum_{j=1}^5 \mathbf{F}_j^i(\mathbf{x}(t), \mathbf{v}(t)); \\ (x^{i,\top}, v^{i,\top})^\top(0) &= (x_0^{i,\top}, v_0^{i,\top})^\top \sim \mathbb{P}(0), \end{aligned} \quad (1)$$

where $\mathbb{P}(0) \in \mathcal{P}(D)$, is a probability measure on D a.c. w.r.t. the Lebesgue measure. The forces are:

$$\mathbf{F}_1^i(\mathbf{x}, \mathbf{v}) := \frac{1}{|N^i(1)|} \sum_{k=1}^{|N^i(1)|} (v^k - v^i);$$

$$\mathbf{F}_2^i(\mathbf{x}, \mathbf{v}) := \frac{1}{|N^i(2)|} \sum_{k=1}^{|N^i(2)|} (x^k - x^i);$$

$$\mathbf{F}_3^i(\mathbf{x}, \mathbf{v}) := \frac{1}{|N^i(\frac{1}{2})|} \sum_{k=1}^{|N^i(\frac{1}{2})|} \frac{x^i - x^k}{\|x^i - x^k\|_{\mathbb{R}^3}^2};$$

$$\mathbf{F}_4^i(\mathbf{x}, \mathbf{v}) := -\left(\frac{1}{16} \|v^i\|_{\mathbb{R}^3}^2 - 1\right) v^i;$$

$$\mathbf{F}_5^i(\mathbf{x}, \mathbf{v}) := -4 \sum_{k=1}^M \frac{\mathbb{I}_{r < 1}(d(x^i, \partial D_m))}{d(x^i, \partial D_m)} \frac{x^i - \xi_k}{\|x^i - \xi_k\|_{\mathbb{R}^3}}.$$

It is also assumed that each particle is subject to specular reflections:

$$\lim_{t_{i,j}^{*+} \downarrow t_{i,j}^{*-}} v^j(t_{i,j}^{*+}) = (\mathbf{I}_3 - 2\mathbf{n}(x^j(t_{i,j}^*))\mathbf{n}^\top(x^j(t_{i,j}^*)))v^j(t_{i,j}^*),$$

where $t_{i,j}^* \in (0, T]$ is the time of the i -th collision of the j -th agent with the boundary ∂D .

C. Hydrodynamic Model

In the interest of saving space, we direct the reader to our previous work on the Euler Alignment System for a description of the process one takes to obtain, at least in form, a hydrodynamic model with non-penetrative boundary conditions from a particle-based model with specular reflections [17]. This approach is based on [18], [19]. The probability density of a collective of particles $\rho : [0, T] \times D \rightarrow \mathbb{R}$ and

the momentum density $\mathbf{j} : [0, T] \times D \rightarrow \mathbb{R}^3$ evolve according to non-locally forced compressible Euler equations:

$$\begin{aligned}
& \partial_t \rho + \nabla_x \cdot \mathbf{j} = 0 \text{ in } (0, T] \times D; \\
& \partial_t \mathbf{j} + \nabla_x \cdot [\rho^{-1} \mathbf{j} \mathbf{j}^\top] = \mathbf{S}[\rho, \mathbf{j}] \text{ in } (0, T] \times D; \\
& \Lambda_a \pi_\rho[\rho](t, \cdot) = \mathbb{I}_D \rho(t, \cdot) \text{ in } \mathbb{R}^3; \\
& \Lambda_a \pi_{\mathbf{j}}[\mathbf{j}](t, \cdot) = \mathbb{I}_D \mathbf{j}(t, \cdot) \text{ in } \mathbb{R}^3; \\
& \Lambda_c v_c[\rho](t, \cdot) = \mathbb{I}_D \rho(t, \cdot) \text{ in } \mathbb{R}^3; \\
& \Lambda_r v_r[\rho](t, \cdot) = \mathbb{I}_D \rho(t, \cdot) \text{ in } \mathbb{R}^3; \\
& \mathbf{S}[\rho, \mathbf{j}] := \rho \pi_{\mathbf{j}}[\mathbf{j}] - \mathbf{j} \pi_\rho[\rho] - \dots \\
& \rho \nabla_x [\mathbf{1}_2^\top \mathbf{V}[\rho] + U] + k_p \mathbf{j} (1 - F(\rho^{-1} \|\mathbf{j}\|_{\mathbb{R}^3})); \\
& \mathbf{j}^\top \mathbf{n} = 0 \text{ in } [0, T] \times \partial D; \\
& (\rho(0, \cdot), \mathbf{j}(0, \cdot)) = (\rho_0, \mathbf{j}_0) \text{ in } D,
\end{aligned} \tag{2}$$

