

Non-metric interaction rules in models of active matter

Daniel M. Sussman^{1,*}

¹*Department of Physics, Emory University, Atlanta, GA, USA*

(Dated: March 19, 2021)

This article is based on lecture notes for the Marie Curie Training school "Initial Training on Numerical Methods for Active Matter." It is common in the study of a dizzying array of soft matter systems to perform agent-based simulations of particles interacting via conservative and often short-ranged forces. In this context, well-established algorithms for efficiently computing the set of pairs of interacting particles have established excellent open-source packages to efficiently simulate large systems over long time scales – a crucial consideration given the separation in time- and length-scales often observed in soft matter. What happens, though, when we think more broadly about what it means to construct a neighbor list? What if interactions are non-reciprocal, or if the "range" of an interaction is determined not by a distance scale but according to some other consideration? As the field of soft and active matter increasingly considers the properties of living matter – from the cellular to the super-organismal scale – these questions become increasingly relevant, and encourage us to think about new physical and computational paradigms in the modeling of active matter. In this chapter we examine case studies in the use of non-metric interactions.

I. INTRODUCTION

In this chapter, we are going to think about new phenomena that emerge as soft matter moves from the study of squishy equilibrium materials to thinking about active and sometimes living "material" systems. To connect with what we've seen so far in this series, we'll begin by thinking first *not* about this physics, but about generalizations of some of the algorithms we use to efficiently perform simulations. We'll see that these algorithmic generalizations have natural analogs in physical systems. Here we go!

Agent-based numerical modeling of soft matter systems have been enormously influential in the development of the field of soft matter. A number of powerful, flexible, open-source packages for performing large-scale molecular dynamics (MD) simulations are readily available and continue to be developed [1–5], helping radically lower the barrier to entry for researchers in the field.

At its most basic level, the essential computational pattern in a molecular simulation is sketched in Alg. 1: A system of N "particles" (colloids, soft spheres, interacting Lennard-Jones units, etc.) are initialized in a simulation domain equipped with a set of boundary conditions, and a sequence of time steps is executed. At each time step, given the current state of the system, all interparticle forces are computed and the information about these forces is used to forward integrate an equation of motion – Newton's equations of motion, say, or the Langevin equation, and perhaps coupling the system degrees of freedom with a thermostat or barostat to perform simulations that appropriately sample different statistical ensembles [6]. Already we are assuming that the forces we are interested in are simple, isotropic, pairwise, conservative ones – this situation arises quite naturally in many

physical systems, but there are certainly soft matter systems that require going beyond such assumptions!

Algorithm 1 Fundamental MD loop

```
Initialize system,  $\vec{X}$ 
for Desired number of timesteps do
    Evaluate all pairwise forces,  $\vec{F}$ 
    Update  $\vec{X}$  based on  $\vec{F}$ 
end for
```

One reason that these simulation techniques have been so powerful in the study of soft matter is that they are readily parallelizable, allowing large simulations of many orders of magnitude in time to be performed on modern hardware. This is crucial, since many soft matter systems are characterized by widely separated time and length scales! For instance, a micron-scale colloid is orders of magnitude larger than the atoms in the solvent it interacts with, and the time scales of solvent motion are radically faster than the time scales of colloidal motion. Even in single-component systems this issue arises: in a melt of high-molecular-weight polymers – often represented by chains of repulsive spheres connected by stiff springs – the time scale of overall polymer motion can be much longer than the time scale associated with the vibration of adjacent units composing the polymer [7]. Thus, to faithfully integrate the equations of motion, soft matter simulations often involve very many very short individual time steps.

Earlier in this series we learned that in a naive implementation of the fundamental pattern of Alg. 1, the most computationally expensive piece is the calculation of the forces. In particular, even the evaluation of all pairwise distances between particles requires $\mathcal{O}(N^2)$ operations, and for many of the simulations we want to do N^2 is already too many. If the interparticle interactions are truly long-ranged, a variety of methods have been proposed that improve on the naive scaling, such as Ewald summation, particle-mesh methods, fast multipole methods

* daniel.m.sussman@emory.edu

[6].

In many soft matter systems, however, it is common to care about potentials that have a finite range. In this setting, we can radically reduce the computational cost of building lists of neighbors. For instance, before computing distances between particles we can spatially partition the domain into a grid of “cells” and, during the force-computation step, only compute pairwise distances between points in nearby cells. If the simulation is of particles that all have roughly the same interaction range and are roughly evenly distributed throughout the domain, computing pairwise distances among particles that exert forces on each other becomes $\mathcal{O}(N)$. Not only is this version of the calculation computationally less expensive, it is also *embarrassingly parallel* (a nice feature given the current state of computational hardware). Note, by the way, that for simulating systems with large disparities in particle sizes or densities other algorithms exist that similarly accelerate the simulation of short-ranged forces, such as using stenciled cell lists [1] or using a hierarchical tree-based structure to help partition the simulation domain rather than a grid of cells [8].

