Received: 25 January 2021 | Accepted: 2 August 2022

DOI: 10.1111/rssc.12592

ORIGINAL ARTICLE

ROYAL STATISTICAL SOCIETY DATA | EVIDENCE | DECISIONS
Series C Applied Statistics

Contour models for physical boundaries enclosing star-shaped and approximately star-shaped polygons

Hannah M. Director[1] | **Adrian E. Raftery**[2]

[1]Department of Statistics, University of Washington, Seattle, Washington, USA

[2]Departments of Statistics and Sociology, University of Washington, Seattle, Washington, USA

Correspondence
Hannah M. Director, Department of Statistics, University of Washington, Seattle, WA, USA.
Email: hdirector7@gmail.com

Funding information
National Oceanic and Atmospheric Administration, Grant/Award Number: NA15OAR4310161; National Science Foundation, Grant/Award Number: DGE-1256082

Abstract

Boundaries on spatial fields divide regions with particular features from surrounding background areas. Methods to identify boundary lines from interpolated spatial fields are well established. Less attention has been paid to how to model sequences of connected spatial points. Such models are needed for physical boundaries. For example, in the Arctic ocean, large contiguous areas are covered by sea ice, or frozen ocean water. We define the ice edge contour as the ordered sequences of spatial points that connect to form a line around set(s) of contiguous grid boxes with sea ice present. Polar scientists need to describe how this contiguous area behaves in present and historical data and under future climate change scenarios. We introduce the Gaussian Star-shaped Contour Model (GSCM) for modelling boundaries represented as connected sequences of spatial points such as the sea ice edge. GSCMs generate sequences of spatial points via generating sets of distances in various directions from a fixed starting point. The GSCM can be applied to contours that enclose regions that are star-shaped polygons or approximately star-shaped polygons. Metrics are introduced to assess the extent to which a polygon deviates from star-shapedness. Simulation studies illustrate the performance of the GSCM in different situations.

KEYWORDS
contours, sea ice, spatial statistics, star-shaped polygons

1 | INTRODUCTION

Boundaries that enclose regions are often subjects of scientific interest. Contour lines divide a contiguous region with some defining feature(s) from surrounding background areas. This paper introduces the Gaussian Star-shaped Contour Model (GSCM) for the distribution of such contours. The GSCM is designed for modelling contours that enclose regions that are *star-shaped polygons* (Definition 4) or approximately star-shaped polygons.

The GSCM is motivated by the need for models appropriate for the sea ice edge. In the Arctic ocean, large areas are covered by sea ice, or frozen ocean water. The sea ice edge contour forms a boundary between the area covered by sea ice and the surrounding open water. Data on the location of sea ice is provided by remotely sensed gridded products that indicate if a grid box is ice-covered. Since most grid boxes with sea ice are in one contiguous region, this data can be converted to binary values that indicate if each grid box is inside or outside the main ice-covered area. The ice edge contour is then the set of points that connect to form the boundary between the grid boxes inside and outside the region. Thus the ice edge contour is modelled as a collection of ordered, connected spatial points in the two-dimensional (2D) plane.

Figure 1 shows a sample sea ice edge contour. Polar scientists are interested in the variability of ice edge contours and the extent to which ice edge contours change over time. Predictions of the ice edge contour are also needed weeks to months in advance for maritime planning. Distributions of ice edge contours are inferred from observing multiple ice edge contours, such as those observed at different times.

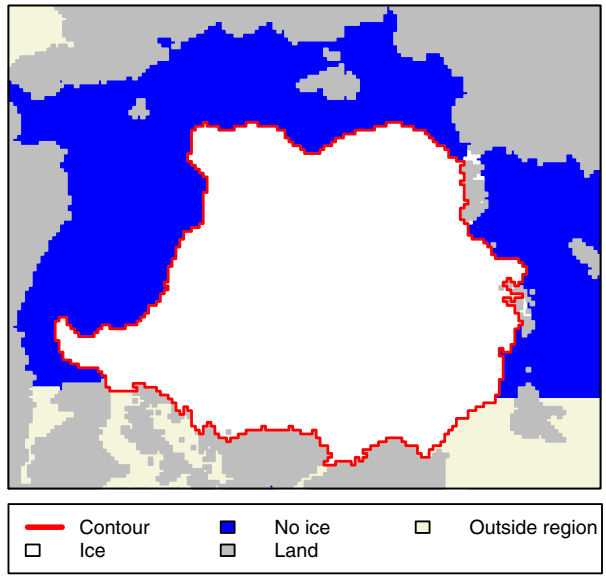

FIGURE 1 The contour forming the boundary around the main contiguous area covered by sea ice in a central region of the Arctic in September 2017. Section 6 introduces methods for modelling contours like this one that enclose approximately star-shaped polygons. [Colour figure can be viewed at wileyonlinelibrary.com]

Previous research has developed contour and boundary models for other data types. Analysis has often focused on inferring a single boundary from observations on a spatial field. Research on exceedance levels has developed methods to infer contours that describe where a property goes above some level (Bolin & Lindgren, 2015; French & Hoeting, 2016; French & Sain, 2013). Wombling methods find contours by identifying curvilinear gradients (Banerjee & Gelfand, 2006; Womble, 1951). Statistical shape analysis (Dryden & Mardia, 2016; Srivastava & Klassen, 2016) provides tools to model boundaries corresponding to particular objects with discernible features. None of these methods infer distributions of contours from multiple observed contours like the sea ice edge data. Similarly, remote sensing research on sea ice has focused on how to infer the location of the sea ice edge at individual time points given information from active and passive remote sensing. So, the focus is again on how to infer a specific boundary from data. The GSCM addresses a different question: how to model the distribution of sea ice edges given samples of fully observed contours represented as connected sequences of points.

We first develop and assess the GSCM for modelling boundaries enclosing star-shaped and approximately star-shaped polygons. We then apply GSCMs to solve our motivating problem of how to model the sea ice edge. Section 2 defines contours and how to represent them. Section 3 introduces the GSCM for modelling contours enclosing star-shaped polygons and discusses model fitting. Section 4 introduces a metric for assessing coverage of star-shaped contours and Section 5 presents results from simulation studies. Section 6 extends the GSCM to contours enclosing approximately star-shaped polygons. Section 7 applies the GSCM to sea ice edge contours. Section 8 concludes the paper with discussion, including Section 8.1 that examines how the GSCM compares to other contour and boundary methods.

2 | CONTOUR DEFINITIONS

Our focus is on modelling contours that act as the boundary between a region that has some feature(s) and the surrounding background region, that is, ice-covered areas versus open water. There are multiple ways such contours could be defined. In this section, we give two representations for these contours that will be used as a basis for subsequent modelling and assessment.

2.1 | Point-sequence representation

Contours and the regions they enclose can be described using connected sequences of points. We refer to this description of a contour as a *point-sequence representation*. We define the following concepts.

Definition 1. Contour point sequence, $\boldsymbol{S}$: An ordered set of spatial points $\boldsymbol{S} = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n)$, with $n > 2$, where each $\boldsymbol{s}_i$ consists of the x-y coordinates of a spatial location.

Definition 2. Contour line, $\overline{\boldsymbol{S}}$: The connected line formed by connecting $\boldsymbol{s}_i$ to $\boldsymbol{s}_{i+1}$ for $i = 1, \ldots, n-1$ and connecting $\boldsymbol{s}_n$ to $\boldsymbol{s}_1$.

Definition 3. Enclosed polygon, $\underline{\boldsymbol{S}}$: The polygon formed by the interior of the contour line $\overline{\boldsymbol{S}}$.

The left panel of Figure 2 illustrates these definitions for a contour described by a point-sequence representation. The main advantage of the point-sequence representation is its flexibility. Any contour enclosing a polygon can be represented exactly with a sequence of spatial

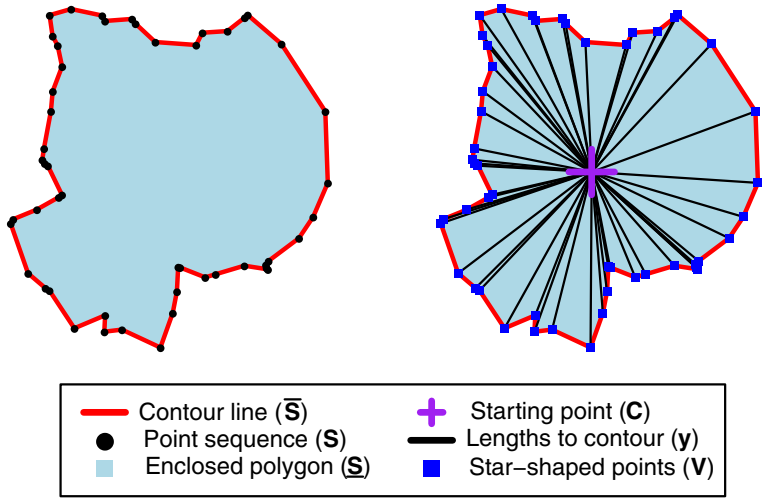

FIGURE 2 Components of a contour represented by a point-sequence representation (left) and a star-shaped representation (right) [Colour figure can be viewed at wileyonlinelibrary.com]

points, $\boldsymbol{S}$. Also, the level of detail represented can be increased simply by increasing the number of points.

