Supporting Information: Block copolymers beneath the surface: measuring and modeling complex morphology at the sub-domain scale

Abhiram Reddy, Xueyan Feng, Edwin Thomas,* and Gregory Grason*

E-mail: elt@exchange.tamu.edu; grason@mail.pse.umass.edu

Introduction

In this document, we briefly describe the algorithms used for medial set construction which are implemented in the accompanying software package and used for medial analysis of sub-domain geometry that resulted in main text Figs.7, 9 and 20. For any domain in Euclidean space $\mathbb{E}^3$ enclosed by domain boundary S, the medial set MS is defined as collection of points $m_{\pm}(p)$ that are equidistant to two or more points p_i on the boundary S.[1,2] Following Amenta *et al.*[1] and Schröder *et al.*[2] both the algorithms we describe here are based on Voronoi tesellations on point cloud obtained from generating a discrete mesh of domain boundary S.

In the context of medial analysis for sub-domain thickness of self-assembled block copolymer morphologies, the domain boundary S here represents the inter-material dividing surface(s) (IMDS) between unlike blocks. A standard way of characterizing these morphologies is by their respective composition fields ϕ_α where $\alpha = A/B$ for linear AB diblock copolymers. We define the IMDS for any morphology as the level set of composition fields for fixed value i.e $\phi(\mathbf{r})_{A/B} = 0.5$. In section II, we describe in detail the workflow presented in Fig.1

to obtain required data structures that are provided as an input to algorithms to construct medial set MS. Pseudo-codes for these algorithms are presented in section III.

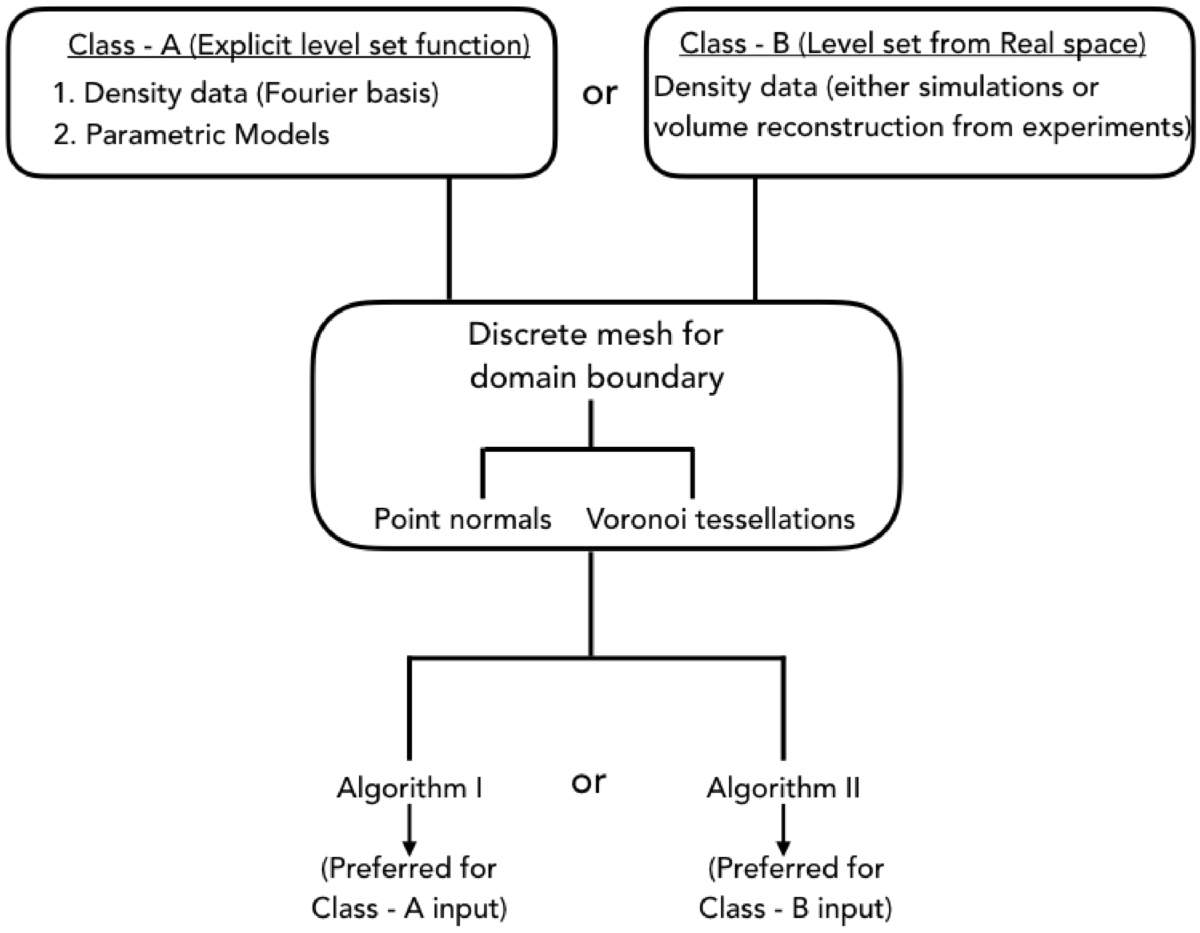
Class - A (Explicit level set function)
1. Density data (Fourier basis)
2. Parametric Models
or
Class - B (Level set from Real space)
Density data (either simulations or volume reconstruction from experiments)
Discrete mesh for domain boundary
Point normals
Voronoi tessellations
Algorithm I
or
Algorithm II
(Preferred for Class - A input)
(Preferred for Class - B input)