II. WON’T YOU BE MY NEIGHBOR?

We spend so much of our time thinking about physical systems evolving according to conservative forces that this connection between *neighbor lists* – lists of interacting pairs of degrees of freedom – and the interactions according to which the degrees of freedom exert forces is often assumed without much further thought. As active matter increasingly looks to build highly coarse-grained models of living systems – at the level of cells or entire organisms, as shown in Fig. 1 – we should pause and question even these fundamental assumptions.

For example, in the case studies below we will discuss classic models of flocking birds, in which each “bird” tries to align with its neighbors. Looking at a flock, or thinking about the neural processes involved in birds deciding which way to turn, it is far from clear that the “neighbor list” specifying which other birds a member of the flock is aligning with should be tied to strict distance cutoffs. Similarly, looking at a dense packing of cells, it seems clear that cell “neighbors” should be essentially determined by whether two cells share an edge – an indirect function of distance, perhaps, but one which has the character of a graph or network of interactions more than a distance-based cutoff.

These are just two of a broader class of systems that we might say interact “topologically,” that is, in which we define neighbors via something other than a strict metric criteria. Generalizing our notion of neighbor relations takes us to the forefront of some very exciting current research in the soft active matter community, ranging from the new phenomena and scaling that can be found in such non-metric models [10, 11] to the role and importance of non-reciprocal forces [12–15].

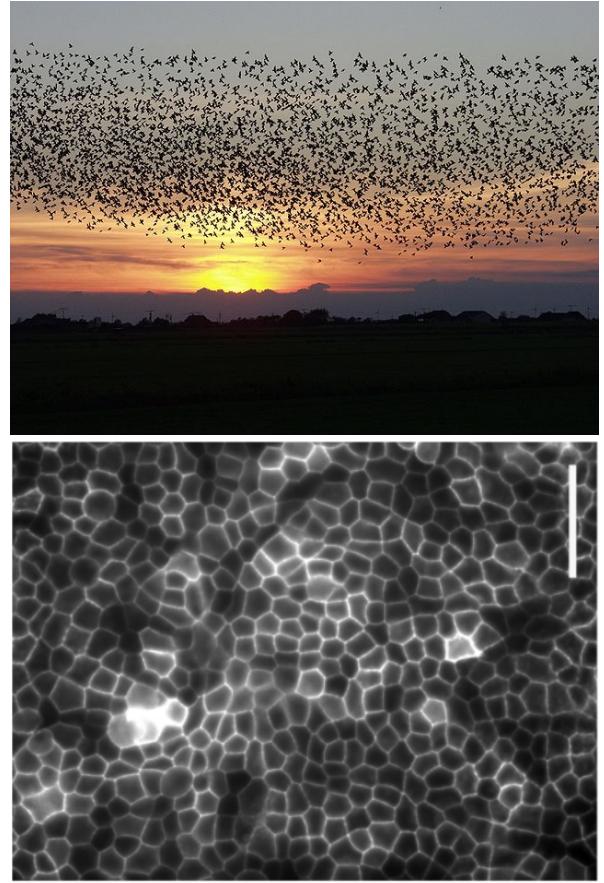


FIG. 1. (Top) A flock of starlings (Image from <http://www.pdfnet.dk> – Material is in Public Domain). (Bottom) Cross-sectional view of a monolayer of Madin-Darby Canine Kidney cells grown on a collagen substrate [9]. Scale bar is 50 microns.

III. METRIC AND TOPOLOGICAL FLOCKING MODELS

The Vicsek model [16] is one of the canonical models in the study of the active matter. The original form of it studied out-of-equilibrium point particles evolving according to a simple dynamical rule, and showed that beautiful collective organizational states emerged. Many variations of the Vicsek model have been proposed and studied, but for our purposes we will focus on the simple “vectorial” version of the model [17].

