

Robust self-assembly of nonconvex shapes in two dimensions

Lukas Mayrhofer^{1,*}, Myfanwy E. Evans^{2,†} and Gero Friesecke^{1,‡}

¹*Technische Universität München, Department of Mathematics, Boltzmannstraße 3, 85748 Garching, Germany*

²*University of Potsdam, Institute for Mathematics, Karl-Liebknecht-Strasse 24-25, 14476 Potsdam, Germany*



(Received 15 January 2024; accepted 20 June 2024; published 24 July 2024)

We present fast simulation methods for the self-assembly of complex shapes in two dimensions. The shapes are modeled via a general boundary curve and interact via a standard volume term promoting overlap and an interpenetration penalty. To efficiently realize the Gibbs measure on the space of possible configurations we employ the hybrid Monte Carlo algorithm together with a careful use of signed distance functions for energy evaluation. Motivated by the self-assembly of identical coat proteins of the tobacco mosaic virus which assemble into a helical shell, we design a nonconvex two-dimensional model shape and demonstrate its robust self-assembly into a unique final state. Our numerical experiments reveal certain essential prerequisites for this self-assembly process: blocking and matching (i.e., local repulsion and attraction) of different parts of the boundary, and nonconvexity and handedness of the shape.

DOI: [10.1103/PhysRevE.110.015309](https://doi.org/10.1103/PhysRevE.110.015309)

I. INTRODUCTION

Virus capsids are formed by robust self-assembly from copies of a small number of different coat proteins into a unique, often highly symmetric configuration [1]. Yet the coat proteins themselves have nonsymmetric, nonconvex shapes which are not intuitive. An example is the single coat protein of the tobacco mosaic virus (TMV), the copies of which self-assemble into a helical shell [2,3]. By contrast, the self-assembly of symmetric, convex shapes tends to lead to degenerate, nonunique configurations. Lennard-Jones clusters assembled from radially symmetric particles are prototypical [4]; a detailed analysis of the degeneracy for the simplified Heitmann-Radin potential in two dimensions is given in [5]. The self-assembly of copies of a small number of convex shapes has also been systematically studied [6].

Here we investigate the self-assembly of identical copies of two-dimensional (2D) shapes, and identify certain essential features which facilitate robust self-assembly into a unique configuration, among them nonconvexity.

The shapes are modeled via a general boundary curve and their interaction is described by an energy functional which contains a standard volume term promoting overlap and an interpenetration penalty:

$$E := V + \gamma P. \quad (1)$$

Here $\gamma > 0$ is a penalty strength. The penalty term P is nonuniform, allowing more or less overlap along the boundary. In physical terms, the volume term models depletion interactions, the penalty term models the impossibility of interpenetration, and the heterogeneity of the boundary captures

the effect that the hydrophobicity and the local electrostatic charge of proteins vary on the protein surface.

A common approach to simulate self-assembly is to decompose general shapes into simpler components which then interact via explicit pair potentials. For example, particle models use the Lennard-Jones pair potential [7] and the mean-field approach defines effective pair potentials for ellipsoids and polytopes taking into account the relative orientation [8].

In this paper, we instead represent the shapes using signed distance functions which are well established in the computer graphics community [9,10]. This type of representation is also used in the level-set method [11–13]. This approach allows one to model interactions between complex (nonconvex) shapes without decomposing them into artificial subcomponents.

The volume and penalty terms are defined via integrals over the ambient space, which are evaluated using quadrature over a grid. A key advantage of this approach is that the computational cost of computing the interaction energy scales linearly with respect to the number of shapes. By contrast, pair-potential based methods *a priori* scale quadratically, and the reduction of this scaling requires additional ingredients (neighborhood search).

For the evolution model, we use the hybrid Monte Carlo (HMC) algorithm [14,15] to compute approximate samples of the Gibbs measure:

$$\exp(-E(x)/T). \quad (2)$$

We design a nonconvex 2D model shape that achieves robust self-assembly into a unique structure (see Fig. 1). The shape is inspired by the three-dimensional (3D) shape of the TMV coat protein. Numerical results show that self-assembly is sensitive to small changes in the shape, the energy, and the evolution model. In particular, we demonstrate how making the shape convex or making the penalty term uniform along the boundary curve leads to misassemblies.

*Contact author: lukas.mayrhofer@tum.de

†Contact author: evans@uni-potsdam.de

‡Contact author: friesecke@tum.de

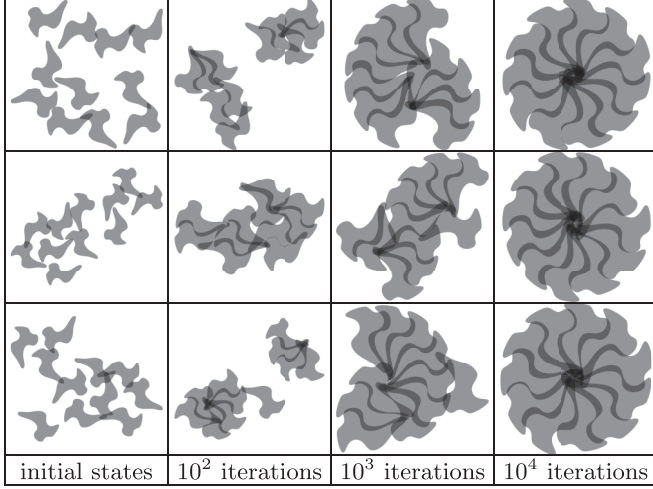


FIG. 1. Assembly progress of our model shape during three simulation runs. Each row corresponds to a run. One can see that the model shape robustly self-assembles into a unique rotationally symmetric structure.