A binary grid indicating membership in the region of interest can be converted to the point-sequence representation and modelled accordingly. The contour $\boldsymbol{S}$ is made up of corner points of grid boxes that touch the outside of the region on one side and the inside of the region on the other. The points of $\boldsymbol{S}$ are ordered to align with the order in which they would be touched if one were to trace around the boundary. Where and in what direction to start tracing around the boundary is arbitrary. These choices only determine the indexing of the points in $\boldsymbol{S}$, not the line, $\overline{\boldsymbol{S}}$, or enclosed polygon, $\underline{\boldsymbol{S}}$. The point-sequence representation can be used with any grid resolution, though finer grids will require more points in $\boldsymbol{S}$. For sea ice, the binary grid is what is observed.

2.1.1 | Notation

We need to distinguish between points, lines and polygons. An ordered sequence of spatial points will be denoted by a boldface letter, such as $\boldsymbol{S}$. A line formed by connecting points will be denoted with an overline, such as $\overline{\boldsymbol{S}}$. A line segment will be denoted by an overline over two letters that represent the start and end points of the segment, such as $\overline{\boldsymbol{CD}}$. The polygon enclosed by a line will be denoted by an underline, such as $\underline{\boldsymbol{S}}$.

2.1.2 | Contours with fractal characteristics

We acknowledge that the point-sequence representation does not directly account for contours whose true nature is fractal. As such, representing fractal contours as connected sequences of points may be an approximation. In contours of sea ice and other physical world examples, as the spatial scale of observations increases, the level of detail of the contour also increases (Mandelbrot, 1967). With each increase in spatial resolution additional line segments are needed to describe the increased detail. In other words, the length of the contour increases each time the

spatial resolution increases. This fractal nature of some contours means that these contours can never be fully expressed with a finite, ordered set of spatial points. These contours' true lengths are infinite.

For the applications of interest, however, limits exist on the precision of measurements and the relevant scale of scientific interest. While the fractal or Hausdorff dimension can be estimated (Gneiting et al., 2012), the level of detail of the boundary that will be measured or needed will rarely be a fractal. So, making the simplifying assumption that the contour can be defined by a sequence of spatial points is reasonable for modelling the sea ice edge contour. Additional discussion of contours as fractals is given in Section 8.2.

2.2 | Star-shaped representation

Point-sequence representations are natural and describe contours accurately. However, point-sequence representations are ill-suited for describing multiple contours and their distributional behaviour. Contours differ in length, so two points with the same index on two different point-sequence representations are not likely to be in the same physical location. Comparing spatially dependent features and inferring distributions is therefore difficult. We build an alternate *star-shaped representation* that avoids the weaknesses of point-sequence representations. The star-shaped representation is appropriate for contours that enclose star-shaped polygons or approximately star-shaped polygons. The general idea of star-shaped representations is to describe contours based on the length of the lines extending to the contour from a fixed point in different directions rather than as a sequence of spatial points as in a point-sequence representation. Figure 2 illustrates how a point-sequence representation (left) and a star-shaped representation (right) differ.

Before defining the star-shaped representation, we review the standard definitions of a star-shaped polygon and its kernel (Preparata & Shamos, 1985, p. 18).

Definition 4. Star-shaped polygon: A polygon $\underline{\boldsymbol{P}}$ is star-shaped if there exists a point $\boldsymbol{D}$ within $\underline{\boldsymbol{P}}$ such that the line segment $\overline{\boldsymbol{Dp}}$ is fully contained within $\underline{\boldsymbol{P}}$ for all points $\boldsymbol{p}$ on line $\overline{\boldsymbol{P}}$.

All convex polygons are star-shaped, but the set of star-shaped polygons is substantially larger. Figure 3 shows nine example star-shaped polygons. These examples highlight the variety of polygons that can be star-shaped.

Definition 5. Kernel of a star-shaped polygon, $\mathcal{K}(\underline{\boldsymbol{P}})$: The set of point(s) that satisfy the criterion for $\boldsymbol{D}$ in Definition 4 is referred to as the kernel of the polygon, $\mathcal{K}(\underline{\boldsymbol{P}})$.

Convex polygons are the subset of star-shaped polygons such that $\mathcal{K}(\underline{\boldsymbol{P}}) = \underline{\boldsymbol{P}}$.

For any star-shaped polygon, $\underline{\boldsymbol{S}}$, lines can be drawn from some point, $\boldsymbol{C}$, in the kernel of $\underline{\boldsymbol{S}}$ to all points on a contour, $\overline{\boldsymbol{S}}$. Assume for the moment that the contour point sequence $\boldsymbol{S}$ is unknown, but that the location of $\boldsymbol{C}$ is known along with the lengths and directions of the lines from $\boldsymbol{C}$ to $\boldsymbol{S}$. Then $\boldsymbol{S}$ could be derived from this information with trigonometry. Taking inspiration from this fact, we develop the star-shaped representation. First we define a line set:

Definition 6. Line set, $\mathcal{L}(\boldsymbol{C}, \boldsymbol{\theta})$: A set of $p > 2$ lines, $\mathcal{L} = (\ell_1, \ldots, \ell_p)$, extending infinitely outward from some starting point $\boldsymbol{C} = (C_x, C_y)$ at p unique angles, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)$, where C_x and C_y are x- and y-coordinates, respectively.

We also define a set of spatial points that produce a star-shaped polygon when connected in order:

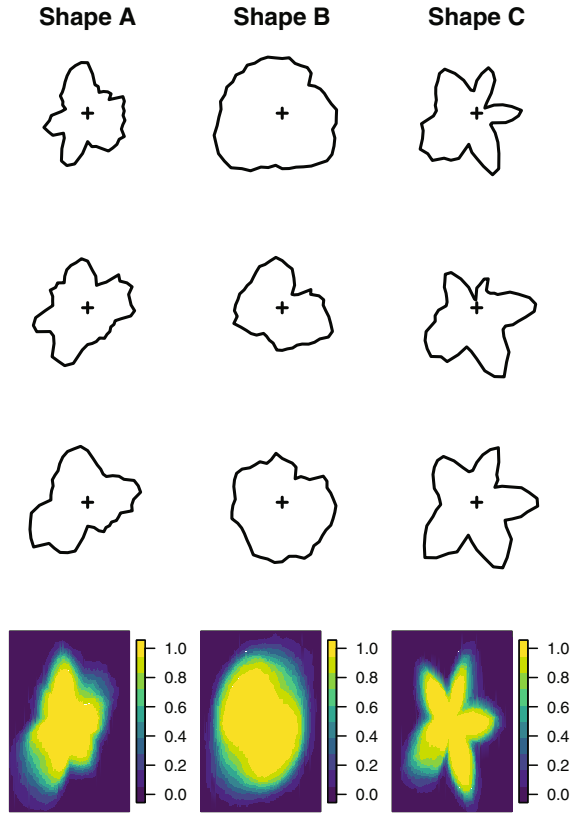

FIGURE 3 Rows 1–3: Nine examples of contours enclosing star-shaped polygons with an exponential covariance generated from Gaussian star-shaped contour models (GSCMs) with three different parameter settings, organised by column. The cross sign denotes the starting point, $\boldsymbol{C}$. Row 4: Estimated probability of a grid box being contained within contours generated by GSCMs with the column's parameters settings. Probabilities estimated from 100 generated contours. The GSCM parameter settings are referred to as *Shape A* (left), *Shape B* (middle) and *Shape C* (right). For all shapes $p = 50$ and $\kappa = 2$. Values for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are given in Appendix C. [Colour figure can be viewed at wileyonlinelibrary.com]

Definition 7. Star-shaped point set, $\boldsymbol{V}(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{y})$: A set of $p > 2$ spatial points $\boldsymbol{V} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_p)$ such that

$$\boldsymbol{v}_i = (C_x + y_i \cos(\theta_i), C_y + y_i \sin(\theta_i)), \tag{1}$$

where $\boldsymbol{y} = (y_1, \ldots, y_p)$ is a set of p distances, $\boldsymbol{C} = (C_x, C_y)$ is a spatial point, and $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)$ is a set of p unique angles.