The essence of the model is to posit the overdamped dynamics of N self-propelled particles. Each particle has a position, \vec{r}_i , and an orientation, \hat{n}_i , and at each time step the positions of the particles are updated by moving at a constant speed, v_0 , in the direction of the particle’s orientation. Computationally, in a simple Eulerian integration of the equations of motion, at each time step we update the position of particle i according to

$$\Delta\vec{r}_i = (v_0\Delta t)\hat{n}_i(t). \quad (1)$$

So far, this is just an uninteresting gas of non-interacting particles, each moving at some velocity. The dynamical rule added to make this model both deeply fascinating and technically non-trivial is to say that particles tend to align their direction of motion with nearby particles. Thus, at each time step we also update the orientation of particle i by

$$\hat{n}_i(t + \Delta t) = \mathcal{N} \left(\eta \vec{\xi}_i^t + \frac{1}{m_i^t} \sum_{\langle ij \rangle} \hat{n}_j(t) \right). \quad (2)$$

In the above equation, η is a parameter that characterizes the strength of the noise in the model, \vec{x}_i^t is a random unit vector delta-correlated in both time and particle index, the sum is over the m_i^t neighbors, j , that particle i has at time t , and \mathcal{N} is an operator that simply normalizes its argument. More simply: to determine a particles new direction of motion, we average the current direction of the particle's neighbors, add a bit of vectorial noise to this direction, and normalize the result so that \hat{n}_i stays a unit vector.

How, though, should we choose the set of neighbors that each particle attempts to align with at each time step? Early versions of the model proposed a straightforward and quite natural metric criterion: there is a distance scale associated with particle alignment, and so a parameter was added to the model. In a given timestep, all pairs of particles within some distance, r_0 , would count as neighbors, and particles farther apart would not. This solution feels quite natural, and it also has the virtue of mapping perfectly onto some of the efficient computational techniques we have learned for dealing with finite-ranged interactions. With this choice, there is a high-noise regime in which the particles are in a disordered state, a low-noise regime in which there is a phase transition to a polar flocking state, and an intermediate regime in which a “banded” phase is observed, characterized by both overall polar order and periodic traveling density bands [17].

While this kind of distance threshold for alignment feels natural, *is it how birds actually align with each other?* A beautiful body of literature has attempted to infer simple, phenomenological interaction rules for flocking birds, shoaling fish, and other collectively moving groups of organisms [18–20], and a consistent theme seems to be that these natural metric thresholds are likely poor descriptions of the “social forces” that represent organismal-scale interactions in moving-group dynamics. Some brief self-reflection of your own behavior in, say, moving crowds of different densities might supply the intuition that this should probably not be a surprise.

For flocks of birds, the StarFlag project reconstructed full three-dimensional positions and trajectories of flocks of thousands of birds flying above a railway station in Rome. Statistical analyses of these trajectories revealed what your intuition perhaps suggested: a typical bird interacts not with all birds up to some distance, but rather

with its six or seven *closest neighbors* – regardless of how far away those closest neighbors actually are [18, 21, 22].

We should be quite open, then, to interpreting the “sum over neighbors” in Eq. 2 without being restricted to the kinds of distance-based cutoffs we are so used to. What if, instead, we replaced the fixed-distance criteria with a rule that each particle interacted with its k nearest neighbors (with k a parameter)? Or the k nearest neighbors within some angular “field of view”? What about a rule that constructed some triangulation of the set of positions – say, a Delaunay triangulation – with neighbor relations chosen according to the set of edges in the triangulation? In all of these cases, the interactions are still “local,” but they are local in a topological rather than metric sense.

In pioneering work, Ginelli and Chaté showed that these changes *matter*, with implications (for instance) about the nature of the transition between the different dynamical phases of the model [10]. It is equally clear that the different topological criteria are *not equivalent to each other*. For instance, in a Delaunay triangulation the neighbor relation is still reciprocal – particles i and j share an edge, say, so each is a neighbor of the other. In contrast, k -nearest-neighbors need not be reciprocal; in the presence of density inhomogeneities particle i and be among the k nearest neighbors of particle j without j being among the k nearest neighbors of i . This opens up a universe of possibilities, with the effects of making these different choices for neighbor relations on the structural or dynamical phases of a model both interesting and poorly understood.

IV. COMPUTATIONAL TOOLS: VORONOI NEIGHBOR LISTS

In our exploration of topological neighbor relations, we focus on Delaunay-based neighbor lists in this chapter. They break away from strictly metric considerations, but they still maintain reciprocity of particle interactions; in this sense they are a natural first generalization to consider. So, what *is* a Delaunay triangulation of a given set of points? One definition is that for points in a two-dimensional plane, it is a triangulation that maximizes the minimum angle of all triangles in the triangulation (that is, the minimum angle in the Delaunay triangulation is at least as large as the minimum angle found in all possible other triangulations of the point set). Equivalently, the Delaunay triangulation can be defined as the triangulation with the *empty circumcircle* property: it is the triangulation such that no point in the point set is inside the circumcircle of any triangle in the triangulation (the circumcircle is simply the circle that passes through all of the vertices of a given polygon). Higher-dimensional versions of the Delaunay triangulation can be defined by the appropriate generalization of this to the empty circumsphere (or circum-hypersphere) property.