II. ENERGY MODEL

The total volume of n given shapes $S_1, \dots, S_n \subset \mathbb{R}^2$ is

$$\lambda\left(\bigcup_{k=1}^n S_k\right) = \int_{\Omega} 1 - \prod_{k=1}^n (1 - \mathbf{1}_{S_k}(x)) d\lambda(x). \quad (3)$$

We replace the indicator function $\mathbf{1}_{S_k}$ with a smooth approximation $H[\alpha f_k(\cdot)]$, where f_k is the signed distance function

$$f_k(x) := \begin{cases} \text{dist}(x, \partial S_k) & \text{if } x \in S_k, \\ -\text{dist}(x, \partial S_k) & \text{if } x \notin S_k, \end{cases} \quad (4)$$

H is the smooth step function

$$H(t) := \begin{cases} 0 & \text{if } t < -1, \\ \frac{3}{16}t^5 - \frac{5}{8}t^3 + \frac{15}{16}t + \frac{1}{2} & \text{if } -1 \leq t < 1, \\ 1 & \text{if } 1 \leq t, \end{cases} \quad (5)$$

and $\alpha > 0$ is the transition steepness. The precise algebraic form of H is not important. Here, we use the minimal degree polynomial that makes H twice continuously differentiable. The volume term is given as

$$V := \int_{\Omega} 1 - \prod_{k=1}^n (1 - H(\alpha f_k(x))) d\lambda(x). \quad (6)$$

The overlap penalty term quantifies how much the shapes overlap. Its role is to enforce an interpenetration constraint between the shapes so that the shapes do not collapse into trivial volume-minimizing structures in which all shapes overlap at a single point.

The strength of the overlap penalty can vary along the boundary of the shapes. Some regions allow a lot of overlap, while other regions allow only minimal overlap. We call these regions *matching* and *blocking*, respectively. To model matching and blocking, we associate a non-negative penalty shift function $g_k : \mathbb{R}^2 \rightarrow \mathbb{R}$ to each shape S_k so that g_k is large on the *matching* part of S_k , allowing more overlap, and smaller on

the *blocking* part, allowing less overlap. The physical origin of matching and blocking is described in Sec. IV.

The overlap penalty term is given by

$$P := \int_{\Omega} \prod_{k=1}^n (1 + R(\beta[f_k(x) - g_k(x)])) d\lambda(x), \quad (7)$$

where R is the smooth ramp function

$$R(t) := \begin{cases} 0 & \text{if } t < -1, \\ -\frac{1}{16}t^4 + \frac{3}{8}t^2 + \frac{1}{2}t + \frac{3}{16} & \text{if } -1 \leq t < 1, \\ t & \text{if } 1 \leq t, \end{cases} \quad (8)$$

and $\beta > 0$ is the steepness of the ramp. Note that g_k offsets the signed distance function to control how soon the penalty becomes active.

The volume *interaction* term is the result after subtracting

$$\int_{\Omega} \sum_{k=1}^n H(\alpha f_k(x)) d\lambda(x) \quad (9)$$

from the volume term and the penalty *interaction* term is the result after subtracting

$$\int_{\Omega} 1 + \sum_{k=1}^n R(\beta[f_k(x) - g_k(x)]) d\lambda(x) \quad (10)$$

from the penalty term. The subtracted terms are invariant under rigid movements applied to the shapes.

For our actual simulation, we put the shapes into the flat torus $\Omega := \mathbb{R}^2/\eta\mathbb{Z}^2$ of size $\eta > 0$. The shapes S_1, \dots, S_n are rigidly translated and rotated copies of a reference shape S . Consequently, the signed distance functions f_k and penalty shift functions g_k are transformed versions of functions f and g associated to S . The signed distance function f is approximated using the procedure detailed in Appendix A. The integrals are evaluated using an equidistant quadrature grid with uniform weights (which corresponds in our context of periodic functions to the trapezoidal rule). For best performance, we set the grid spacing just fine enough such that the acceptance rate of our Metropolis-type evolution algorithm (introduced in Sec. III) is not too much affected. This ensures that the numerical errors stay of the same order of magnitude as the thermal fluctuations introduced by our evolution algorithm. In practice, this results in a grid spacing that just captures the important geometric details.

We visualize the energy terms for a configuration of 16 balls in Fig. 2. The penalty shift function is set to a positive constant for this visualization.

In summary, our energy model depends on the following parameters: (1) shape (S), (2) number of copies of the shape (n), (3) size of ambient space (η), (4) step transition and ramp steepness (α, β), (5) penalty shift function (g) and penalty strength (γ), and (6) quadrature grid spacing.