A star-shaped point set can be used to represent a contour when the distances are selected systematically:

Definition 8. Star-shaped representation, $\tilde{\boldsymbol{V}}(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})$: Let $\underline{\boldsymbol{S}}$ be a star-shaped polygon and $\boldsymbol{C} \in \mathcal{K}(\underline{\boldsymbol{S}})$ be a starting point. Then, the star-shaped representation of the contour $\boldsymbol{S}$, denoted by $\tilde{\boldsymbol{V}}(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})$, is the star-shaped point set, $\boldsymbol{V}(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{y})$, where $\boldsymbol{y} = (y_1, \ldots, y_p)$ is the set of distances from $\boldsymbol{C}$ to the intersection point of the contour line $\overline{\boldsymbol{S}}$ and each line ℓ_i in the line set $\mathcal{L}(\boldsymbol{C}, \boldsymbol{\theta})$.

The right panel of Figure 2 shows the components of the star-shaped representation for a sample contour. Let $\overline{\boldsymbol{V}}$ refer to the contour line formed by connecting $\boldsymbol{v}_i$ to $\boldsymbol{v}_{i+1}$ for $i = 1, \ldots, p-1$ and $\boldsymbol{v}_p$ to $\boldsymbol{v}_1$ for $p > 2$. Let $\underline{\boldsymbol{V}}$ refer to the polygon contained within $\overline{\boldsymbol{V}}$.

Theorem 1. *Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)$ with $\theta_i < \theta_{i+1}$ and $\theta_i \in (0, 2\pi)$ for all i. For a star-shaped polygon $\underline{\boldsymbol{S}}$ there exist $\boldsymbol{\theta}$ and $\boldsymbol{y}$ such that $\tilde{\boldsymbol{V}}(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{y}) = \boldsymbol{S}$ for any $\boldsymbol{C} \in \mathcal{K}(\underline{\boldsymbol{S}})$. (Proof in Appendix A.)*

Corollary 1. *Let ℓ_θ denote the line that extends infinitely outward from $\boldsymbol{C}$ at angle $\theta \in (0, 2\pi)$ and that intersects $\overline{\boldsymbol{S}}$. For any θ, the line ℓ_θ is distinct, that is, $\ell_\theta \neq \ell_{\theta'}$ for any θ, θ' such that $\theta \neq \theta'$. (Proof in Appendix A.)*

The star-shaped representation allows for finding how contours differ and what the variability of the contours is in different spatial areas. For example, consider two contours $\boldsymbol{S}_k$ and $\boldsymbol{S}_\ell$ described with star-shaped representations, $\tilde{\boldsymbol{V}}(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S}_k)$ and $\tilde{\boldsymbol{V}}(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S}_\ell)$, for common line set $\mathcal{L}(\boldsymbol{C}, \boldsymbol{\theta})$. To find how much further one contour extends in any direction, simply find the difference between $y_{i,k}$ and $y_{i,\ell}$ where $y_{i,k}$ and $y_{i,\ell}$ are the distances from $\boldsymbol{C}$ to contours $\boldsymbol{S}_k$ and $\boldsymbol{S}_\ell$ along a line extending at angle θ_i. The variability of the contours along any line $\ell_i \in \mathcal{L}(\boldsymbol{C}, \boldsymbol{\theta})$ is estimated from the variability of the corresponding $\boldsymbol{y}_i$ values in the contours' star-shaped representations.

For a contour enclosing a star-shaped polygon, the star-shaped representation is identical to the point-sequence representation when $p = n$, $\boldsymbol{C} \in \mathcal{K}(\underline{\boldsymbol{P}})$, and θ_i aligns with the direction of the line segments $\overline{\boldsymbol{C}\boldsymbol{s}_i}$ for all i. When these conditions are met, the points $\boldsymbol{V}$ are the same for any choice of starting point $\boldsymbol{C}$ within the kernel of $\underline{\boldsymbol{S}}$. However, the angles $\boldsymbol{\theta}$ and lengths $\boldsymbol{y}$ will differ depending on $\boldsymbol{C}$.

3 | STAR-SHAPED CONTOUR MODEL

3.1 | General model

We now propose the *star-shaped contour model* for generating contours that enclose star-shaped polygons. In later sections, we will build upon this idealised model for star-shaped polygons to develop a model appropriate for ice edge contours that are only approximately star-shaped.

Definition 9. Star-shaped contour model, $\{\boldsymbol{C}, \boldsymbol{\theta}, \Gamma\}$: Let $\boldsymbol{C} = (C_x, C_y)$ be a fixed starting point, let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)$ be a fixed set of $p > 2$ unique angles, and let Γ be a probability distribution from which a set of values $\boldsymbol{y} = (y_1, \ldots, y_p) > 0$ can be drawn. This set of parameters form a star-shaped probability model if each drawn set $\boldsymbol{y}$ can be used to form a corresponding star-shaped points set, $\boldsymbol{V}(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{y})$, as given in Definition 7.

We now consider a distribution that is appropriate in many circumstances. We assume that $\boldsymbol{y}$ follows a Gaussian distribution,

$$\boldsymbol{y} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{2}$$

where $\boldsymbol{\mu}$ is a mean vector and the parameter $\boldsymbol{\Sigma}$ is a positive-definite covariance function. We further assume that $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are such that mass on non-positive $\boldsymbol{y}$ is negligible. (In practice, if in a small proportion of cases, a generated y_i is non-positive, its value can be set to some small $\eta > 0$.) We call this model the GSCM. The GSCM can be seen as a finite approximation to the planar version of Gaussian Random Particles proposed in Hansen et al. (2015). GSCMs can produce a fairly flexible set of contours. The first three rows in Figure 3 illustrate some types of contours GSCMs can produce.

Because of how $\boldsymbol{y}$ is constructed, reasonable $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ that align with typical observable $\boldsymbol{y}$ values will avoid substantial non-positive $\boldsymbol{y}$. The values $\boldsymbol{y}$ represent distances from a starting point $\boldsymbol{C}$, so are automatically non-negative. Some points in the kernel of the polygon will typically be centrally located points. Lengths close to zero can be avoided by using one of these points for $\boldsymbol{C}$.

Covariance matrices, $\boldsymbol{\Sigma}$, are based on the structure of the lines in the line set. The correlation structure for the set of distances, $\boldsymbol{y}$, is a function of the angles, $\boldsymbol{\theta}$, of the lines in the line set. Covariances based on angles are complicated by the fact that 0 and 2π represent the same angle. So, the difference between two angles does not necessarily correspond to how far apart the angles actually are. Specialised covariance functions have been derived that remain valid when distances are indexed by angle (Gneiting, 2013). Denote the angle between θ_i and θ_j by $d(\theta_i, \theta_j) \in [0, \pi]$.

Typically, the correlation between y_i and y_j will decrease as $d(\theta_i, \theta_j)$ increases. For the simulation examples in this paper, we focus on an exponential covariance structure, $\boldsymbol{\Sigma}(\boldsymbol{\sigma}, \kappa)$ where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_p)$ and $\kappa > 0$. The element, $\boldsymbol{\Sigma}_{ij}$, in the ith row and jth column of this covariance is

$$\boldsymbol{\Sigma}_{ij} = \sigma_i \sigma_j \exp\left(-\frac{d(\theta_i, \theta_j)}{\kappa}\right). \tag{3}$$

Different parametric covariance structures give different forms for how the correlation between y_i and y_j decreases as $d(\theta_i, \theta_j)$. For an exponential covariance, the parameter κ controls how rapidly the contour can change over space, that is, the roughness or smoothness of the contour. For example, Figure 4 shows sample contours with the same $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ values as in Figure 3. However, the contours appear smoother because of the different covariance structure. The resolution of the data should also be considered in considering the roughness of the contour. If the data resolution is high, information about how smooth or rough the contours are can be assessed and modelled accurately with the κ selection. If the resolution is low, though, only smoother contours will be possible, and κ must be restricted accordingly. In other words, estimates of κ from data may be limited by the data's resolution.

3.2 | Fitting GSCMs

We now turn to building a GSCM given observed contours. We assume that the data are N observed contours, $\mathbf{S} = (\boldsymbol{S}_1, \ldots, \boldsymbol{S}_N)$, that enclose regions that are star-shaped polygons. (Note that each $\boldsymbol{S}$ may be derived from binary grid as described in Section 2.) We also assume that the contours are generated from a common, but unknown, $\boldsymbol{C}$ and $\boldsymbol{\theta}$, that is, $\boldsymbol{C} = \boldsymbol{C}_1 = \cdots = \boldsymbol{C}_N$ and $\boldsymbol{\theta} = \boldsymbol{\theta}_1 = \cdots = \boldsymbol{\theta}_N$. We first find a starting point, $\hat{\boldsymbol{C}}$, and angles, $\hat{\boldsymbol{\theta}}$. Then, we estimate $\boldsymbol{\mu}$ and the parameters controlling $\boldsymbol{\Sigma}(\cdot)$ based on the observed $\boldsymbol{y}$ for the selected $\hat{\boldsymbol{C}}$ and angles $\hat{\boldsymbol{\theta}}$.