An example Delaunay triangulation, along with the

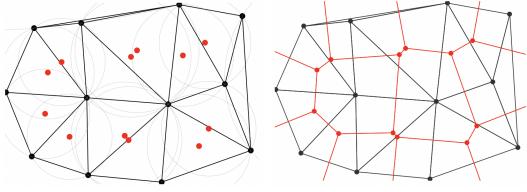


FIG. 2. (Left) A Delaunay triangulation of a set of points (in black). Circumcircles of corresponding to each triangle are shown, with circumcircle centers in red. (Right) The Voronoi tessellation is dual to the Delaunay triangulation. Connecting the circumcircle centers of triangles that share an edge produces the Voronoi diagram of the original point set.

(empty) circumcircles of each triangle), is shown in Fig. 2. The Delaunay triangulation is intimately related to the *Voronoi diagram* of a point set, which is a partition of a space into polygonal (or polyhedral, or appropriate higher-dimensional generalizations) regions that are closer to a given point than to any other point in the set. In fact, the Voronoi diagram and the Delaunay triangulation are dual to one another: the vertices of the Voronoi diagram correspond to the circumcenters of the Delaunay triangulation, and two triangles share an edge in the Delaunay triangulation then the vertices corresponding to those circumcenters will be connected by an edge in the Voronoi diagram. This relationship is similarly shown in Fig. 2.

The dual relation between Delaunay triangulations and Voronoi diagrams can often be exploited in computational settings. A wide variety of algorithms have been created to construct both types of graph, and if one has a Delaunay triangulation it is a trivial ($\mathcal{O}(N)$) operation to extract the Voronoi diagram (and vice versa). For instance, if one is simulating a Vicsek-style model where at each timestep every particle aligns with any particle in the first shell of its Voronoi neighbors [10], it might be computationally beneficial to avoid recomputing the entire Voronoi diagram if no change in the diagram occurred as a result of the most recent update of positions. It might be computationally convenient to formulate this condition – “do I need to recompute the Voronoi diagram?” – as a test of the empty circumcircle property of the Delaunay triangulation.

These computational and algorithmic considerations can be quite important. Most MD algorithms exploit the easily parallelizable character of particles evolving according to short-ranged interactions to accelerate simulations (across multiple CPU cores, or using GPUs or TPUs), but most efficient algorithms for computing the edges and faces of Delaunay triangulations and Voronoi diagrams are fundamentally serial ones. In some ways, this is natural: even though, e.g., the Voronoi diagram can be thought of as a basically local characterization of the area around each point in a set, it must still obey global consistency conditions (e.g., in 2D the graph of the Voronoi diagram must obey Euler’s formula for plane

graphs).

This makes it more difficult to build efficient MD simulations in systems where neighbor relations are determined by such a criteria. One option is to have a subroutine in an otherwise highly parallelized simulation that makes a call to an efficient algorithm to construct the Voronoi diagram (or Delaunay triangulation) of the entire set of particle positions in the simulation. Although scripting languages are not typically well-suited to large-scale simulations, existing packages can provide this functionality in Python, Matlab, Mathematica, and similar environments. On the high-performance side, a wrapper to quality open-source libraries that have been optimized for geometric calculations – like the Computational Geometry Algorithms Library (CGAL) [23, 24] and Voro++ – can be incorporated into your MD simulation software and called when needed. Recently, some work has gone into developing MD software packages that incorporate these sorts of tessellations at a fundamental level, with a high level of parallelism exploited even in the triangulation protocol [25]. To date, most of these efforts have centered around efficiently simulating specific models of dense tissue, which we discuss next.

V. GEOMETRIC MODELS OF DENSE TISSUE

A natural setting in which this sort of tessellation-based neighbor list arises is in modeling dense, confluent tissue – tissues in which there are no gaps between the cells composing the tissue. There is a rich history of looking at images of the sort seen in Fig. 1 and taking the idea of “cells as shapes” seriously, representing the confluent tissue by polygonal or polyhedral tilings of space [26–33]. These fundamentally geometrical models of cells have fascinating properties – in some cases they display the sort of behavior one might expect given roughly *any* reasonably-biologically-informed agent-based model in which each cell corresponds to just one or a handful of degrees of freedom, but in other cases they can be shown to support exotic mechanical states [32, 34, 35] and display unusual structural or dynamical scaling [36, 37]. To investigate these unusual properties, in recent years there have been substantial advances in the efficient numerical simulations of these and related agent-based models of dense tissue [25, 38–41].