where $\mathbf{V}[\rho] := (v_c[\rho], v_r[\rho])^\top$, and the operators $\Lambda_{(\cdot)}$ are:

$$\frac{1}{4\pi k_{(\cdot)}} (\nabla_x^2 - \lambda_{(\cdot)})$$

which is a Bessel-type operator with fundamental solution [17]:

$$G(x, s; k_{(\cdot)}, \lambda_{(\cdot)}) = \frac{k_{(\cdot)} e^{-\lambda_{(\cdot)} \|x-s\|_{\mathbb{R}^3}}}{\|x-s\|_{\mathbb{R}^3}}.$$

We do not specify concretely the domains and ranges of these operators, as we feel that the functional analysis required to define them clearly does not contribute much to the thrust of the paper, which is application-focused. Those who desire details are encouraged to refer to [17]. For vector functions, take the operator to apply element-wise.

We take $F(s) := 1 + \tanh(\frac{s^2}{\lambda_p^2} - 1)$. Finally, we take:

$$U(\cdot) := k_o \eta(\cdot; \omega_o) * (\mathbb{I}_{d(\xi, \partial D_1) < \omega_o}(\cdot) + \sum_{i=2}^M \mathbb{I}_{\xi \in D_i}(\cdot)),$$

where $\eta(\cdot; \epsilon)$ is the standard mollifier/ bump function [20].

III. PROBLEM FORMULATION

In this section, we frame the optimization problems used for system identification. First, let us clearly define some quantities. For two probability measures $P, Q \in \mathcal{P}(D)$ which are a.c. w.r.t. the Lebesgue measure on D , the squared Hellinger distance is:

$$H^2(P, Q) := \frac{1}{2} \int_D (\sqrt{p(x)} - \sqrt{q(x)})^2 dx,$$

where p, q are the densities (Radon-Nikodym derivatives) of P, Q w.r.t. the Lebesgue measure, respectively. Let $v_{max} := \max_{(t,i) \in [0, T] \times \{1, \dots, N\}} \|v^i(t)\|_\infty$, and define a sequence of regions: $\{\Omega_{ijk} \times \Upsilon_{lmn}\}_{i,j,k,l,m,n=1}^{\mathfrak{N}}$ which cover $[-R_1, R_1]^3 \times [-v_{max}, v_{max}]^3$ where Ω_{ijk} has volume $\Omega := (2R_1 \setminus \mathfrak{N})^3$, and Υ_{lmn} has volume $\Upsilon := (2v_{max} \setminus \mathfrak{N})^3$. Now, we define the position-velocity histogram over these cells as:

$$\mu^N(t, \Omega_{ijk}, \Upsilon_{lmn}) := \frac{1}{N} \sum_{\nu=1}^N \mathbb{I}_{(x,v) \in \Omega_{ijk} \times \Upsilon_{lmn}}(x^\nu(t), v^\nu(t)).$$

Via the histogram, we define a position-velocity density as:

$$m^N(t, x, v) := \dots \frac{1}{\Omega \Upsilon} \sum_{i,j,k,l,m,n=1}^{\mathfrak{N}} \mathbb{I}_{(\xi,v) \in \Omega_{ijk} \times \Upsilon_{lmn}}(x, v) M^N(t, \Omega_{ijk}, \Upsilon_{lmn}),$$

associated to measure for $A \in \mathcal{B}([-R_1, R_1]^3 \times [-v_{max}, v_{max}]^3)$:

$$M^N(t, A) := \int_A m^N(t, x, v) dl(x, v).$$

The position histogram is:

$$q(t, x) := \int_{v \in \mathbb{R}^3} dM^N(t, x, v)$$

and the associated measure is:

$$Q(t, A) := \int_A q(t, x) dl(x),$$

with $A \in \mathcal{B}([-R_1, R_1]^3)$. For a solution to the PDE (2) at a particular time $t \in [0, T]$, define a measure (assuming $\rho(t, \cdot)$ is extended by 0 from D to D_1):

$$P(t, A) := \int_A \rho(t, x) dl(x) \tag{3}$$

for $A \in \mathcal{B}([-R_1, R_1]^3)$.