3.2.1 | Fixing the starting point $\hat{\boldsymbol{C}}$ and the set of angles $\hat{\boldsymbol{\theta}}$

The accuracy of the GSCM depends on the selection of $\hat{\boldsymbol{C}}$ and $\hat{\boldsymbol{\theta}}$. If $\hat{\boldsymbol{C}}$ is selected outside the kernel of the polygons, the line set will not be able to intersect with all sections of the contour line. So some area within the true enclosed contour will be missed by the contour model. Similarly, if $\hat{\boldsymbol{\theta}}$ is not dense enough, some direction changes in the contour will not be modelled. The influence of the density of $\hat{\boldsymbol{\theta}}$ will be greater if the contour changes direction rapidly. The set of angles, $\hat{\boldsymbol{\theta}}$, is selected to keep the mean difference in area between the observed contours' enclosed polygons and the star-shaped representations of these contours' enclosed polygons below some value.

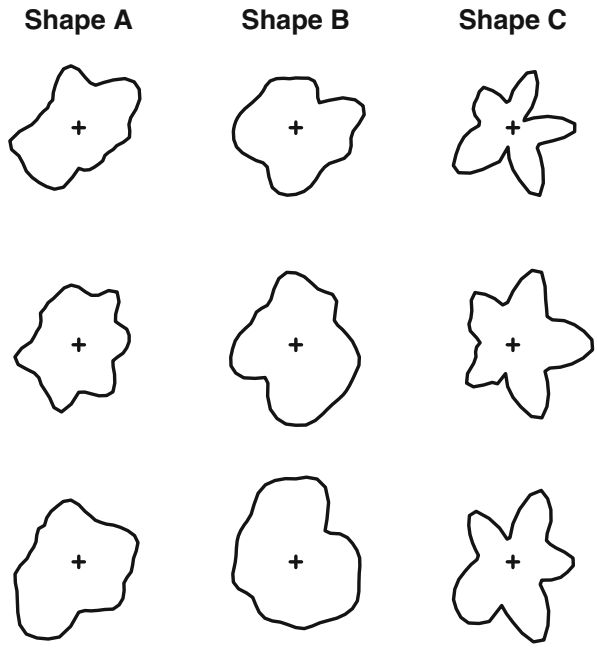

FIGURE 4 Rows 1–3: Nine examples of contours enclosing star-shaped polygons with a multi-quadric covariance generated from Gaussian star-shaped contour models (GSCMs) with three different parameter settings, organised by column. The cross sign denotes the starting point, $\boldsymbol{C}$. The GSCM parameter settings are referred to as *Shape A* (left), *Shape B* (middle) and *Shape C* (right). For all shapes $p = 50$, $\tau = 2$ and $\delta = 0.5$. Values for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are given in Appendix C.

Following Corollary 1, any set of distinct angles can be used to form a star-shaped representation of a contour. We recommend using evenly-spaced angles. Figure 5 illustrates how a star-shaped contour is approximated with a star-shaped representation with evenly-spaced $\boldsymbol{\theta}$. Using evenly spaced angles ensures that the model represents all sections of the contours with the same level of precision. While some sections of the contours may change more rapidly than others and require more lines to represent well, exactly where these changes occur is hard to determine without having already modelled the contour. Therefore it is easier to select a density of lines to use consistently for all sections of the contour.

This sensitivity of GSCMs to the choice of $\hat{\boldsymbol{C}}$ and $\hat{\boldsymbol{\theta}}$ motivates the development of a careful procedure to select them. Our procedure seeks to minimise the area that cannot be represented by the GSCM while balancing computational constraints. We fix a starting point, $\hat{\boldsymbol{C}}$, and set of angles, $\hat{\boldsymbol{\theta}}$, that can be used to describe the observed contours accurately. We first describe how to find $\hat{\boldsymbol{C}}$ conditional on $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ conditional on $\boldsymbol{C}$ separately. Then, we describe an iterative algorithm to fix both values together.

Finding $\hat{\boldsymbol{C}}$ conditional on $\boldsymbol{\theta}$: The starting point $\hat{\boldsymbol{C}}$ used in modelling is selected to minimise the difference in area between the observed contours' enclosed polygons and the star-shaped representations of the observed contours' enclosed polygons. Conditional on $\boldsymbol{\theta}$, we define the set that differs between an observed polygon, $\underline{\boldsymbol{S}}_i$, and a star-shaped representation of that contour with starting point, $\hat{\boldsymbol{C}}$, as

$$A(\hat{\boldsymbol{C}}, \boldsymbol{\theta}, \boldsymbol{S}_i) := \left\{ (\underline{\boldsymbol{S}}_i^c \cap \underline{\tilde{\boldsymbol{V}}}(\hat{\boldsymbol{C}}, \boldsymbol{\theta}, \boldsymbol{S}_i)) \cup (\underline{\boldsymbol{S}}_i \cap \underline{\tilde{\boldsymbol{V}}}^c(\hat{\boldsymbol{C}}, \boldsymbol{\theta}, \boldsymbol{S}_i)) \right\}. \tag{4}$$

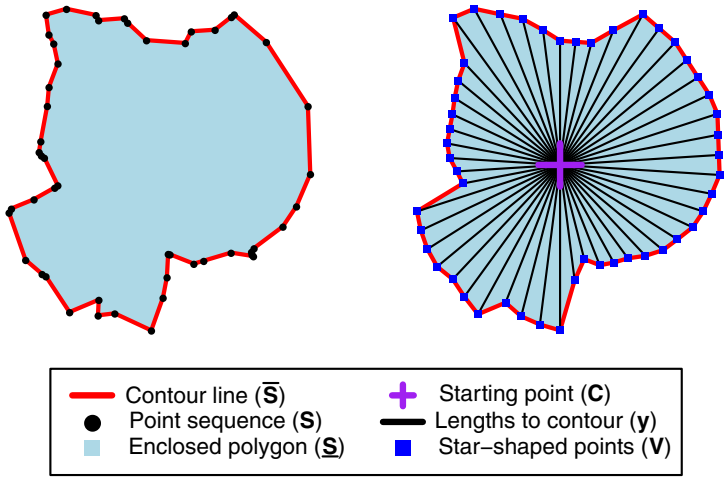

FIGURE 5 Components of a contour represented by a point-sequence representation (left) and a star-shaped representation approximating the point sequence with evenly spaced angles (right) [Colour figure can be viewed at wileyonlinelibrary.com]

where the superscript c denotes the complement of the set. The area contained within this set is denoted by $|A(\hat{\boldsymbol{C}}, \boldsymbol{\theta}, \boldsymbol{S}_i)|$. The lower this area, the more accurately the GSCM will be able to represent the enclosed polygons. So, selecting a $\hat{\boldsymbol{C}}$ such that $|A(\hat{\boldsymbol{C}}, \boldsymbol{\theta}, \boldsymbol{S}_i)| = 0$ for all i would be optimal. Assuming $\boldsymbol{\theta}$ is known, Theorem 1 guarantees that at least one point $\hat{\boldsymbol{C}}$ exists where $|A(\hat{\boldsymbol{C}}, \boldsymbol{\theta}, \boldsymbol{S}_i)| = 0$ for all i. If only one point satisfies this condition, then $\hat{\boldsymbol{C}} = \boldsymbol{C}$. However, any location $\hat{\boldsymbol{C}}$ for which $|A(\hat{\boldsymbol{C}}, \boldsymbol{\theta}, \boldsymbol{S}_i)|$ is approximately zero will be able to represent the area within the contour well. As such, we select $\hat{\boldsymbol{C}}$ numerically with

$$\hat{\boldsymbol{C}} = \underset{\boldsymbol{d} \in \boldsymbol{D}}{\operatorname{argmin}} \left\{ f(|A(\boldsymbol{d}, \boldsymbol{\theta}, \boldsymbol{S}_1)|, \ \ldots \ , |A(\boldsymbol{d}, \boldsymbol{\theta}, \boldsymbol{S}_N)|) \right\}, \tag{5}$$

where f is the mean function and $\boldsymbol{D}$ is a grid of points. The finer the grid of $\boldsymbol{D}$ the closer $|A(\hat{\boldsymbol{C}}, \boldsymbol{\theta}, \boldsymbol{S}_i)|$ will be to zero. The number of points in $\boldsymbol{D}$ can be reduced by restricting assessment to the kernels of star-shaped polygons.