To take a specific example, we consider using a computational package to generate the Voronoi diagram for a set of N points in the plane. In the context of the sort of Voronoi-based neighbor list discussed above, it is natural to assign each Voronoi “cell” to its generating point: the location of the points will be the degrees of freedom of the model, and each point will have a particular geometric region associated with it at every point in time. Here we are thinking of a “Voronoi model” in the context of dense tissue, but related models have been used to also study foams, geological formation processes, and the partitioning of animal territorial domains [42]. We proceed,

as promised, to take the tessellation seriously and consider an energy functional that depends explicitly on the geometry of the tessellation:

$$E = \sum_i^N (K_A(A_i - A_0)^2 + K_P(P_i - P_0)^2). \quad (3)$$

Here A_i and P_i are the area and perimeter of cell i , A_0 and P_0 are “preferred” values of the area and perimeter for cell i , and K_A and K_P are area and perimeter moduli. In the context of dense tissue, the quadratic dependence on cell area can be interpreted as a cell monolayer’s resistance to height fluctuations due to adhesions between cells and cell incompressibility, and the quadratic dependence on cell perimeter as a competition between active contractility of the actomyosin sub-cellular cortex and tension due to both cell-cell adhesions and cortical tension [36]. More generally, one might view this as subset of the possible low-order Taylor expansions that can be written in the various geometric properties of the tessellation. Indeed, in the absence of any area terms precisely this interpretation has been given, where the terms nonlinear in perimeter are what separate out foam-like energy functionals from tissue-like energy functionals [43].

Since we will be simulating this model on a computer, it is convenient to non-dimensionalize the energy by choosing the unit of length to be $\sqrt{\langle A \rangle}$, where $\langle A \rangle$ is the average area of all of the cells in our simulation, letting $k_r = K_A \langle A \rangle / K_P$, and writing

$$\frac{E}{K_P \langle A \rangle} \equiv e = \sum_i k_r (a_i - 1)^2 + (p_i - p_0)^2, \quad (4)$$

where a and p refer to dimensionless areas and perimeters. This non-dimensionalization differs from Eq. 3 only by an overall constant (assuming a finite domain that is completely occupied by cells), and it makes clear that the density-dependence of the model enters via $p_0 = P_0 / \sqrt{\langle A \rangle}$ – an important point given the early interest of this model as one in which the jamming transition was one controlled by preferred shape rather than strict density considerations [37, 44, 45].

Now that we have a set of degrees of freedom (again, here, the location of points in 2D space), a rule for finding neighbors, and an energy functional (which depends on that neighbor-finding rule), we have all of the ingredients necessary to perform agent-based simulations. One could perform energy minimizations to find tissue ground states, use classic molecular dynamics algorithms to simulate them in NVE or NVT ensemble, or choose from any of a number of equations of motion corresponding to non-equilibrium, active-matter scenarios (self-propelled particle dynamics, or self-propulsion with noise, or chemotaxis, or...) – the world is our oyster. All that is needed is an efficient way of taking the appropriate gradients of this energy functional in order to compute sets of forces acting on our degrees of freedom. For completeness, we show two variations of this calculation.

A. Vertex model forces

First, suppose we had a simpler version of our geometric model. Rather than taking a tessellation of a point set, suppose our degrees of freedom were actually the *vertices* of a set of polygons tiling the plane. This sort of model – often, unsurprisingly, referred to as a *vertex model* – has a similar flavor to Voronoi-based models, in that once again the neighbor list of our degrees of freedom comes *not* from strictly distance-based considerations, but rather stems from a graph structure (i.e., from an explicit enumeration of vertex-vertex connections). Given such an enumeration, computing the forces on the vertices given the energy functional in Eq. 3 is straightforward.

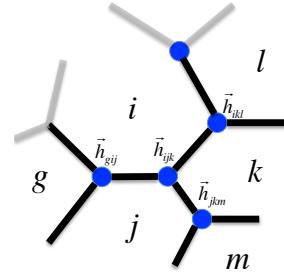


FIG. 3. Schematic diagram of cell i and some of its neighbors, along with the associated Voronoi vertices (the circumcenters of three adjacent cells) labeled for convenient reference.

For reference, Fig. 3 provides a schematic picture of a relevant patch of a two-dimensional tissue model, with each vertex labeled by the three cells it is adjacent to. From here, computing the forces is simply an exercise in patience / appropriate propagation of the chain rule. From the figure, since the motion of vertex \vec{h}_{ijk} only changes the shape of cells i , j , and k , the force on vertex \vec{h}_{ijk} is

$$-\frac{\partial E}{\partial \vec{h}_{ijk}} = - \left(\frac{\partial E_i}{\partial \vec{h}_{ijk}} + \frac{\partial E_j}{\partial \vec{h}_{ijk}} + \frac{\partial E_k}{\partial \vec{h}_{ijk}} \right). \quad (5)$$