A. Initial Conditions

For the initial conditions, we assume that they are of the form:

$$(\rho_0, \mathbf{j}_0) = (\rho_0(\cdot; \theta), \mathbf{j}_0(\cdot; \theta))$$

where $\theta \in \mathbb{R}^d$ is a vector of neural network weights, i.e. the initial conditions are members of a family of neural network-parametrized functions. The neural network structure selected is that of [21] called the ‘‘Deep Galerkin Method’’ (DGM), which is similar to the LSTM structure [22]. The input is 3D, there are 2 DGM hidden layers with 50 DGM neurons each, and an output layer with 4 outputs, 1 for the density, and 3 for the momentum. This results in $d = 20350$ weights. The hidden layer neurons were *tanh* neurons, and the outputs were a *softplus* neuron for the density, and *tanh* neurons for the momentum. Let $P_0(\cdot; \theta)$ be the probability measure with $\rho_0(\cdot, \theta)$ as its density. The learning problem we solve for the initial conditions is:

$$\begin{aligned}
& \min_{\theta \in \mathbb{R}^d} L_1(\theta) := H^2(P_0(\cdot; \theta), Q(0, \cdot)) + \|\mathbf{j}_0(\cdot; \theta) - \hat{\mathbf{j}}_0\|_{\mathbb{R}^3}^2 \\
& \text{s.t. } \int_D dP_0(x; \theta) = 1, \rho_0(\cdot; \theta) \geq 0
\end{aligned} \tag{4}$$

where

$$\hat{\mathbf{j}}_0 := \int_{v \in \mathbb{R}^3} v dM^N(0, x, v).$$

This deep learning problem is solved in the typical way using the stochastic gradient method ADAM [23]. We scheduled the learning rate inversely w.r.t. the iteration number. The DGM network was structured so as to satisfy the constraints specified. The learned functions were used as the initial conditions for the PDEs in the system identification problem (5) posed below.

B. System Identification

In the model (2), there are $\hat{d} = 10$ parameters describing the agent interactions, which we denote by $\theta := (k_a, k_c, k_r, k_p, k_o, \lambda_a, \lambda_c, \lambda_r, \lambda_p, \lambda_o)$. We pose the following PDE-constrained optimization problem:

$$\begin{aligned} \min_{\theta \in \mathbb{R}^{\hat{d}}} L_2(\theta) &:= \frac{1}{T} \int_0^T H^2(P(t, \cdot; \theta), Q(t, \cdot)) dt \\ \text{s.t.} & \text{ (2), (3).} \end{aligned} \quad (5)$$

We solve this problem numerically using a Newton-conjugate gradient method as we did in [5].

IV. NUMERICAL METHODS

A. Numerical Integration

All integrals in this work are approximated simply using Riemann integrals. Take the discretization of D to be given by $\{\Omega_{i,j,k}\}_{i,j,k=1}^{\mathfrak{N}}$, with centroids $\{s_{i,j,k}\}_{i,j,k=1}^{\mathfrak{N}}$. In (2), notice the sequence of elliptic problems. Via the fundamental solution, for example, the first problem is solved as:

$$\pi_\rho[\rho](t, \cdot) = \int_{\mathbb{R}^3} G(\cdot, s; k_a, \lambda_a) \rho(t, s) ds,$$

which, due to the structure of $G(\cdot; k_a, \lambda_a)$, is a singular convolution integral. With a slight abuse of notation, denote:

$$G(x, s; k_a, \lambda_a) = G(x - s; k_a, \lambda_a).$$

In discretization, the convolution integral becomes the Riemann (convolution) sum:

$$\pi_\rho[\rho](t, \cdot) = \sum_{i,j,k=1}^{\mathfrak{N}} \tilde{G}(\cdot - s_{i,j,k}; k_a, \lambda_a) \rho(t, s_{i,j,k}) \Omega$$

where (assuming $\lambda_a \neq 0$):