III. EVOLUTION MODEL

The n copies of the shape can freely translate and rotate. For each copy, the position and orientation are prescribed by a translation (an element of $\mathbb{R}^2/\eta\mathbb{Z}^2$) and a rotation [an element of $\text{SO}(2)$] about the shape center. This leads to the

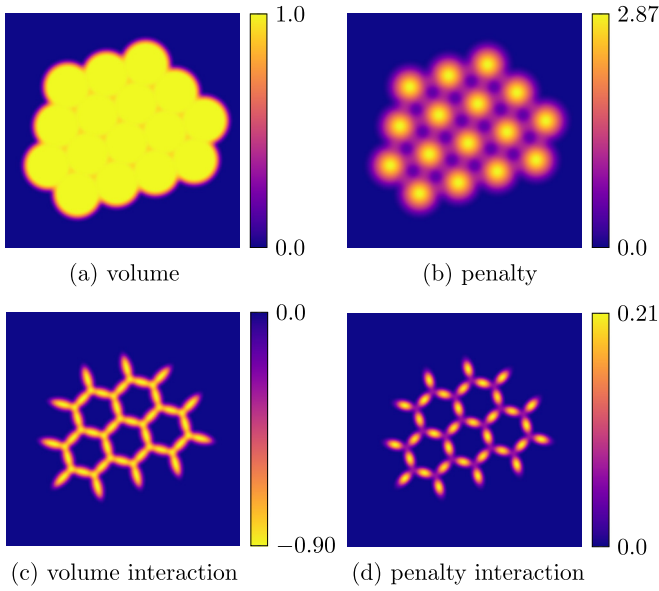


FIG. 2. Visualization of the energy integrands: (a) volume [Eq. (6)], (b) penalty [Eq. (7)], (c) volume interaction [Eq. (6) minus Eq. (9)], and (d) penalty interaction [Eq. (7) minus Eq. (10)]. For the volume interaction, the integrand of which is negative, bright regions correspond to low values.

configuration space $G := [\mathbb{R}^2/\eta\mathbb{Z}^2 \times \text{SO}(2)]^n$, which is a compact Lie group.

To drive the simulation, we first tried a standard Brownian dynamics method and the Metropolis-Hastings algorithm with a random-walk proposal. However, we found that the HMC algorithm significantly outperforms these methods when the dynamics become constrained, as in our case where shapes attach to each other and then move collectively. The HMC algorithm computes samples from the Gibbs measure on G given by

$$\mu(A) := c \int_A \exp(-E(x)/T) d\lambda_G(x), \quad (11)$$

where c is the normalization constant, $T > 0$ is a simulation temperature parameter, and λ_G is the Haar measure on G .

An iteration of HMC consists of two steps: First, we compute a trajectory of the artificial Hamiltonian

$$H(x, p) := E(x)/T + \frac{1}{2}\langle p, p \rangle_{\mathfrak{g}} \quad (12)$$

with position x in G and velocity p in the Lie algebra \mathfrak{g} corresponding to G . This is done by integrating

$$x'(t) = p(t), \quad (13)$$

$$p'(t) = (-1/T)\nabla E(x(t)) \quad (14)$$

on the interval $[0, L]$ using the time-reversible and volume-preserving Leapfrog algorithm. Here, $\langle \cdot, \cdot \rangle_{\mathfrak{g}}$ is a fixed inner product on \mathfrak{g} and $L > 0$ is a fixed trajectory length. The initial value $x(0)$ is set to the current configuration of shapes and the initial velocity $p(0)$ is drawn from a Gaussian distribution with density proportional to

$$\exp\left(-\frac{1}{2}\langle p, p \rangle_{\mathfrak{g}}\right). \quad (15)$$

We employ the exponential map to adapt the Leapfrog algorithm to our Lie group setting.

The generated proposal $x(L)$ is accepted with probability

$$\min\{\exp(H(x(0), p(0)) - H(x(L), p(L))), 1\}. \quad (16)$$

If it is accepted, HMC sets the current state to $x(L)$ and generates a new proposal from this state. Otherwise, a new proposal is generated from $x(0)$.

It can be shown that the Markov chain generated by this iteration leaves the Gibbs measure invariant. Invariance is further discussed in Appendix B.

For the initial shape configuration, we draw a sample from the Haar measure on G , which means choosing initial translations uniformly in $[0, \eta]^2$ and rotations uniformly in $\text{SO}(2)$. To ensure that the shapes do not overlap before the actual simulation starts, we preprocess this sample using the Metropolis-Hastings algorithm with a simple repulsive energy model.

For our simulations, we select the inner product

$$\langle p, q \rangle_{\mathfrak{g}} := \sum_{i=1}^n \sigma_T^{-2} \langle p_i^T, q_i^T \rangle + \sigma_R^{-2} p_i^R q_i^R, \quad (17)$$

where p_i^T and q_i^T are the translation components and p_i^R and q_i^R are the rotation components. This means that the initial velocities for HMC are drawn from a normal distribution with a diagonal covariance matrix. The parameters σ_T and σ_R define the standard deviations of the translation and rotation components, respectively.

IV. NUMERICAL RESULTS

Our numerical experiments were carried out in the JULIA programming language [16].