Figure S1: Flow-chart describing the steps involved in creating prerequisite data structures for algorithms I and II to compute medial set

Prerequisites

Domain boundary (IMDS)

The composition fields for individual blocks of BCPs are represented in real space and/or fourier space depending on the tools used to obtain them i.e theoretical models (SCFT) or coarse-grained simulations or volume reconstruction from experimental data. Explicit level surfaces are also commonly used as an approximation to IMDS in triply periodic network morphologies[3]

Nodal Approximations(class A)

For analysis presented in Fig. 7 of Section II in main paper, we have used nodal approximation of G surface given by

$$f(x,y,z) = \sin\left[\frac{2\pi x}{D}\right]\cos\left[\frac{2\pi y}{D}\right] + \cos\left[\frac{2\pi x}{D}\right]\sin\left[\frac{2\pi z}{D}\right] + \sin\left[\frac{2\pi y}{D}\right]\cos\left[\frac{2\pi z}{D}\right] \tag{S1}$$

where D is the length of the unit cell. We chose a level surface for $f(x,y,z) = \pm 1.07$ as a bounding surfaces that are similar in topology as tubular sub-domains of double gyroid networks and occupies 30% volume of the total unit cell.

Density data from SCFT(class A and B)

For medial surfaces of block copolymer Frank-Kasper sphere crystals presented in Fig 9, we have used output from Polymer Self-Consistent Field (PSCF).[5] Within PSCF, scalar density $\phi_\alpha(\mathbf{r})$ and chemical potential $\omega_\alpha(\mathbf{r})$ fields are represented using a *symmetry adapted basis functions* as it helps with faster computation of self-consistent iterative computations using the psedo-spectral method for solving modified diffusion type equation.[5] We can build density function $\phi_\alpha(\mathbf{r})$ as

$$\phi_\alpha(\mathbf{r}) = \sum_{i=1}^{N_{star}} \phi_{i,\alpha} f_i(\mathbf{r}) \tag{S2}$$

where N_{star} is the number of basis functions needed to approximate the field and $\phi_{i,\alpha}$ is the co-efficient for each basis function $f_i(\mathbf{r})$. Alternatively, PSCF can also convert and output these composition fields $\phi_\alpha(\mathbf{r})$ in real space as a scalar value for each of the voxels within the unit cells.