Each of the cell-specific energy derivatives are straightforward, e.g.,

$$\frac{\partial E_i}{\partial \vec{h}_{ijk}} = 2K_A(A_i - A_0) \frac{\partial A_i}{\partial \vec{h}_{ijk}} + 2K_P(P_i - P_0) \frac{\partial P_i}{\partial \vec{h}_{ijk}}, \quad (6)$$

where the area- and perimeter-derivatives with respect to vertex positions can themselves be written as follows. Let $\vec{t}_{ij} = \vec{h}_{ijk} - \vec{h}_{gij}$, $\vec{t}_{ik} = \vec{h}_{ikl} - \vec{h}_{ijk}$, with \hat{t}_{ij} and \hat{t}_{ik} being the unit vectors in those directions. Similarly, let l_{ij} be the length of the edge between cell i and j , and \hat{n}_{ij} be the unit vector pointing outwardly normal to that cell

edge. Then one can write

$$\frac{\partial A_i}{\partial \vec{h}_{ijk}} = \frac{1}{2} (l_{ij} \hat{n}_{ij} + l_{ik} \hat{n}_{ik}), \quad (7)$$

$$\frac{\partial P_i}{\partial \vec{h}_{ijk}} = -(\hat{t}_{ij} + \hat{t}_{ik}). \quad (8)$$

B. Voronoi model forces

Returning to the class of models where the degrees of freedom are the positions of a point set we have tessellated, the same approach to computing forces can be used. Of course, the positions of the vertices of the Voronoi tessellation can be explicitly computed from the original point set, and so ultimately we are simply forced to carry out an extra layer of chain-rule calculations. Let's see this explicitly.

Following the notation in Bi et al. [36], the force on cell i in cartesian direction λ can be computed as

$$F_{i\lambda} = -\frac{\partial E}{\partial r_{i\lambda}} = -\frac{\partial E_i}{\partial r_{i\lambda}} - \sum_{<ij>} \frac{\partial E_j}{\partial r_{i\lambda}}, \quad (9)$$

the motion of cell i changes the geometry, and hence the energy, of cell i itself and all of its $\langle ij \rangle$ neighbors, j . Again using the configuration in Fig. 3, all of these terms can

be expanded via the chain rule, for instance:

$$\frac{\partial E_k}{\partial r_{i\lambda}} = \sum_{\nu} \left(\frac{\partial E_k}{\partial h_{ijk,\nu}} \frac{\partial h_{ijk,\nu}}{\partial r_{i\lambda}} + \frac{\partial E_k}{\partial h_{ikl,\nu}} \frac{\partial h_{ikl,\nu}}{\partial r_{i\lambda}} \right). \quad (10)$$

Here these are the only terms needed, since the other voronoi vertices associated with cell k (the middle of the three neighboring cells in clockwise order) do not depend on the position of cell i . The partial derivatives depend on the positions of \vec{h}_{jkm} and \vec{h}_{klm} , where m is the cell other than i that has both j and k as neighbors, and n is the cell other than i that has both k and l as neighbors (for this energy functional there is a dependence on the nearest and some of the next-nearest neighbors of cell i).

The derivative of the energy with respect to the vertices was calculated in the previous subsection, so the only additional piece that is needed are the derivatives of the Voronoi tessellation's vertices with respect to the position of the cell, e.g., $(\partial \vec{h}_{ijk}) / (\partial \vec{r}_i)$. These derivatives can be calculated efficiently as follows. We let \vec{r}_{ij} denote the vector from i to j , and define the auxiliary quantities

$$c = \vec{r}_{ij,x} \vec{r}_{kj,y} - \vec{r}_{ij,y} \vec{r}_{kj,x} \quad (11)$$

$$d = 2c^2 \quad (12)$$

$$\vec{z} = \beta d \vec{r}_{ij} + \gamma d \vec{r}_{ik} \quad (13)$$

$$\beta d = -|\vec{r}_{ik}|^2 \cdot (\vec{r}_{ij} \cdot \vec{r}_{jk}) \quad (14)$$

$$\gamma d = |\vec{r}_{ij}|^2 \cdot (\vec{r}_{ik} \cdot \vec{r}_{jk}). \quad (15)$$

with I_2 representing the 2×2 identity matrix and \otimes the dyadic product, the desired change in Voronoi vertex position with respect to cell position is

$$\frac{\partial \vec{h}_{ijk}}{\partial \vec{r}_i} = I_2 + \frac{1}{d} \left[\vec{r}_{ij} \otimes \left(\frac{\partial(\beta d)}{\partial \vec{r}_i} \right) + \vec{r}_{ik} \otimes \left(\frac{\partial(\gamma d)}{\partial \vec{r}_i} \right) - (\beta d + \gamma d) I_2 - \vec{z} \otimes \left(\frac{1}{d} \frac{\partial d}{\partial \vec{r}_i} \right) \right], \quad (16)$$