A. Balls assembling into nonunique states

Simulations with simple convex shapes such as balls do not settle into a unique state (see Fig. 3). We trace this high degeneracy of the assembled state to the extreme symmetry of such model shapes.

B. A nonconvex model shape

Some intuition about relevant features leading to unique assembled states can be gained from biological systems. An example, the coat protein of the tobacco mosaic virus, is shown in Fig. 4; it was designed by evolution to robustly assemble into a helix.

By combining this intuition with numerical tests we designed a 2D model shape (see Fig. 5) which achieves robust assembly into a unique state. This shape captures some important features of the TMV protein: it is highly asymmetric, nonconvex, and handed. The detailed form of our shape is not important, but—as we demonstrate below—these features are. A precise description of the model shape is given in Appendix C. The main simplification is that our model shape forms a 2D ring, whereas the TMV protein forms a helix, due to the fact that neighboring proteins are also mildly offset in the third dimension (see Fig. 4).

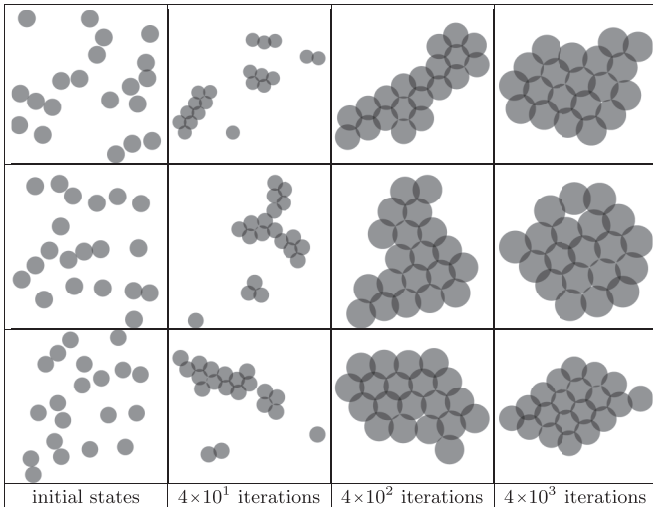


FIG. 3. Assembly progress of balls during three simulation runs. Each row corresponds to a run. One can see that the balls assemble into different configurations.

Another important aspect about protein assembly into unique states is the heterogeneity of the proteins: the hydrophobicity and the electrostatic charge vary on the protein surface, leading to local repulsion and attraction. For a detailed computational model of the resulting heterogeneous solvation free energy see [17]. As a simple model interaction that captures this effect, we divide the shape into a *matching* and a *blocking* part (indicated in Fig. 5) which allow more and less overlap, respectively, with other shapes as explained in detail in Sec. II.

C. Numerical results for the model shape

We conduct assembly experiments with the simulation parameters in Table II. Using these parameters, we observe robust self-assembly into a unique configuration

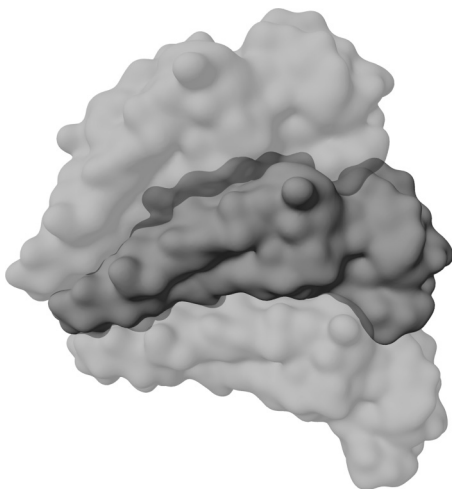


FIG. 4. Coat protein of the tobacco mosaic virus [2,3] (dark gray), and two neighbors in the native helical assembly (light gray).



FIG. 5. Model protein shape. Like the 3D shape in Fig. 4 it is asymmetric, nonconvex, and handed. The shape is divided into a light *matching* part and dark *blocking* part.

(see Fig. 1).

To quantify the self-assembly progress, we consider the graph of correctly attached shapes. In this assembly graph, the vertices represent the shapes and the edges represent connections between shapes that are (within a small tolerance in relative translation and rotation) correctly attached to each other. We call a subset of shapes a *correctly assembled component* if it is a connected component of this graph. The number of correctly assembled components is a useful measure of self-assembly progress. Furthermore, we call a configuration correctly assembled if it consists of a single correctly assembled component.

In Fig. 6 we show the distribution of the number of correctly assembled components as a function of HMC iterations. Initially, there are typically 11 correctly assembled components, which gradually reduce to one component. At 10^4 iterations, most runs are in a correctly assembled configuration.

1. Parameter identification

To achieve rapid and robust self-assembly, we need the following.

- (1) *Convergence.* The evolution algorithm should converge quickly to the Gibbs measure.
- (2) *The right Gibbs measure.* Samples drawn from the Gibbs measure should be in the assembled state with high probability.