Theorem 2. *Let $\underline{\boldsymbol{S}}$ be a set of N star-shaped polygons, let $\boldsymbol{C}$ be the true starting point, and let $\hat{\boldsymbol{\mathcal{K}}}(\underline{\boldsymbol{S}}) = \mathcal{K}(\underline{\boldsymbol{S}}_1) \cap \ \ldots \ \cap \mathcal{K}(\underline{\boldsymbol{S}}_N)$ denote the intersection of the kernels of all polygons. Then, $\boldsymbol{C} \in \hat{\boldsymbol{\mathcal{K}}}(\underline{\boldsymbol{S}})$. (Proof in Appendix A.)*

Therefore, we need only perform the optimisation in Equation (5) for $\boldsymbol{D} \in \hat{\boldsymbol{\mathcal{K}}}(\underline{\boldsymbol{S}})$. Algorithms for computing $\hat{\boldsymbol{\mathcal{K}}}(\underline{\boldsymbol{S}})$ are given in Appendix B.

Finding $\hat{\boldsymbol{\theta}}$ conditional on $\boldsymbol{C}$: As motivated in Section 3.2.1, GSCMs can be fit with evenly spaced angles. So, finding the set of angles $\hat{\boldsymbol{\theta}}$ only requires finding $\hat{p}$, the number of elements in $\hat{\boldsymbol{\theta}}$.

Since $\hat{p}$ controls the dimensions of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, larger $\hat{p}$ requires more computation. We then identify the approximately lowest $\hat{p}$ that keeps the mean difference in area below some value. To allow comparisons of polygons of different sizes, we often express this allowable mean difference in area as a proportion, δ, of the average area of the polygons. This constraint on the allowable differing area is then

$$f(|A(\boldsymbol{C}, \hat{\boldsymbol{\theta}}, \boldsymbol{S}_1)|, \ \ldots \ , |A(\boldsymbol{C}, \hat{\boldsymbol{\theta}}, \boldsymbol{S}_N)|) < \frac{\delta}{N} \sum_{i=1}^{N} |\underline{\boldsymbol{S}}_i|, \tag{6}$$

Algorithm 1. Finding $\hat{\boldsymbol{\theta}}$ conditional on $\boldsymbol{C}$

```
Initialise p^(0) and set t ← 0
Compute RHS of Equation (6)
while Equation (6) does not hold do
    Compute θ^(t) for p^(t)
    Compute LHS of Equation (6) with θ = θ^(t)
    if Equation (6) does not hold then
        Set t ← t + 1
        Set p^(t+1) ← a p^(t) where a > 1
    else
        Set θ̂ ← θ̂^(t)
    end
end
```

where f is the mean function and $|\underline{\boldsymbol{S}_i}|$ denotes the area of the polygon $\underline{\boldsymbol{S}_i}$. We use Algorithm 1 to find an approximately minimal $\hat{p}$ that satisfies the constraint in Equation (6). To avoid selecting a larger $\hat{p}$ than necessary, $p^{(0)}$ should generally be initialised such that Equation (6) is not satisfied. Smaller values of a will make $\hat{p}$ more precise, but will require more computation than larger a. Using lower values of δ will generally result in higher $\hat{p}$ and lower differences in area. Section 5.3 uses simulation to explore different δ.

Finding $\hat{\boldsymbol{C}}$ and $\hat{\boldsymbol{\theta}}$: In practice, neither $\hat{\boldsymbol{C}}$ nor $\hat{\boldsymbol{\theta}}$ will be known. So, to find both values, we iterate between setting $\hat{\boldsymbol{C}}$ conditional on $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\theta}}$ conditional on $\hat{\boldsymbol{C}}$. Algorithm 2 describes this process. As in Algorithm 1, the initial value, $p^{(0)}$, should be selected to be low enough that Equation (6) is not satisfied. Otherwise, we may select a larger $\hat{p}$ than is needed. Smaller values of a will result in more precise determination of $\hat{p}$, but will require more computation.

3.2.2 | Computing a posterior

Once $\hat{\boldsymbol{C}}$ and $\hat{\boldsymbol{\theta}}$ are determined, model fitting is straightforward. For each observed contour, the observed $\boldsymbol{y}$ values are computed given $\hat{\boldsymbol{C}}$ and $\hat{\boldsymbol{\theta}}$. Then from Equation (2), the corresponding likelihood for these $\mathbf{y}$ is just that of a multivariate normal distribution:

$$\prod_{j=1}^{N}(2\pi)^{-\hat{p}/2}\det(\boldsymbol{\Sigma}(\cdot))^{-1/2}\exp\left\{-\frac{1}{2}(\boldsymbol{y}_j-\boldsymbol{\mu})^T\boldsymbol{\Sigma}(\cdot)^{-1}(\boldsymbol{y}_j-\boldsymbol{\mu})\right\}. \tag{7}$$

Estimates of the mean vector $\boldsymbol{\mu}$ and the parameters controlling $\boldsymbol{\Sigma}(\cdot)$ can be estimated using any standard method.

For demonstration purposes in the simulations and for modelling the sea ice edge we take a Bayesian approach. We assume an exponential covariance as in Equation (3) with parameters $\boldsymbol{\sigma}$ and κ. We use the following simple prior distributions for $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ and κ. We assume a multivariate normal prior distribution for $\boldsymbol{\mu}$:

$$\boldsymbol{\mu} \sim \mathrm{MVN}(\boldsymbol{\mu_0}, \boldsymbol{\Lambda}_0). \tag{8}$$

Algorithm 2. Finding $\hat{\boldsymbol{C}}$ and $\hat{\boldsymbol{\theta}}$

Initialise $p^{(0)}$ and set $t \leftarrow 0$
Compute RHS of Equation (6)
while *Equation (6) does not hold* **do**
  Compute $\boldsymbol{\theta}^{(t)}$ for $p^{(t)}$
  Find $\hat{\boldsymbol{C}}^{(t)}$ using Equation (5) with $\boldsymbol{\theta} = \boldsymbol{\theta}^{(t)}$
  Compute LHS of Equation (6) with $\boldsymbol{C} = \hat{\boldsymbol{C}}^{(t)}$ and $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}^{(t)}$.
  if *Equation (6) does not hold* **then**
    Set $t \leftarrow t + 1$
    Set $p^{(t+1)} \leftarrow ap^{(t)}$ where $a > 1$
  else
    Set $\hat{\boldsymbol{C}} \leftarrow \hat{\boldsymbol{C}}^{(t)}$
    Set $\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}}^{(t)}$
  end
end

The hyperparameters μ_0 and $\boldsymbol{\Lambda}_0$ give the mean and covariance of the prior distribution. For the example exponential covariance defined in Equation (3) we assume uniform priors on κ and $\boldsymbol{\sigma}$:

$$\kappa \sim \text{Unif}(0, \beta_{\kappa,0}), \tag{9}$$

$$\sigma_j \sim \text{Unif}(0, \beta_{\sigma,0}), \tag{10}$$

where the hyperparameters $\beta_{\kappa,0}$ and $\beta_{\sigma,0}$ are upper bounds. Samples from the posterior distributions of the parameter can be found via standard Markov chain Monte Carlo (MCMC).

3.2.3 | Estimating gridded probabilities and credible intervals

Sampled contours can be used to estimate the probability of a given area being contained within a contour. Consider K sample contours, $\hat{\boldsymbol{S}} = (\hat{\boldsymbol{S}}_1, \ldots \hat{\boldsymbol{S}}_K)$. Each generated contour can be approximated by a binary grid, $\mathbf{G}$, of dimension $r \times v$. Let $g_{i,j}$ indicate the grid box in the ith row and jth column of $\mathbf{G}$. Let $\mathbb{1}g_{i,j,k}$ indicate whether the majority of the area in grid box $g_{i,j}$ is inside or outside contour k. Most grid boxes will be entirely inside or outside the contour; however, grid boxes that intersect the generated contour will contain area both inside and outside the contour. Ideally, the grid selected should be fine enough to ensure that little area is contained within these transitional grid boxes. Averaging the binary grids produces an $r \times v$ matrix, $\hat{\boldsymbol{P}}$, with elements $\hat{p}_{i,j} = \sum_{i=1}^{K} \mathbb{1}g_{i,j,k}/K$, that indicate the probability of grid box $g_{i,j}$ being contained within a contour. The last row of Figure 3 shows estimated gridded probabilities obtained from $K = 100$ generated contours from the corresponding GSCMs.

Credible regions for the location of the contour can be computed from $\hat{\boldsymbol{P}}$. The $(1 - \alpha)$ credible region, $\boldsymbol{I}_{1-\alpha}$, is formed from a union of grid boxes that satisfy the condition

$$\boldsymbol{I}_{1-\alpha} = \left\{ g_{i,j} : \frac{\alpha}{2} < \hat{p}_{i,j} < 1 - \frac{\alpha}{2} \right\}. \tag{11}$$

3.2.4 | Rescaling data

For numerical convenience in fitting and generating contours, it is often desirable for all contours to be contained within the $[0,1] \times [0,1]$ unit square. Observed data will typically need to be rescaled to be within these bounds. Data should be re-scaled such that generated contours do not extend outside the unit square. A good re-scaling also ensures that the contours that will be generated rarely, if ever, extend outside the unit square.