Density data from experiments(class B)

For the composition fields used for medial surfaces within sub-domains of PS-PDMS double gyroid presented in Fig. 20 of Section III of main text, we processed raw greyscale SVSEM images of PS-PDMS sample using 3D FFT to apply Bragg-filtering and then used inverse FFT to get grey scale images. A 3D volume reconstruction was then done by stacking 2D Bragg-filtered SEM images. We later used the SEM intensity of each voxel in real space (high intensity regions for PDMS and low intensity regions for PS) along with constraints on volume fraction of PS vs. PDMS in the sample to compute the IMDS separating PS-PDMS domains. We refer the reader to ref[6] for detailed description of our procedure and also to ref[7] to access source data repository for software used to do this.

Discrete mesh

A discrete mesh of IMDS is obtained by using inbuilt *Mathematica* functions

```
R = ImplicitRegion[f[x,y,z] == t, {x,y,z}]
Rd = DiscretizeRegion[R,{{xmin,xmax},{ymin,ymax},{zmin,zmax}},
         MaxCellMeasure -> {"Area" -> 0.001}]
```

$f(x, y, z)$ can either be explicit level set function as in eq. S1 or as a Fourier representation of density field $\phi_{A/B}(\mathbf{r})$ in eq. S2. While working with real space composition field data this function is obtained using inbuilt periodic interpolation function in *Mathematica.* Using this approach domain boundary (IMDS) is represented as a triangulated mesh and we can export the list of points and connectivity of these points which forms the discrete mesh. $x_{min}, x_{max}, ...$ sets the minimum and maximum of the domain boundary in three dimensions while we can change value of *Area* in $MaxCellMeasure$ and refine the coarseness of discrete mesh. Here, we define coarseness as a ratio, $\max(A_F)/(A_T)$ where A_T is the total area of the domain boundary surface and A_F is the area of a face on the discrete mesh.

Point normals

For explicit level set functions $f(x, y, z)$ in class-A type of domain boundary surfaces, we compute normal data for each of the points by taking a gradient of $f(x, y, z)$ at every point p_i

$$\mathbf{n} = \nabla f(x, y, z) \tag{S3}$$

and for other types i.e class-B where we essentially create a function for $f(x, y, z)$ based on interpolation of scalar density values $\phi(\mathbf{r})$,we compute face normals to each face of triangulated mesh and then for point normal we average over all the faces that contain a particular point on the discrete mesh.

Voronoi cells

Independent of source of the boundary surface S and how it is processed to obtain a point cloud i.e discrete mesh, a significant step in this workflow involves generating Voronoi cells and storing geometric data of these cells. $C_V(p_i)$ denotes a Voronoi cell for a point p_i where the cell has v_j vertices and f_k faces which connect these vertices. $\mathbf{n}_{f_k}$ denotes the components of a normal vector to each face f_k that belongs to a particular cell $C_V(p_i)$. We have used *Voro++*[4] to compute Voronoi tesellation of points cloud and face normals of each Voronoi cell. The following commands were passed onto *Voro++* to generate input files C-E of algorithm I i.e Voronoi tesellation of points cloud, indices of vertices of forming faces $f_k \in C_V(p_i)$ and face normals $\mathbf{n}_{f_k}$ of each point $p_i \in S$.

```
voro++ -c '%i %P' -o -p xmin  xmax ymin  ymax zmin  zmax 'coords.txt'

voro++ -c '%i %t' -o -p xmin  xmax ymin  ymax zmin  zmax 'coords.txt'

voro++ -c '%i %l' -o -p xmin  xmax ymin  ymax zmin  zmax 'coords.txt'
```