$$\tilde{G}(x - s; k_a, \lambda_a) = \begin{cases} 3k_a \frac{1 - \exp(-\lambda_a \frac{R_1}{2\mathfrak{N}})}{\lambda_a^2 (\frac{R_1}{2\mathfrak{N}})^2} (\lambda_a \frac{R_1}{2\mathfrak{N}} + 1) & \text{if } \|x - s\|_{\mathbb{R}^3} < \frac{R_1}{2\mathfrak{N}}, \\ G(x - s; k_a, \lambda_a) & \text{else.} \end{cases}$$

so as to deal with the singularity when $x = s$. The first branch is constructed by averaging the value of $G(\cdot; k_a, \lambda_a)$ over a ball of radius $\frac{R_1}{2\mathfrak{N}}$ centered on the origin, i.e. where the singularity occurs in the kernel. To calculate the convolution sum efficiently, we take the very typical FFT-based approach.

B. Numerical ODEs and Collision Event Handling

To simulate the ODE system (1), we employ the velocity-Verlet [10] method, commonly used in particle dynamics. To deal with the collisions of agents with the boundary of the region D , we employ a typical approach taken in gas particle dynamics and molecular dynamics. The regions we deal with are cuboidal, so identifying if a particle has penetrated them in a timestep is rather simple. Between timesteps, particles are assumed to travel along a ray. To correct a boundary penetration, we trace this ray back to when it intersects the boundary of the region D . Again, as D is constructed from cubes, this is rather trivial. The time when this occurs is also calculated, and the particle is marched in time up to the

collision. Then, the particle's velocity is reflected about the surface normal, easily determined as the boundaries of the region D are entirely flat, and then the particle's position is marched up to the next timestep. Thus, the specular reflection condition specified is satisfied. Inter-agent collisions are not considered.

C. Numerical PDEs

In short, we employ a three-dimensional analogue of the second-order finite volume method we describe in our work in [2]. We do modify the fluxes in this method slightly. In this work, we use the Kurganov-Tadmor [16] flux with the min-mod flux-limiter modified with the positivity-preserving flux-limiter of [24]. The positivity-preserving flux employed is the classical Rusanov-Local Lax-Friedrichs flux. In addition, we employ adaptive time-stepping so that the CFL number of the simulation is kept below .05. Time-marching is done via the strong stability-preserving second-order Runge-Kutta method [16].

There is also a small difference in the ‘‘dissipation’’ term $\mathbf{S}[\cdot]$ in this model. In our previous work, we only applied an alignment term. In this work, we deal with several more terms in the functional $\mathbf{S}[\cdot]$. The arrays which contain the discretizations of the terms of $\mathbf{S}[\cdot]$ are simply added together.

D. Choice of Parameters

For the workspace, we take 4 cubic obstacles, each of size $R_2 = \dots = R_5 = 1$. For the outer boundary, we let $R_1 = 5$. The centers of the obstacles were the columns of:

$$\begin{bmatrix} 2.5 & 2.5 & -2.5 & -2.5 \\ 2.5 & -2.5 & 2.5 & -2.5 \\ -4 & -4 & -4 & -4 \end{bmatrix}.$$

so each obstacle rests on the bottom of the arena. As stated earlier, timestepping is adaptive. The spatial mesh was taken to be $11 \times 11 \times 11$. We simulated 2000 Boids. Their initial conditions were sampled from a Gaussian, and re-sampled if they lay outside D .

V. RESULTS

The training error in the optimization process is depicted in Fig. 1. In Fig. 2, the loss in our system identification problem is plotted against the iteration index. The Newton-conjugate gradient method brought quickly the PDE-constrained optimization problem we posed to a suboptimal solution. The Newton iterations terminated after 11 iterations. The final loss was $L_2(\theta_{11}) = 0.20008555429515784$.