Therefore, we re-scale observed contours $\mathbf{S} = (\boldsymbol{S}_1, \ldots \boldsymbol{S}_N)$ to be within an $[\epsilon, 1-\epsilon] \times [\epsilon, 1-\epsilon]$ square. This re-scaling provides a buffer region of width ϵ on the outside of the unit square in which no contours have been observed. Therefore, if generated contours extend farther than the observed contours, they will typically go into this buffer region rather than outside the unit square. The higher the variability of contours, the larger the value of ϵ needed to avoid generating contours that go beyond the unit square.

To transform a set of observed coordinates, $\mathbf{S} = (\boldsymbol{S}_1, \ldots, \boldsymbol{S}_n)$, to the square of dimension $[\epsilon, 1-\epsilon] \times [\epsilon, 1-\epsilon]$, let $\min(\mathbf{S}_x)$ and $\max(\mathbf{S}_x)$ denote the minimum and maximum observed x-coordinates from all spatial points in all contours in $\mathbf{S}$. Define $\min(\mathbf{S}_y)$ and $\max(\mathbf{S}_y)$ analogously for the y-coordinates. Let $\delta = \max(\max(\boldsymbol{S}_x) - \min(\boldsymbol{S}_x), \max(\boldsymbol{S}_y) - \min(\boldsymbol{S}_y)))$ and let $\boldsymbol{s}_{i,j} = \left(s^x_{i,j}, s^y_{i,j}\right)$ denote the ith point in the jth.

Then for all i and all j, we shift and re-scale all coordinates for observed points $\boldsymbol{s}_{i,j}$ with the transformation

$$\tilde{s}^x_{i,j} = \epsilon + (1-2\epsilon)\left(s^x_{i,j} - \min(\mathbf{S}_x)\right)/\delta, \tag{12}$$

$$\tilde{s}^y_{i,j} = \epsilon + (1-2\epsilon)\left(s^y_{i,j} - \min(\mathbf{S}_y)\right)/\delta. \tag{13}$$

Note that Equations (12) and (13) adjust the coordinates such that all lengths $\boldsymbol{y}$'s are re-scaled by the same constant and so still follow a Gaussian distribution.

4 | COVERAGE METRIC

To assess if our probabilistic contour model performs well in representing the ice edge, a metric is needed. A good model correctly identifies the region where the contour could plausibly be located. So we focus on the coverage of prediction intervals for star-shaped contours. With an accurate contour model, the variability of the generated contour would be correctly represented along all parts of the contour. In designing an appropriate metric, we leverage the star-shaped structure of the data. The general idea is to assess coverage for each line in a line set individually.

To make this idea precise, we define several quantities illustrated in Figure 6. As in Equation (11), let $\boldsymbol{I}_{1-\alpha}$ be the $1-\alpha$ credible region obtained from some contour model. Define some test line set $\mathcal{L}^*(\boldsymbol{C}^*, \boldsymbol{\theta}^*)$ with M evenly spaced lines. Define $I_{1-\alpha,k}$ as the line segment formed from the intersection of the $\boldsymbol{I}_{1-\alpha}$ credible region and the line $\ell_k \in \mathcal{L}^*(\boldsymbol{C}^*, \boldsymbol{\theta}^*)$. We refer to $I_{1-\alpha,k}$ as a test line. Also, define $R_{i,k}$ as the intersection of some observed contour $\boldsymbol{S}_i$ and line ℓ_k. Note that $R_{i,k}$ will always be a single point when the polygon $\underline{\boldsymbol{S}}$ is exactly star-shaped and $\boldsymbol{C}^* \in \mathcal{K}(\underline{\boldsymbol{S}_i})$.

Let $W_{i,k} = \mathbb{1}[R_{i,k} \in I_{1-\alpha,k}]$ indicate whether the intersection points of the observed contour and the line $\ell_k \in \mathcal{L}^*(\boldsymbol{C}^*, \boldsymbol{\theta}^*)$ are contained within the intersection of the credible region and line ℓ_k. Then, for credible intervals with perfect coverage, for any i, k,

$$Pr(W_{i,k}) = \mathbb{E}[W_{i,k}] = 1 - \alpha. \tag{14}$$

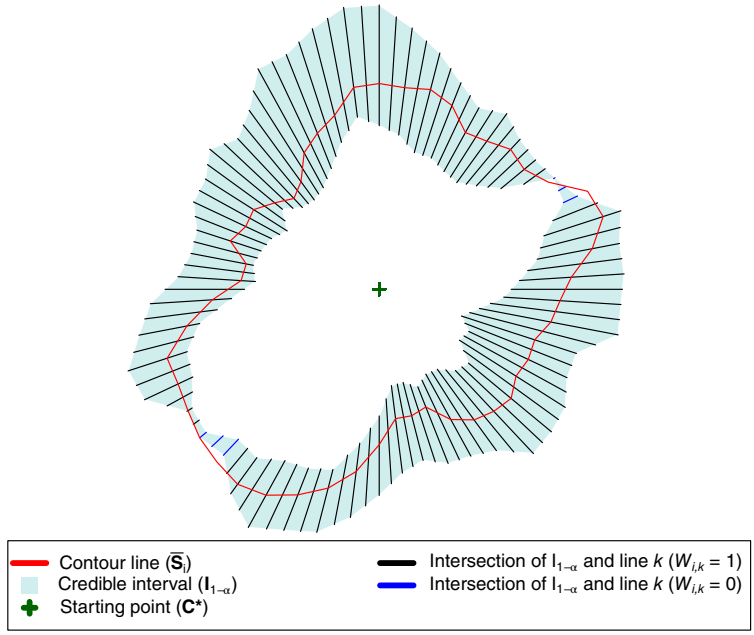

FIGURE 6 Illustration of coverage assessment for a contour line, $\bar{\boldsymbol{S}}_i$ (red), and a $1-\alpha$ credible region, $\boldsymbol{I}_{1-\alpha}$ (light blue). The line segments, $I_{1-\alpha,k}$, corresponding to the intersection of the $\boldsymbol{I}_{1-\alpha}$ credible region and line ℓ_k are coloured black when they cover $S_{i,k}$ and blue otherwise. The centre of the star-shaped polygon from which the contour is generated is denoted by a green cross sign. The line set $\mathcal{L}^*(\boldsymbol{C}^*, \boldsymbol{\theta}^*)$ contains 100 line segments evenly spaced by angle. [Colour figure can be viewed at wileyonlinelibrary.com]

In other words, the part of the true contour that intersects line ℓ_k is contained within the credible contour region along line ℓ_k with probability $1-\alpha$.

We consider coverage behaviour for a set of N observed contours, $\mathbf{S} = (\boldsymbol{S}_1, \ldots, \boldsymbol{S}_N)$ that enclose star-shaped polygons assumed to have been generated from the same model. Each $\boldsymbol{S}_i$ is assumed to be independent of $\boldsymbol{S}_j$ for all i, j.

For a credible interval with perfect coverage for any i, k,

$$\sum_{i=1}^{N} W_{i,k} \approx N(1-\alpha). \tag{15}$$

for sufficiently large sample size, N. This equation is used to assess coverage in practice. Equation (15) not holding for any k indicates the variability of the contour on test line ℓ_k is not correctly represented by the credible region.

With this set-up, $W_{i,k}$ and $W_{j,k}$ are independent for all i, j conditional on the parameters of the GSCM. So for a given line ℓ_k,

$$\sum_{i=1}^{N} W_{i,k} \sim \text{Binomial}(N, 1-\alpha). \tag{16}$$

Since the distributional behaviour of the quantity $\sum_{i=1}^{N} W_{i,k}$ is known, the expected variability of the mean coverage given a particular sample size is known. This information can be accounted for in a simulation study or cross-validation experiment.

We also consider how coverage on one test line in $\mathcal{L}^*(\boldsymbol{C}^*, \boldsymbol{\theta}^*)$ relates to coverage on another test line. The location of points on a contour are generally correlated. So, $W_{i,k}$ and $W_{i,\ell}$ are correlated. Typically $W_{i,k}$ and $W_{i,\ell}$ will be more correlated the smaller the angle distance from θ_k and θ_ℓ. What this means is that while

$$\mathbb{E}\left[\sum_{k=1}^{M} W_{i,k}\right] = M(1-\alpha), \tag{17}$$

is fixed and known, the quantity $\sum_{k=1}^{M} W_{i,k}$ is not a good metric to assess to coverage. The distribution of $\sum_{k=1}^{M} W_{i,k}$ depends on the correlation structure of the points on the contour. For a contour with high correlation among points, the quantity $\sum_{k=1}^{M} W_{i,k}$ will be substantially affected by what contour happens to be sampled. For intuition, consider a contour with high correlation among all i, j. In this case, the contour is likely to be entirely within the credible interval or entirely outside of it. So, $\sum_{k=1}^{M} W_{i,k}$ will be either 0 or M.