Algorithms for medial set construction

In Figs. S2 and S3, we illustrate two algorithms for medial set construction following Schröder *et al.*[2] and Chen *et al.*[8] Our procedure to create input files are described above and the actual *Python* codes along with input and output files can be accessed online (https://doi.org/10.7275/vqe1-sm17). Depending on how point normal data is computed we recommend users to use algorithm I or II. For class-A type, where it is possible to express the IMDS level set as an explicit function, it is possible to compute point normals exactly from gradient to the level set, we recommend using algorithm-I. This is advantageous for obtaining good quality medial set from a relatively coarse discrete mesh of domain boundary. Typically, coarseness (as defined in section B.2) $< 10^{-4}$ works well with this approach. For other cases (i.e class-B), it is recommended to use algorithm-II. Additionally, we found that having a 'regular' mesh i.e triangular faces of the discrete mesh are closer to having uniform edge lengths, on average produces a good quality medial surface. Often, generating a discrete mesh of a surface results in 'roughness' or displacing points slightly away from the intended surface. We refer the reader to *Mathematica* software (regularizeMesh.nb) that can be accessed in ref[7] which implements an algorithm to achieve edge length regularization by minimizing a functional $F_{reg} = \sum_{\langle ij \rangle}(L_{ij} - \bar{L})^2$ based on edge lengths(L) of triangular faces where $\bar{L}$ is the mean edge length while ensuring that vertices of the mesh lies on the intended surface. This software also contains a module which uses a gradient descent approach to push back all deviated vertices of the mesh onto surface either based on composition field data. Similar excercise to eliminate 'roughness' on the discrete mesh can also be performed for surfaces with explicit level set function.

Algorithm - I

INPUT

A. List of coordinates (x, y, z) for each point p_i on domain boundary S

B. List of components (n_x, n_y, n_z) of a normal vector $\hat{\mathbf{n}}_i$ for every point p_i in list A

C. List of coordinates (x, y, z) vertices v_j forming Voronoi cell $C_V(p_i)$ for every point p_i in list A

D. List of vertex indices j in that form a face $f_k \in C_V(p_i)$

E. List of components (nF_x, nF_y, nF_z) of a face normal vector $\hat{\mathbf{n}}_{f_k}$ for each face $f_k \in C_V(p_i)$

OUTPUT

List of coordinates (x, y, z) for every medial point $m_{\pm,i}$ on medial surface $MS_\pm$ in direction of normal $\pm\hat{\mathbf{n}}_i$

```
for ($i = 1, i_{max}$)    (Iterating through all points $p_i$ on domain boundary $S$)
    $\mathbf{l}_0 = [p_x, p_y, p_z]$              (coordinates for $p_i$)
    $\mathbf{n}_p = [n_x, n_y, n_z]$              (components of unit normal for $+\hat{\mathbf{n}}_i$)
    $\mathbf{n}_m = -[n_x, n_y, n_z]$             (components of unit normal for $-\hat{\mathbf{n}}_i$)
    for ($k = 1, k_{max}$)                  (Iterating through all faces $f_k \in C_V(p_i)$)
        $\mathbf{nF} = [n_{f_k,x}, n_{f_k,y}, n_{f_k,z}]$   (components for normal vector $\hat{\mathbf{n}}_{f_k}$ for face $f_k$)
        $\mathbf{l}_1 = [v_x, v_y, v_z]$              (coordinates of a vertex $v_j$ which lies on face $f_k$)
        $d_+(k) = \frac{(\mathbf{l}_1 - \mathbf{l}_0) \cdot \mathbf{nF}}{\mathbf{nF} \cdot \mathbf{n}_p}$     ($d_\pm(k)$ is the distance between $\mathbf{l}_0$ and
        $d_-(k) = \frac{(\mathbf{l}_1 - \mathbf{l}_0) \cdot \mathbf{nF}}{\mathbf{nF} \cdot \mathbf{n}_m}$     intersection of normal vector $\pm\hat{\mathbf{n}}_i$ with face $f_k$)
    end
    $m_{+,i} = \mathbf{l}_0 + \mathbf{n}_p \min_k[|d_+(k)|]$   For a every $p_i \in S$, this is the medial map onto $MS_+$
    $m_{-,i} = \mathbf{l}_0 + \mathbf{n}_m \min_k[|d_-(k)|]$   For a every $p_i \in S$, this is the medial map onto $MS_-$
end
```

Figure S2: Pseudo-code for computing medial set points $m_{\pm,i}$ by finding intersection of unit normal vector with a face belonging to $C_V(p_i)$ and lies along the direction of $\hat{\mathbf{n}}$