The approximately optimal mean-field distribution does not become nearly as spread as the Boid distribution did. In the beginning, the squared Hellinger distance is very low, as we learned a representation of the initial conditions directly from Boid data. Then, as evolution starts, the squared Hellinger distance attains a peak, as in Fig. 3. This seems to be due to the spread of the Boid flock as opposed to the mean-field. The most concentrated regions of the Boid flock and the mean-field seem to coincide, as in Fig. 4. After this initial peak in divergence from each other, the Boid distribution and the mean-field become much more similar as

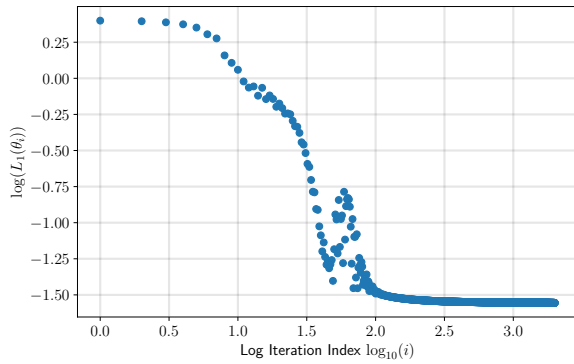


Fig. 1. Training loss L_1 of the deep learning approximation problem (4) representing the initial conditions of the distribution of the agents simulated with the Boids model.

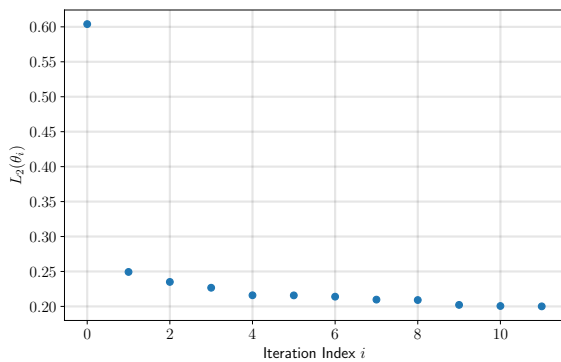


Fig. 2. The loss functional in the PDE-constrained optimization problem (5). The initial descent steps of the Newton method are quite steep, in accordance to our observations in our previous work [4].

time progresses. It is typical for the two flocks to appear as in Fig. 5, of course subject to some fluctuations in distance as shown in the plot of the Hellinger distance in Fig. 3. The times corresponding to Fig. 4 and Fig. 5 are highlighted with the blue and green dots, respectively, in Fig. 3. As mentioned in the previous section, the spatial mesh was taken to be $(11 \times 11 \times 11)$, which was sufficient to recover the dynamics in sufficient detail - flocks with finer features or subflocks obviously require finer grids to capture.

VI. CONCLUSION

In this work, we study the inverse problem of identifying complex flocking dynamics in a domain cluttered with obstacles using PDE-constrained optimization methods. Our analysis and simulated experiments show that the behavior of cohesive flocks can be recovered accurately with approximate PDE solutions with a relatively coarse mesh in our finite volume method. In addition to this result, we developed a simulator for Boids in a bounded region composed of a cube with obstacles formed from cubes, and developed a solver for a variety of Euler alignment system in the same regions. We performed the optimization in our PDE-constrained optimization problem with a Newton-conjugate gradient method.

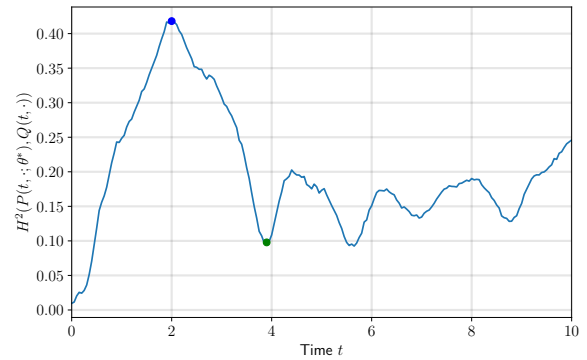


Fig. 3. The squared Hellinger distance between the Boid distribution and the mean-field distribution at each simulation time. The mean-field distribution is interpolated linearly in time to match those of the Boid simulation as it is timestepped according to its own stability criteria. We have highlighted two points. In blue, we highlight the time (time instant $t \simeq 2$) at which the boid and mean-field distributions are the most different. In green, we highlight a time (time instant $t \simeq 4$) when they are more similar. We plot these distributions in Fig. 4 and Fig. 5, respectively.