These relationships among coverage for M and N show that sample sizes need to be considered in terms of the number of contours observed. The metric $\sum_{i=1}^{N} W_{i,k}$ should be considered for all elements in $\boldsymbol{\theta}$. The number of test lines, M, should be set such that accuracy is assessed with detail appropriate for the application. Since the true exact value of p, is unknown, we cannot simply set $M = p$. However, based on the observed data, we should have a general idea of p. So, we set $M >> p$, to ensure that $M > p$. We also evenly space $\boldsymbol{\theta}^*$ with $\theta_1^* = (2\pi/M)/2 = \pi/M$. By assessing coverage on a substantially greater number of test lines than the true number of lines, we ensure that coverage is at least assessed near every true line.

To carry out this assessment, a fixed starting point $\boldsymbol{C}^*$ and $\boldsymbol{\theta}^*$ should be selected. In simulation studies, the true value of $\boldsymbol{C}$ will be known and we can let $\boldsymbol{C}^* = \boldsymbol{C}$. For assessment of real data such as in a cross-validation study, a starting point for $\boldsymbol{C}^*$ will be unknown and must be determined. We recommend using a $\boldsymbol{C}^*$ that minimises the difference in area between the observed contours' enclosed polygons and the star-shaped representations of the observed contours' enclosed polygons as in Section 3.2.1, that is, let $\boldsymbol{C}^* = \hat{\boldsymbol{C}}$.

This assessment approach differs from how contours have been assessed in the context of level exceedances. There, a credible region or confidence region has often been defined as the region that covers the true contour in its entirety $(1-\alpha)$-proportion of the time (e.g., Bolin & Lindgren, 2015; French, 2014). With credible (confidence) regions constructed to satisfy this definition, coverage can be assessed by determining what proportion of the time the true contour is fully contained within the region. We opt not to use this metric since our goal is to develop a method to generate ice edge contours directly. Correlation along the contour makes assessing the probability of capturing the entire contour difficult. Our metric reflects that we are most concerned with getting the right variability in all parts of the contour. We are less concerned with identifying a larger area that contains that entirety of the contour with high probability. Our intervals are therefore narrower than would be required for these global intervals.

5 | SIMULATION STUDIES

5.1 | Simulation details

Before evaluating the ice edge data, we consider how the star-shaped model performs in inferring distributions of simulated data. We consider performance with varying numbers of observations,

different constraints for the allowable mean difference in area (δ as defined in Section 3.2.1), and varied GSCM parameters.

In many of our simulations, we focus on a particular GSCM with $p = 50$ that we will refer to as *Shape A*. The correlation structure of $\boldsymbol{y}$ follows the exponential form given in Equation (3). The vector of mean distances, $\boldsymbol{\mu}$, and variance parameter vector, $\boldsymbol{\sigma}$, change gradually. The exact values of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ can be found in Appendix C. Unless otherwise noted, κ is set to 2. Example generated contours and gridded probability estimates for *Shape A* are given in the left panel of Figure 3.

The values $\hat{\boldsymbol{C}}$ and $\hat{\boldsymbol{\theta}}$ are found as described in Section 3.2. The parameter values are fit from observations using MCMC. Chains are run for 50,000 iterations with the first 15,000 iterations discarded as burn-in. The prior parameters are $\boldsymbol{\mu_0} = (0.2, \ldots, 0.2)$, $\beta_{\kappa_0} = (8, \ldots, 8)$, $\beta_{\sigma_0} = (0.15, \ldots, 0.15)$, and $\boldsymbol{\Lambda}_0 = .05\mathbf{I}_{\hat{p}}$ where $\mathbf{I}_{\hat{p}}$ is a diagonal matrix of dimension $\hat{p}$ by $\hat{p}$. The value $\hat{p}$ refers to the number of angles in $\hat{\boldsymbol{\theta}}$.

For all simulations, we use 40 evaluation runs. On each evaluation run, we estimate the GSCM parameters using N contours generated from the true GSCM as training data. From the resulting fitted GSCM, we generate 100 contours and find credible intervals as described in Section 3.2.3. To evaluate coverage, we compare the credible interval with a single 'true' contour drawn from the true GSCM. We record the coverage for a set of $M = 100$ evenly spaced test lines with $\theta_1^* = \pi/M$ as described in Section 4. We report the mean coverage over the 40 evaluations and the SD across the $M = 100$ test lines.

5.2 | Varying number of observations, N

Our first simulation varies N, the number of simulated 'observed' contours used to fit the *Shape A* GSCM. We consider coverage performance for 10, 20 and 50 simulated observed contours with $\delta = 0.02$. Table 1 displays the results of this simulation.

We find that coverage improves for $N = 20$ compared to $N = 10$. We find slightly worse performance for $N = 50$ than $N = 20$, although the performances are not substantially different given that we had only 40 evaluation runs. These results indicate that obtaining some minimum sample size is important for coverage performance. For small sample sizes, on the order of $N = 10$, the data alone may not be enough to produce accurate coverage, particularly for the 80% credible interval. However, data sets of this size can potentially still be modelled correctly if informative priors can supplement the observations.

5.3 | Varying allowable difference in area, δ

In these simulations, we evaluate how coverage accuracy is affected by the parameter δ introduced in Section 3.2.1. This parameter controls the allowable differing area when setting the number of

TABLE 1 Mean coverage values for 40 simulations of fitting the contour distribution for *Shape A* with different number of observed contours sampled as training data

Nominal	$N = 10$	$N = 20$	$N = 50$
0.80	0.87 (0.04)	0.79 (0.07)	0.76 (0.06)
0.90	0.94 (0.04)	0.89 (0.05)	0.86 (0.05)
0.95	0.98 (0.02)	0.95 (0.03)	0.93 (0.04)

Notes: In each simulation, $M = 100$ evenly spaced test lines were evaluated with $\theta_1^* = \pi/M$. SDs across the test lines are given in parentheses. Priors and Markov chain Monte Carlo details are given in Section 5.1.

TABLE 2 Mean coverage values for 40 simulations fitting the contour distribution for *Shape A* with different values of δ

		$\delta = 0.03$		$\delta = 0.02$		$\delta = 0.01$	
κ	**Nominal**	**Coverage**	**Mean $\hat{p}$**	**Coverage**	**Mean $\hat{p}$**	**Coverage**	**Mean $\hat{p}$**
1	0.8	0.86 (0.05)	38.48 (0.8)	0.87 (0.06)	45.65 (1.8)	0.86 (0.05)	55.20 (9.2)
	0.9	0.94 (0.04)	”	0.94 (0.04)	”	0.94 (0.03)	”
	0.95	0.98 (0.02)	”	0.98 (0.02)	”	0.98 (0.02)	”
2	0.8	0.82 (0.05)	32.65 (0.9)	0.80 (0.06)	41.27 (1.2)	0.84 (0.06)	50.50 (1.9)
	0.9	0.92 (0.04)	”	0.90 (0.05)	”	0.91 (0.04)	”
	0.95	0.96 (0.03)	”	0.95 (0.04)	”	0.95 (0.03)	”
4	0.8	0.86 (0.06)	28.38 (0.67)	0.79 (0.05)	36.67 (0.9)	0.80 (0.07)	48.45 (1.0)
	0.90	0.93 (0.04)	”	0.89 (0.04)	”	0.91 (0.04)	”
	0.95	0.97 (0.03)	”	0.94 (0.03)	”	0.97 (0.02)	”

Notes: In each simulation, 20 observed contours were sampled as training data and $M = 100$ evenly-spaced test lines with $\theta^* = \pi/M$ were evaluated. SDs across the test lines are given in parentheses. The mean $\hat{p}$ is given for each δ along with the SD across the evaluation runs in parentheses. Apostrophes indicate that the entry is the same as the line above it. Priors and Markov chain Monte Carlo details are given in Section 5.1.

lines used in fitting, $\hat{p}$, and how accurate $\hat{\boldsymbol{C}}$ must be. We evaluate coverage for *Shape A* with δ set to 0.03, 0.02 and 0.01. These δ selections set the allowable mean difference in area to 3%, 2% and 1% of the mean area contained within the observed contours. We also consider how correlation in $\boldsymbol{y}$ affects the need for different p by evaluating each δ for three different κ values: 1, 2 and 4. Table 2 displays the mean coverage across test lines for three α-levels along with the mean $\hat{p}$ found. On each evaluation the number of sampled contours is set to $N = 20$.