Algorithm - II

INPUT

A. List of coordinates (x, y, z) for each point p_i on domain boundary S
B. List of components (n_x, n_y, n_z) of a normal vector $\hat{\mathbf{n}}_i$ for every point p_i in list A
C. List of coordinates (x, y, z) vertices v_j forming Voronoi cell $C_V(p_i)$ for every point p_i in list A
D. List of vertex indices j in that form a face $f_k \in C_V(p_i)$

OUTPUT

List of coordinates (x, y, z) for every medial point $m_{\pm,i}$ on medial surface $MS_\pm$ in direction of normal $\pm\hat{\mathbf{n}}_i$

```
for ($i = 1, i_{max}$)    (Iterating through all points $p_i$ on domain boundary $S$)
    $\mathbf{l}_0 = [p_x, p_y, p_z]$            (coordinates for $p_i$)
    $\mathbf{n}_p = [n_x, n_y, n_z]$            (components of unit normal for $+\hat{\mathbf{n}}_i$)
    $\mathbf{n}_m = -[n_x, n_y, n_z]$           (components of unit normal for $-\hat{\mathbf{n}}_i$)
    for ($j = 1, j_{max}$)    (Iterating through all vertices $v_j \in C_V(p_i)$)
        $\mathbf{l}_1 = [v_x, v_y, v_z]$        (coordinates of vertex with index $j$ from $C_V(p_i)$
        $d_+(j) = (\mathbf{l}_1 - \mathbf{l}_0) \cdot \mathbf{n}_p$  (distance between $p_i$ and $v_j$ projected along $+\hat{\mathbf{n}}_i$ )
        $d_-(j) = (\mathbf{l}_1 - \mathbf{l}_0) \cdot \mathbf{n}_m$  (distance between $p_i$ and $v_j$ projected along $-\hat{\mathbf{n}}_i$ )
    end
    $m_{+,i} = \mathbf{l}_0 + \mathbf{n}_p \max_j[d_+(j)]$    For a every $p_i \in S$, this is the medial map onto $MS_+$
    $m_{-,i} = \mathbf{l}_0 + \mathbf{n}_m \max_j[d_-(j)]$    For a every $p_i \in S$, this is the medial map onto $MS_-$
end
```

Figure S3: Pseudo-code for computing medial set points $m_{\pm,i}$ by finding farthest vertex belonging to $C_V(p_i)$ and lies along the direction of $\hat{\mathbf{n}}$

References

(1) Amenta N,Bern MW, Kamvysselis M, A new voronoi-based surface reconstruction algorithm, *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 415-421 (1998).

(2) Schröder GE, Ramsden SJ, Christy AG, Hyde ST, Medial surfaces of hyperbolic structures *Euro. Phys. J B*, **35**, 551-564 (2003).

(3) Wohlgemuth M, Yufa N, Hoffman J, Thomas EL, Triply Periodic Bicontinuous Cubic Microdomain Morphologies by Symmetries *Macromolecules* **34**, 6083-6089 (2001).

(4) Rycroft CH, A three-dimensional voronoi cell library in C++, *Chaos* **19**, 041111 (2009).

(5) Arora A, Qin J, Morse DC, Delaney KT, Fredrickson GH, Bates FS, Dorfman KD, Broadly accessible self-consistent field theory for block polymer materials discovery *Macromolecules* **49**, 4675-4690 (2016).

(6) Feng X, Burke CJ, Zhuo M, Guo H, Yang K, Reddy A, Prasad I, Ho R-M., Avgeropoulos A., Grason GM, Thomas EL, Seeing mesoatomic distortions in soft matter crystals of a double gyroid block copolymer, *Nature* **575**, 175-179 (2019).

(7) Feng X, Burke CJ, Zhuo M, Guo H, Yang K, Reddy A, Prasad I, Ho R-M., Avgeropoulos A., Grason GM, Thomas EL, source data for *Nature* **575**, 175-179 (2019) , https://scholarworks.umass.edu/data/88/

(8) Chen H, Jin C, Competition brings out the best: modelling the frustration between curvature energy and chain stretching energy of lyotropic liquid crystals in bicontinuous cubic phases, *Interface Focus*, **7**, 20160114 (2017).