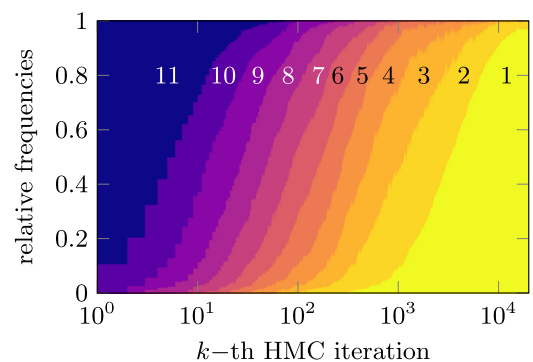


FIG. 6. Assembly progress from 11 correctly assembled components (dark, disassembled configuration) to one component (bright, fully assembled configuration). The distribution over 512 hybrid Monte Carlo runs is shown.

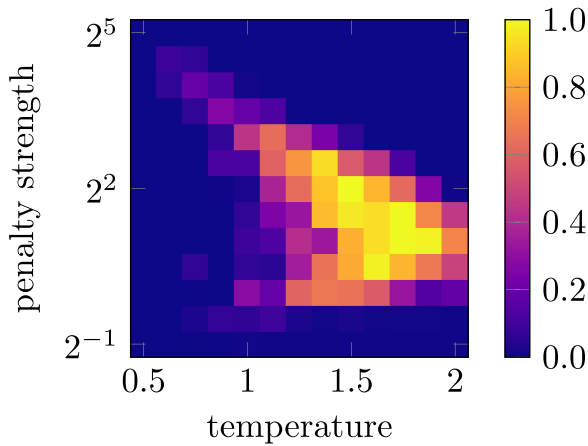


FIG. 7. Self-assembly robustness, showing the proportion of configurations that are correctly assembled. The proportions are computed from iterations 9×10^3 to 10^4 of 16 hybrid Monte Carlo runs for each choice of parameters.

We investigate how different choices of simulation temperature and penalty strength affect the robustness of self-assembly. For each choice, we tune the Leapfrog step size to achieve an acceptance rate of about 65%. The results are shown in Fig. 7.

It turns out that there is a sweet spot for both parameters: Low temperatures give more weight to low energy states but slow down the convergence to the Gibbs measure. High temperatures converge faster but make the interesting configurations less likely to occur. Low penalty strengths give the evolution algorithm more leeway to perturb shape configurations but increase shape overlap and the probability of the shapes collapsing. High penalty strengths make the configurations unstable and likely to disassociate.

2. Ablation study

We investigate how the following changes to the shape influence self-assembly robustness: (1) removing the head repulsion (Fig. 8), (2) removing the curvature (Fig. 9), and (3) removing the head (Fig. 10).

For each case, we consider a smooth transition from the original shape to the modified version and carry out multiple HMC runs for various points during the transition. We see

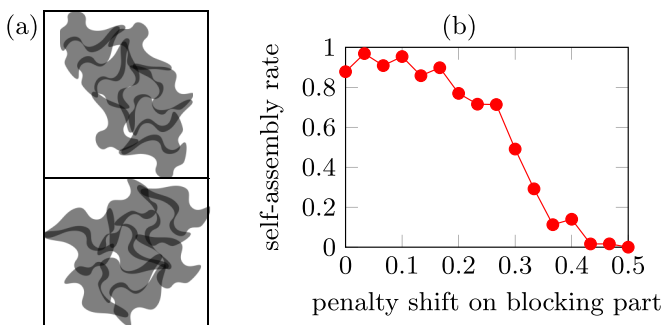


FIG. 8. Removing the head repulsion. We change the value of the penalty shift function on the blocking part. The value 0.5 equals that on the matching part. (a) Failure modes with 0.5. (b) Assembly rate.

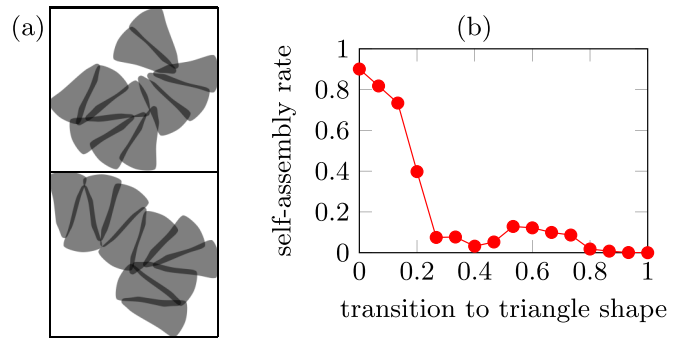


FIG. 9. Removing the curvature. We transition from the default shape (0) to a triangle shape (1). (a) Failure modes with the triangle shape. (b) Assembly rate.

in the figures that the self-assembly robustness decays during this transition.

When self-assembly fails, the question is whether this is a failure of the energy function (the energy minimizer no longer corresponds to the assembled state) or a failure of the evolution model (HMC gets stuck in local minima). In the three cases that we investigate here, the energy minimizer still seems to be the fully assembled configuration. However, the energy difference between the minimizer and the misassembled configurations that HMC gets stuck in is small and the evolution algorithm is unable to find a path to the global minimizer.

3. Ground states and symmetry breaking

We use gradient descent to find stationary points (or ground states) starting from correctly assembly configurations.