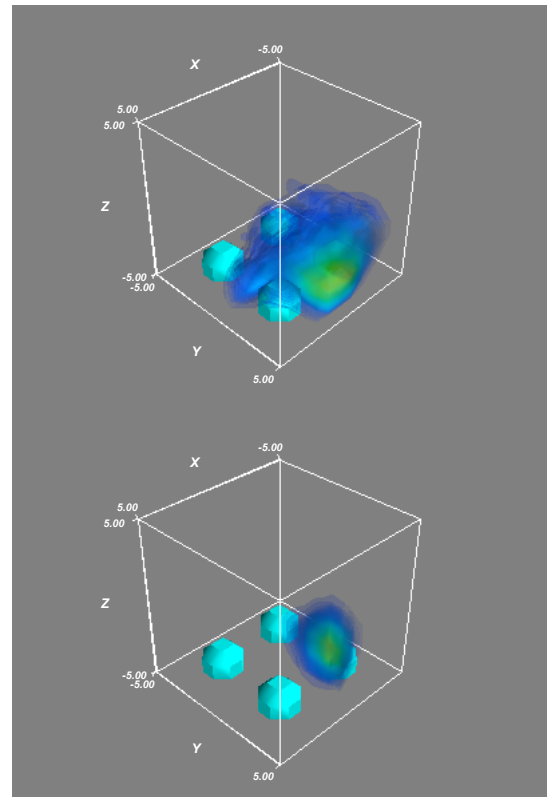


Fig. 4. (top) The Boid distribution at the time (time instant $t \simeq 2$) of maximal Hellinger distance from the mean-field distribution. (bottom) The mean-field distribution at the same time. This is labelled with a blue dot in Fig. 3.

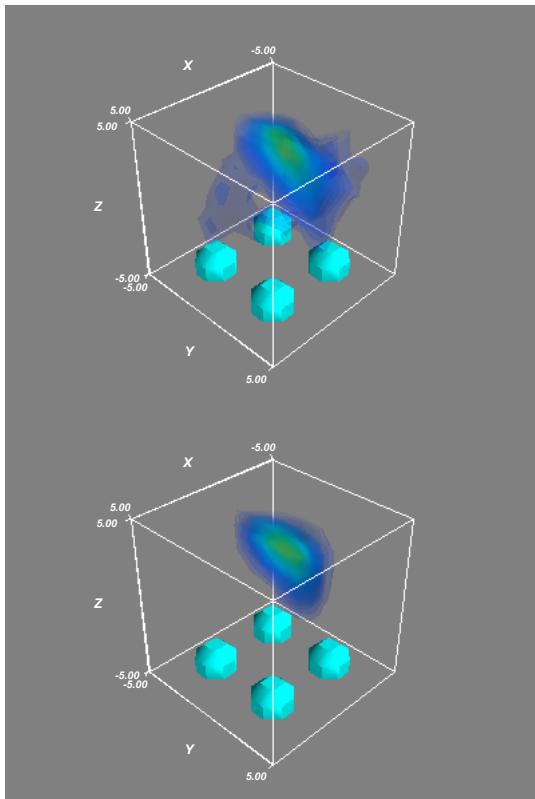


Fig. 5. (top) The Boid distribution at a time (time instant $t \simeq 4$) of lower Hellinger distance from the mean-field distribution. (bottom) The mean-field distribution at the same time. This is labelled with a green dot in Fig. 3.

In current and near future work we aim to prove that the cost functionals we employed in our learning are differentiable with respect to the parameters, providing theoretical guarantees for the convergence of our methods. We will be addressing the identification and learning problems of interaction (or coordination) laws in MA systems from observed data, albeit from local observations (i.e. with only partial information of the swarm density), and also from a few collaborating observers (sensors). In addition we are pursuing similar work for social networks over the Internet and collaborating human-machine swarms.

REFERENCES

- [1] J. S. Baras, "A fresh look at Network Science: Interdependent multi-graphs models inspired from statistical physics," in *Proceedings of the 6th International Symposium on Communication, Control and Signal Processing*. IEEE, 2014, pp. 497–500.
- [2] C. N. Mavridis, A. Tirumalai, and J. S. Baras, "Learning swarm interaction dynamics from density evolution," *IEEE Transactions on Control of Network Systems*, 2022.
- [3] I. Matei, C. N. Mavridis, J. S. Baras, and M. Zhenirovskyy, "Inferring particle interaction physical models and their dynamical properties," in *Proceedings of the 58th IEEE Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 4615–4621.
- [4] C. N. Mavridis, A. Tirumalai, and J. S. Baras, "Learning interaction dynamics from particle trajectories and density evolution," in *Proceedings of the 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 1014–1019.
- [5] C. N. Mavridis, N. Suriyarachchi, and J. S. Baras, "Detection of dynamically changing leaders in complex swarms from observed dynamic data," in *International Conference on Decision and Game Theory for Security*. Springer, 2020, pp. 223–240.