where

$$\frac{\partial(\beta d)}{\partial \vec{r}_i} = 2(\vec{r}_{ij} \cdot \vec{r}_{jk}) \vec{r}_{ik} + |\vec{r}_{ik}|^2 \vec{r}_{jk}, \quad (17)$$

$$\frac{\partial(\gamma d)}{\partial \vec{r}_i} = -2(\vec{r}_{ik} \cdot \vec{r}_{jk}) \vec{r}_{ij} + |\vec{r}_{ij}|^2 \vec{r}_{jk}, \quad (18)$$

and

$$\frac{1}{d} \frac{\partial d}{\partial \vec{r}_i} = \frac{2}{c} (-\vec{r}_{jk,y} \hat{x} + \vec{r}_{jk,x} \hat{y}). \quad (19)$$

VI. DISCUSSION

The point of the explicit calculation is not simply to recapitulate expressions that have appeared previously in the literature for the efficient calculation of vertex- and

Voronoi-model forces [25, 36], but to again emphasize the common character of these models with other agent-based simulations in the study of equilibrium and active matter. In this instance, the energy of a tissue configuration is dependent on the shape of the tessellated regions, and so in our computational work we need additional subroutines and data structures to calculate and keep track of these unusual sets of neighbor relations. Everything else, though, fits into the usual paradigm of a molecular dynamics simulation.

In the context of dense models of tissues, this perspective of starting with an unusual neighbor list and taking its implication seriously suggests a class of models with unusual behavior – in the way they can support stresses [32], in their “low-temperature” properties [46, 47], in the way they can organize into collectively moving states [48], in the way they support “curvotaxis” [49], and so on. In the context of active models with less complex energy

functionals, it is known that this “topological” way of choosing your neighbors changes the nature of the flock-

ing transition [10]. Beyond this, though: an expanse of open questions!