The shape we used up to now leads to a perfectly symmetric ground state. This state, together with the eigenvalues of its Hessian, is shown in Fig. 11. The three eigenvalues near zero correspond to the zero modes of translating and rotating the whole configuration. These eigenvalues are not exactly zero since the energy is not perfectly invariant under these operations due to the use of numerical quadrature on a finite grid. Since the rest of the eigenvalues are positive, we conclude that the ground state is a local energy minimizer. In Fig. 12 we show the energy integrands for the ground state.

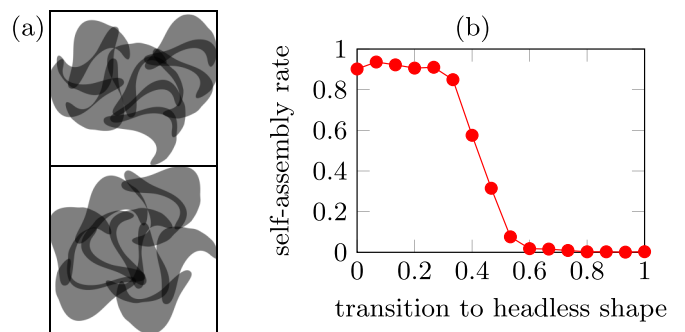


FIG. 10. Removing the head. We transition from the default shape (0) to a headless shape (1). (a) Failure modes with the headless shape. (b) Assembly rate.

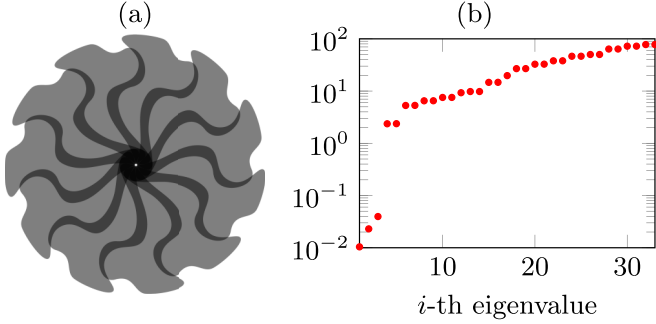


FIG. 11. (a) Ground state of the default shape. (b) Eigenvalues of the ground state's Hessian; see text for detailed interpretation.

The symmetry of the ground state is sensitive to small modifications of the shape. To explore this, we rescale our shape along an axis to make it slightly thinner (down to a factor of 0.95). Thinning the shape speeds up the self-assembly while also causing symmetry breaking for the ground states (see Fig. 13). If the shape is thinned even more, an additional 12th shape would be required for complete assembly.

V. DISCUSSION

We have given a proof of concept that demonstrates the potential of our coarse-grained energy and evolution models for efficiently simulating large-scale self-assembly processes of proteins. Our simulations of nonconvex model shapes indicate the significance of two factors for the robust self-assembly into a unique structure: blocking and matching (i.e., local repulsion and attraction) of different parts of the boundary, and nonconvexity and handedness of the shape.

Our paper can be extended in various ways. The obvious one is simulating self-assembly in three dimensions, which we are aiming for with future work. To simulate protein

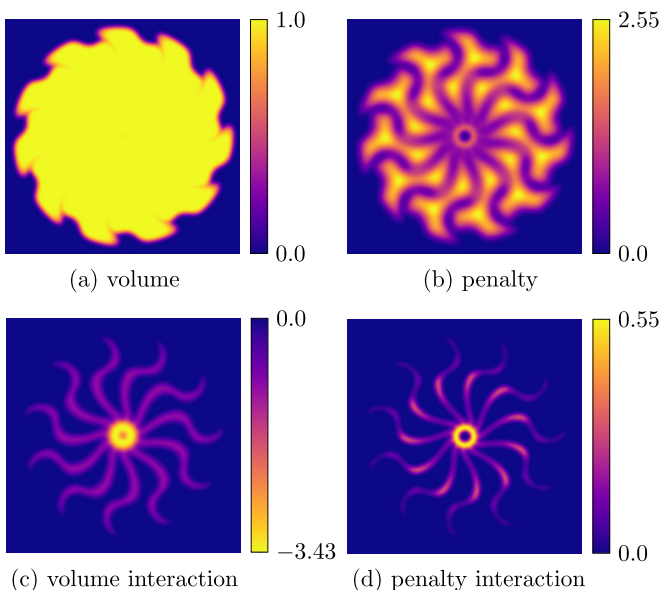


FIG. 12. Visualization of the energy integrands. It is similar to Fig. 2, except for the ground state shown in Fig. 11.

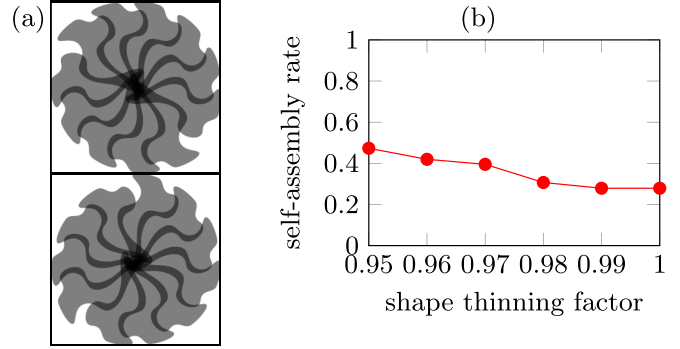


FIG. 13. (a) A slightly thinner shape self-assembles faster but leads to ground states with broken symmetry. (b) Assembly rate computed from HMC iterations 1.5×10^3 to 2×10^3 .

self-assembly in three dimensions, the solvent-accessible surface [18] would be the natural choice for the shape that we would use for our simulations.