- [6] P. Hovareshti and J. S. Baras, "Learning behaviors for coordination in networked systems and a generalized consensus protocol," in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC 2011)*. IEEE, 2011, pp. 2447–2452.
- [7] J. Jiang and Z. Lu, "Learning attentional communication for multi-agent cooperation," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18)*. ACM, 2018, pp. 7265–7275.
- [8] L. Yang, C. Daskalakis, and G. E. Karniadakis, "Generative ensemble regression: Learning particle dynamics from observations of ensembles with physics-informed deep generative models," *SIAM Journal on Scientific Computing*, vol. 44, no. 1, pp. B80–B99, 2022.
- [9] A. J. Hepworth, A. Hussein, D. J. Reid, and H. A. Abbass, "Swarm analytics: Designing information markers to characterise swarm systems in shepherding contexts," *Adaptive Behavior*, p. doi:10.1177/10597123221137090, 2022.
- [10] Z. Mao, Z. Li, and G. E. Karniadakis, "Nonlocal flocking dynamics: Learning the fractional order of PDEs from particle simulations," *Communications on Applied Mathematics and Computation*, vol. 1, pp. 597–619, 2019.
- [11] J. Bongard and H. Lipson, "Automated reverse engineering of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 104, no. 24, pp. 9943–9948, 2007.
- [12] A. M. Reynolds, G. E. McIvor, A. Thornton, P. Yang, and N. T. Ouellette, "Stochastic modelling of bird flocks: accounting for the cohesiveness of collective motion," *Journal of the Royal Society Interface*, vol. 19, no. 189, p. 20210745, 2022.
- [13] M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini, *et al.*, "Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study," *Proceedings of the national academy of sciences*, vol. 105, no. 4, 2008.
- [14] J. A. Carrillo, M. Fornasier, G. Toscani, and F. Vecil, "Particle, kinetic, and hydrodynamic models of swarming," in *Mathematical modeling of collective behavior in socio-economic and life sciences*. Springer, 2010, pp. 297–336.
- [15] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 1987, pp. 25–34.
- [16] A. Kurganov and E. Tadmor, "New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations," *Journal of computational physics*, vol. 160, no. 1, pp. 241–282, 2000.
- [17] A. Tirumalai, C. Mavridis, and J. S. Baras, "Weak solutions to an Euler alignment system in a bounded domain," *arXiv preprint arXiv:2111.05361*, 2021.
- [18] H.-O. Bae, S.-Y. Ha, J. Kim, D. Ko, and S.-I. Sohn, "Flocking behaviors of a Cucker–Smale ensemble in a cylindrical domain," *SIAM Journal on Mathematical Analysis*, vol. 51, no. 3, pp. 2390–2424, 2019.
- [19] L. Desvillettes and C. Villani, "On the trend to global equilibrium for spatially inhomogeneous kinetic systems: the Boltzmann equation," *Inventiones mathematicae*, vol. 159, no. 2, pp. 245–316, 2005.
- [20] L. Evans, *Partial Differential Equations*, ser. Graduate studies in mathematics. American Mathematical Society, 2010. [Online]. Available: https://books.google.com/books?id=Xnu0o_EJrCQC
- [21] A. Al-Arabi, A. Correia, D. d. F. Naiff, G. Jardim, and Y. Saporito, "Extensions of the deep Galerkin method," *arXiv preprint arXiv:1912.01455*, 2019.
- [22] G. Van Houdt, C. Mosquera, and G. Napolos, "A review on the Long Short-Term Memory model," *Artificial Intelligence Review*, vol. 53, pp. 5929–5955, 2020.
- [23] D. P. Kingma and J. Ba, "ADAM: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.
- [24] X. Zhang and C.-W. Shu, "Positivity-preserving high order finite difference WENO schemes for compressible euler equations," *Journal of Computational Physics*, vol. 231, no. 5, pp. 2245–2258, 2012.