-
- [1] S. Plimpton, *Journal of computational physics* **117**, 1 (1995).
 - [2] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale, and K. Schulten, *Journal of computational chemistry* **26**, 1781 (2005).
 - [3] D. Van Der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, and H. J. Berendsen, *Journal of computational chemistry* **26**, 1701 (2005).
 - [4] J. A. Anderson, C. D. Lorenz, and A. Travesset, *Journal of computational physics* **227**, 5342 (2008).
 - [5] S. S. Schoenholz and E. D. Cubuk, arXiv preprint arXiv:1912.04232 (2019).
 - [6] D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications*, Vol. 1 (Elsevier, 2001).
 - [7] M. Rubinstein, R. H. Colby, *et al.*, *Polymer physics*, Vol. 23 (Oxford university press New York, 2003).
 - [8] M. P. Howard, A. Statt, F. Madutsa, T. M. Truskett, and A. Z. Panagiotopoulos, *Computational Materials Science* **164**, 139 (2019).
 - [9] J. Devany, D. M. Sussman, T. Yamamoto, M. L. Manning, and M. L. Gardel, *BioRxiv* , 804294 (2021).
 - [10] F. Ginelli and H. Chaté, *Physical Review Letters* **105**, 168103 (2010).
 - [11] D. M. Sussman, J. M. Schwarz, M. C. Marchetti, and M. L. Manning, *Physical review letters* **120**, 058001 (2018).
 - [12] Z. You, A. Baskaran, and M. C. Marchetti, arXiv preprint arXiv:2005.07684 (2020).
 - [13] L. P. Dadhichi, J. Kethapelli, R. Chajwa, S. Ramaswamy, and A. Maitra, *Physical Review E* **101**, 052601 (2020).
 - [14] A. Ivlev, J. Bartnick, M. Heinen, C.-R. Du, V. Nosenko, and H. Löwen, *Physical Review X* **5**, 011035 (2015).
 - [15] E. Lisin, O. Petrov, E. Sametov, O. Vaulina, K. Stassenko, M. Vasiliev, J. Carmona-Reyes, and T. Hyde, *Scientific Reports* **10**, 1 (2020).
 - [16] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, *Physical review letters* **75**, 1226 (1995).
 - [17] F. Ginelli, *The European Physical Journal Special Topics* **225**, 2099 (2016).
 - [18] A. Cavagna, A. Cimarelli, I. Giardina, G. Parisi, R. Santagati, F. Stefanini, and M. Viale, *Proceedings of the National Academy of Sciences* **107**, 11865 (2010).
 - [19] J. E. Herbert-Read, A. Perna, R. P. Mann, T. M. Schaerf, D. J. Sumpter, and A. J. Ward, *Proceedings of the National Academy of Sciences* **108**, 18726 (2011).
 - [20] J. Gautrais, F. Ginelli, R. Fournier, S. Blanco, M. Soria, H. Chaté, and G. Theraulaz, *PLoS Computational Biology* (2012).
 - [21] W. Bialek, A. Cavagna, I. Giardina, T. Mora, E. Silvestri, M. Viale, and A. M. Walczak, *Proceedings of the National Academy of Sciences* **109**, 4786 (2012).
 - [22] A. Attanasi, A. Cavagna, L. Del Castello, I. Giardina, T. S. Grigera, A. Jelić, S. Melillo, L. Parisi, O. Pohl, E. Shen, *et al.*, *Nature physics* **10**, 691 (2014).
 - [23] The CGAL Project, *CGAL User and Reference Manual*, 5.0 ed. (CGAL Editorial Board, 2019).
 - [24] S. Hert and S. Schirra, in *CGAL User and Reference Manual* (CGAL Editorial Board, 2019) 5.0 ed.
 - [25] D. M. Sussman, *Comput. Phys. Commun.* **219**, 400 (2017).
 - [26] H. Honda, *J. Theor. Biol.* **72**, 523 (1978).
 - [27] T. Nagai and H. Honda, *Philosophical Magazine Part B* **81**, 699 (2001).
 - [28] L. Hufnagel, A. A. Teleman, H. Rouault, S. M. Cohen, and B. I. Shraiman, *Proc. Natl. Acad. Sci. USA* **104**, 3835 (2007).
 - [29] R. Farhadifar, J.-C. Röper, B. Aigouy, S. Eaton, and F. Jülicher, *Curr. Biol.* **17**, 2095 (2007).
 - [30] L. Yan and D. Bi, *Physical Review X* **9**, 011029 (2019).
 - [31] S. Alt, P. Ganguly, and G. Salbreux, *Philosophical Transactions of the Royal Society B: Biological Sciences* **372**, 20150520 (2017).
 - [32] N. Noll, M. Mani, I. Heemskerk, S. J. Streichan, and B. I. Shraiman, *Nature Physics* **13**, 1221 (2017).
 - [33] M. Popovic, V. Druelle, N. Dye, F. Jülicher, and M. Wyart, *New Journal of Physics* (2020).
 - [34] M. Moshe, M. J. Bowick, and M. C. Marchetti, *Physical review letters* **120**, 268105 (2018).
 - [35] R. Kupferman, B. Maman, and M. Moshe, *Journal of the Mechanics and Physics of Solids* **143**, 104085 (2020).
 - [36] D. Bi, X. Yang, M. C. Marchetti, and M. L. Manning, *Physical Review X* **6**, 021011 (2016).
 - [37] E. Teomy, D. A. Kessler, and H. Levine, *Physical Review E* **98**, 042418 (2018).
 - [38] P. Van Liedekerke, M. Palm, N. Jagiella, and D. Drasdo, *Computational particle mechanics* **2**, 401 (2015).
 - [39] D. L. Barton, S. Henkes, C. J. Weijer, and R. Sknepnek, *PLoS Computational Biology* **13**, e1005569 (2017).
 - [40] A. G. Fletcher, M. Osterfield, R. E. Baker, and S. Y. Shvartsman, *Biophys. J.* **106**, 2291 (2014).
 - [41] A. Boromand, A. Signoriello, F. Ye, C. S. O’Hern, and M. D. Shattuck, *Physical review letters* **121**, 248003 (2018).
 - [42] D. Weaire and N. Rivier, *Contemporary Physics* **25**, 59 (1984).
 - [43] S. Kim, Y. Wang, and S. Hilgenfeldt, *Physical review letters* **120**, 248001 (2018).
 - [44] D. Bi, J. Lopez, J. Schwarz, and M. L. Manning, *Nature Physics* **11**, 1074 (2015).
 - [45] M. Merkel and M. L. Manning, *New Journal of Physics* **20**, 022002 (2018).
 - [46] D. M. Sussman, M. Paoluzzi, M. Cristina Marchetti, and M. Lisa Manning, *EPL (Europhysics Letters)* **121**, 36001 (2018).
 - [47] D. M. Sussman and M. Merkel, *Soft matter* **14**, 3397 (2018).
 - [48] F. Giavazzi, M. Paoluzzi, M. Macchi, D. Bi, G. Scita, M. L. Manning, R. Cerbino, and M. C. Marchetti, *Soft matter* **14**, 3471 (2018).
 - [49] D. M. Sussman, *Phys. Rev. Research* **2**, 023417 (2020).