Our energy model can be refined in various ways. For instance underlying mechanisms of matching and blocking, such as local hydrophobicity and local electrostatic charges, could be modeled explicitly. Alternatively, one could use semiempirical refinements of the volume term such as the morphometric approach for the free solvation energy [19–22]

$$E = pV + \sigma A + \kappa C + \tilde{\kappa} X, \quad (18)$$

where the terms are volume (V), surface area (A), mean width (C), and Euler characteristic (X).

ACKNOWLEDGMENTS

The research of L.M. was funded by the Deutsche Forschungsgemeinschaft via Project No. 195170736-TRR109. We thank I. Spirandelli for helpful discussions on the modeling and F. Bornemann for suggesting the HMC algorithm.

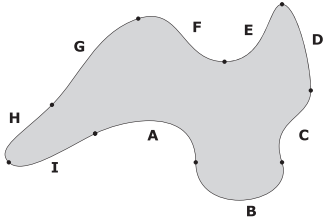
APPENDIX A: APPROXIMATING SIGNED DISTANCE FUNCTIONS

To approximate the signed distance function (SDF) of a shape $S \subset \mathbb{R}^2$, we first sample the exact SDF on a fine grid in a sufficiently large box containing S . We reduce aliasing by convolving the result with a Gaussian kernel before subsampling the values on a coarse grid.

From the data on the coarse grid, we construct a smooth SDF approximation using cubic B-spline functions. In this construction, we do not interpolate. Instead, we directly take the subsampled values as coefficients for the tensorized B-spline functions.

The SDF approximation procedure adds the following parameters to our energy model: (1) SDF fine and coarse grid spacing (for sampling the exact signed distance function, respectively, for B-spline coefficients of the approximate signed distance function) and (2) SDF Gaussian standard deviation (standard deviation of the Gaussian kernel used to low-pass filter the samples on the fine grid).

TABLE I. Description of the model shape.



	A	B	C	D	E	F	G	H	I
β_1^0	-2	5	11	13	11	7	1	-5	-8
β_2^0	-3	-5	-5	0	6	2	5	-1	-5
β_1^1	0	5	10	13	10	4	-2	-7	-7
β_2^1	-2	-9	-2	2	6	2	4	-3	-6
β_1^2	5	12	13	12	10	4	-3	-9	-4
β_2^2	-1	-8	-2	6	2	6	1	-4	-4
β_1^3	5	11	13	11	7	1	-5	-8	-2
β_2^3	-5	-5	0	6	2	5	-1	-5	-3

TABLE II. Simulation parameters.

Parameter	Value
Number of copies of the shape	11
Size of ambient space (η)	64
Step transition and ramp steepness	(1, 1)
Penalty shift value on matching part	0.5
Penalty shift value on blocking part	0.1
Penalty strength	4
Quadrature grid spacing	1
SDF fine grid spacing	0.1
SDF coarse grid spacing	1
SDF Gaussian standard deviation	0.5
Temperature	1.5
Inner product parameters (σ_T, σ_R)	(1, 0.21)
Leapfrog step size	0.16
Leapfrog iterations	15

APPENDIX B: INVARIANCE OF THE GIBBS MEASURE

To show that the Gibbs measure μ is invariant under an HMC iteration, we check that the transition kernel T of HMC satisfies detailed balance with respect to μ , that is,

$$\int_B T(x, A) d\mu(x) = \int_A T(x, B) d\mu(x) \quad (B1)$$

for all measurable sets A and B . This follows from a straightforward calculation that is not specific to our Lie group setting. An important ingredient in this calculation is that the Leapfrog algorithm is time reversible and volume preserving.

To show these properties, we recall the Leapfrog algorithm.

```

p ← p -  $\frac{\epsilon}{2T}$  ∇E(x)
x ← expG( $\epsilon p$ )x
repeat L - 1 times
  p ← p -  $\frac{\epsilon}{T}$  ∇E(x)
  x ← expG( $\epsilon p$ )x
end
p ← p -  $\frac{\epsilon}{2T}$  ∇E(x)
    
```

Time reversibility follows since

$$\exp_G(-p) = \exp_G(p)^{-1}. \quad (B2)$$

To show that Leapfrog is volume preserving, we note that it is a composition of maps of the form $(x, p) \mapsto (f(p)x, p)$ and

$(x, p) \mapsto (x, p + g(x))$. The result then follows by applying Fubini's theorem.

APPENDIX C: DESCRIPTION OF MODEL PROTEIN SHAPE

Our 2D model shape is defined as the region enclosed by a curve that consists of cubic Beziér segments. Each segment $c : [0, 1] \rightarrow \mathbb{R}^2$ is given by

$$c(t) := (1 - t)^3 \beta^0 + 3(1 - t)^2 t \beta^1 + 3(1 - t) t^2 \beta^2 + t^3 \beta^3, \quad (C1)$$

where $\beta^0, \beta^1, \beta^2$, and $\beta^3 \in \mathbb{R}^2$ are the *control points*. Table I lists the control points for each segment. The segments are chosen such that the composite curve has a continuous derivative.

We divide the shape along the line $\{(9.7, t) : t \in \mathbb{R}\}$ into a matching part (left) and a blocking part (right). The penalty shift function is set to 0.5 on the matching part and to 0.1 on the blocking part with a transition between these values via the smooth step function given in Eq. (5). For our simulations, we center the shape using a shift by $(-3.7, 0.7)$.

APPENDIX D: SIMULATION PARAMETERS

Table II shows the simulation parameters.

[1] D. L. D. Caspar and A. Klug, Physical principles in the construction of regular viruses, *Cold Spring Harb. Symp. Quant. Biol.* **27**, 1 (1962).

[2] C. Schmidli, S. Albiez, L. Rima, R. Righetto, I. Mohammed, P. Oliva, L. Kovacic, H. Stahlberg, and T. Braun, Tobacco mosaic virus (TMV) (2019), PDB entry 6r7m, <https://www.rcsb.org/structure/6r7m>.

[3] C. Schmidli, S. Albiez, L. Rima, R. Righetto, I. Mohammed, P. Oliva, L. Kovacic, H. Stahlberg, and T. Braun, Microfluidic protein isolation and sample preparation for high-resolution cryo-EM, *Proc. Natl. Acad. Sci. USA* **116**, 15007 (2019).

[4] D. J. Wales and J. P. K. Doye, Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones

- clusters containing up to 110 atoms, *J. Phys. Chem. A* **101**, 5111 (1997).
- [5] L. De Luca and G. Friesecke, Classification of particle numbers with unique Heitmann-Radin minimizer, *J. Stat. Phys.* **167**, 1586 (2017).
- [6] J. A. Millan, D. Ortiz, G. Van Anders, and S. C. Glotzer, Self-assembly of Archimedean tilings with enthalpically and entropically patchy polygons, *ACS Nano* **8**, 2918 (2014).
- [7] N. Pakalidou, J. Mu, A. J. Masters, and C. Avendaño, Engineering porous two-dimensional lattices *via* self-assembly of non-convex hexagonal platelets, *Mol. Syst. Des. Eng.* **5**, 376 (2020).
- [8] V. Ramasubramani, T. Vo, J. A. Anderson, and S. C. Glotzer, A mean-field approach to simulating anisotropic particles, *J. Chem. Phys.* **153**, 084106 (2020).
- [9] K. Perlin and E. M. Hoffert, Hypertexture, in *Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques* (ACM, New York, 1989), pp. 253–262.
- [10] J. C. Hart, Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces, *The Visual Computer* **12**, 527 (1996).
- [11] A. Dervieux and F. Thomasset, A finite element method for the simulation of a Rayleigh-Taylor instability, in *Approximation Methods for Navier-Stokes Problems*, Lecture Notes in Mathematics, edited by R. Rautmann (Springer-Verlag, Berlin, 1980), Vol. 771, pp. 145–158.
- [12] S. Osher and J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.* **79**, 12 (1988).
- [13] S. Osher and R. Fedkiw, in *Level Set Methods and Dynamic Implicit Surfaces*, edited by S. S. Antman, J. E. Marsden, and L. Sirovich, Applied Mathematical Sciences (Springer, New York, 2003), Vol. 153.
- [14] S. Duane, A. Kennedy, B. J. Pendleton, and D. Roweth, Hybrid Monte Carlo, *Phys. Lett. B* **195**, 216 (1987).
- [15] M. Betancourt, A conceptual introduction to Hamiltonian Monte Carlo, [arXiv:1701.02434](https://arxiv.org/abs/1701.02434).
- [16] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, Julia: A fresh approach to numerical computing, *SIAM Rev.* **59**, 65 (2017).
- [17] D. G. Thomas, J. Chun, Z. Chen, G. Wei, and N. A. Baker, Parameterization of a geometric flow implicit solvation model, *J. Comput. Chem.* **34**, 687 (2013).
- [18] M. L. Connolly, Solvent-accessible surfaces of proteins and nucleic acids, *Science* **221**, 709 (1983).
- [19] H. Hadwiger, Integralsätze im Konvexring, *Abh. Math. Semin. Univ. Hambg.* **20**, 136 (1956).
- [20] H. Hansen-Goos, R. Roth, K. Mecke, and S. Dietrich, Solvation of proteins: Linking thermodynamics to geometry, *Phys. Rev. Lett.* **99**, 128101 (2007).
- [21] H. Hansen-Goos and R. Roth, Density functional theory for hard-sphere mixtures: The white bear version mark II, *J. Phys.: Condens. Matter* **18**, 8413 (2006).
- [22] I. Spirandelli, R. Coles, G. Friesecke, and M. E. Evans, Exotic self-assembly of hard spheres in a morphometric solvent, *Proc. Natl. Acad. Sci. USA* **121**, e2314959121 (2024).