We find that the mean coverage accuracy is only modestly affected by the value of δ. These results support the idea that using a lower δ in many cases will reduce computation while not reducing model performance. We also find that, for a given δ, an increase in κ corresponds to a decrease in $\hat{p}$. In other words, for a contour with higher correlation among $\boldsymbol{y}$, a smaller set of lines can adequately represent the contour distribution.

5.4 | Varying GSCM parameters

We also evaluate contour models that have mean values, $\boldsymbol{\mu}$, that vary more slowly and more quickly than in *Shape A*. These GSCMs are denoted as *Shape B* and *Shape C*, respectively. Figure 3 shows sample contours and probability distributions for these shapes. Both models are defined to have $p = 50$ and $\kappa = 2$. Exact values for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ for *Shape B* and *Shape C* can be found in Appendix C. We report coverage results in Table 3 for three α-levels using $N = 20$ simulated observed samples and $\delta = 0.02$.

We find reasonably accurate coverage performance for all three shapes. The method performs slightly worse for *Shape C* than for *Shape A* and *Shape B*. This performance difference is likely due to the difficulty of getting the pointed sections of *Shape A* in the correct location if $\hat{p}$ is even slightly underestimated. This result indicates that for contours that look like *Shape A*, lower δ values may be appropriate. We note that the simulations in this section are relatively symmetric.

TABLE 3 Mean coverage values for 40 simulations fitting the contour distribution for Shapes A, B and C

Nominal	Shape A	Shape B	Shape C
0.8	0.79 (0.07)	0.79 (0.06)	0.87 (0.05)
0.9	0.89 (0.05)	0.89 (0.05)	0.95 (0.03)
0.95	0.95 (0.03)	0.95 (0.04)	0.98 (0.02)

Notes: In each simulation, $N = 20$ simulated observed contours were sampled as training data and $M = 100$ evenly-spaced test lines were evaluated with $\theta_1^* = \pi/M$. SDs across the test lines are given in parentheses. Priors and Markov chain Monte Carlo details are given in Section 5.1.

Results might be more variable moving around the contours if there was less symmetry. In such cases a lower δ would be appropriate to account for this heterogeneity. Overall, the good performance across parameter settings indicates that a range of contours can be well approximated by a GSCM.

6 | MODELLING CONTOURS ENCLOSING APPROXIMATELY STAR-SHAPED POLYGONS

The ice edge contours enclose polygons that are approximately, but not exactly, star-shaped. This section describes how to assess whether the GSCM is appropriate given the observed data, and how the fitting procedure is altered if the observed contours enclose polygons that are not exactly star-shaped.

6.1 | Assessing appropriateness of GSCM

Two main assumptions must be met to apply the GSCM: the polygons enclosed by the contours must be approximately star-shaped and all contours should have at least one common point. The latter assumption is needed to define a starting point and can be trivially assessed. The former assumption can be assessed using metrics that describe how close an observed contour is to enclosing a polygon that is star-shaped. These metrics focus on the difference in the area between the polygon enclosed by the true contour and the polygon enclosed by the star-shaped representation of the contour. If these differences are small for a set of observed contours, **S**, then the GSCM can be applied.

We relax Definition 8 to make precise how to approximate an arbitrary polygon with a star-shaped representation. Two main differences between star-shaped polygons and arbitrary polygons are addressed in these new definitions: (1) an arbitrary polygon may not have a kernel and (2) the contour line enclosing an arbitrary polygon may intersect with some of the lines in the line set multiple times. The new star-shaped representation definitions are:

Definition 10. Underestimated star-shaped representation, $\tilde{\boldsymbol{V}}_u(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})$: Let $\underline{\boldsymbol{S}}$ be a polygon described by ordered spatial points $\boldsymbol{S} = (s_1, \ldots, s_n)$, let $\boldsymbol{C} \in \underline{\boldsymbol{S}}$ be a starting point, let $\boldsymbol{\theta}$ be an arbitrary set of p unique angles, and let $\boldsymbol{y} = (y_1, \ldots, y_p)$ be a set of distances from $\boldsymbol{C}$ to the **closest** intersection point of the contour line $\overline{\boldsymbol{S}}$ and each line ℓ_i in the line set $\mathcal{L}(\boldsymbol{C}, \boldsymbol{\theta})$. Then, the star-shaped representation of the contour, $\boldsymbol{V}(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{y})$, is the underestimated star-shaped representation, $\tilde{\boldsymbol{V}}_u(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})$.

Definition 11. Overestimated star-shaped representation, $\tilde{\boldsymbol{V}}_o(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})$: Let $\underline{\boldsymbol{S}}$ be a polygon described by ordered spatial points $\boldsymbol{S} = (s_1, \ldots, s_n)$, let $\boldsymbol{C} \in \underline{\boldsymbol{S}}$ be a starting point, let $\boldsymbol{\theta}$ be an arbitrary set of p unique angles, and let $\boldsymbol{y} = (y_1, \ldots, y_p)$ be the set of distances from $\boldsymbol{C}$ to the **farthest** intersection point of the contour line $\overline{\boldsymbol{S}}$ and each line ℓ_i in the line set $\mathcal{L}(\boldsymbol{C}, \boldsymbol{\theta})$. Then, the star-shaped representation of the contour, $\boldsymbol{V}(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{y})$, is the overestimated star-shaped representation, $\tilde{\boldsymbol{V}}_o(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})$.

The names of these representations highlight that these polygons generally under- or overestimate the area contained within the true polygon. For contours that enclose star-shaped polygons, if $\boldsymbol{C} \in \mathcal{K}(\underline{\boldsymbol{S}})$, only one intersection is found between the contour line and all lines in the line set. So, $\tilde{\boldsymbol{V}}_o(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S}) = \tilde{\boldsymbol{V}}_u(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S}) = \tilde{\boldsymbol{V}}(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})$. For notational convenience, let $\tilde{\boldsymbol{V}}_u = \tilde{\boldsymbol{V}}_u(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})$ and $\tilde{\boldsymbol{V}}_o = \tilde{\boldsymbol{V}}_o(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})$. Then the sets that differ between the true contour and the under- and overestimated star-shaped representations are

$$A_u(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S}) := \left\{ \left(\underline{\boldsymbol{S}}^c \cap \underline{\tilde{\boldsymbol{V}}_u} \right) \cup \left(\underline{\boldsymbol{S}} \cap \underline{\tilde{\boldsymbol{V}}_u}^c \right) \right\}, \tag{18}$$

$$A_o(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S}) := \left\{ \left(\underline{\boldsymbol{S}}^c \cap \underline{\tilde{\boldsymbol{V}}_o} \right) \cup \left(\underline{\boldsymbol{S}} \cap \underline{\tilde{\boldsymbol{V}}_o}^c \right) \right\}, \tag{19}$$

where the superscript c denotes the complement of the set. Let $|A_u(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})|$ and $|A_o(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})|$ be the area contained within these sets. Figure 7 illustrates the quantities described in this section for four contours. The difference in area is zero only if the polygons are star-shaped. More precisely:

Theorem 3. *For any polygon $\underline{\boldsymbol{S}}$ that is not star-shaped $|A_u(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})| > 0$ and $|A_o(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})| > 0$ for any $\boldsymbol{C}$ and $\boldsymbol{\theta}$. (Proof in Appendix A.)*

For easier comparison of these differences in area across sets of polygons of different sizes, these areas can be expressed as percentages of the mean total area of the polygons. Table 4 reports the difference in area for the contours in Figure 7, and their star-shaped representations as a percentage of the total area of the polygon.

6.2 | Fitting GSCMs to approximately star-shaped polygons

The approach to fitting in Section 3.2 needs to be altered slightly for contours that enclose regions that are only approximately star-shaped contours. The values of $\boldsymbol{y}$ need to be computed using the under- or overestimated star-shaped approximation as given in Definitions 10 and 11. Whether to use the under- or overestimated star-shaped representation depends on the application. In some cases, asymmetric risks may motivate selecting a model that generally over- or underestimates the area within the polygon. Otherwise, both $|A_u(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})|$ and $|A_o(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})|$ can be computed for the set of observed contours $\boldsymbol{S}$ and whichever representation results in less difference in area can be selected. Fitting the posterior proceeds as in Section 3.2.2. Finding probabilities and credible intervals proceeds as in Section 3.2.3.

We find $\hat{\boldsymbol{C}}$ and $\hat{\boldsymbol{\theta}}$ using nearly the same algorithm as in Section 3.2.1, except that we update the star-shaped representation in Equations (5) and (6) to be the under- or overestimated star-shaped representation. Specifically, we replace Equation (5) with

$$\hat{\boldsymbol{C}}_u = \underset{\boldsymbol{C} \in \boldsymbol{D}}{\operatorname{argmin}} \left\{ f(|A_u(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S}_1)|, \ldots, |A_u(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S}_N)|) \right\}, \tag{20}$$

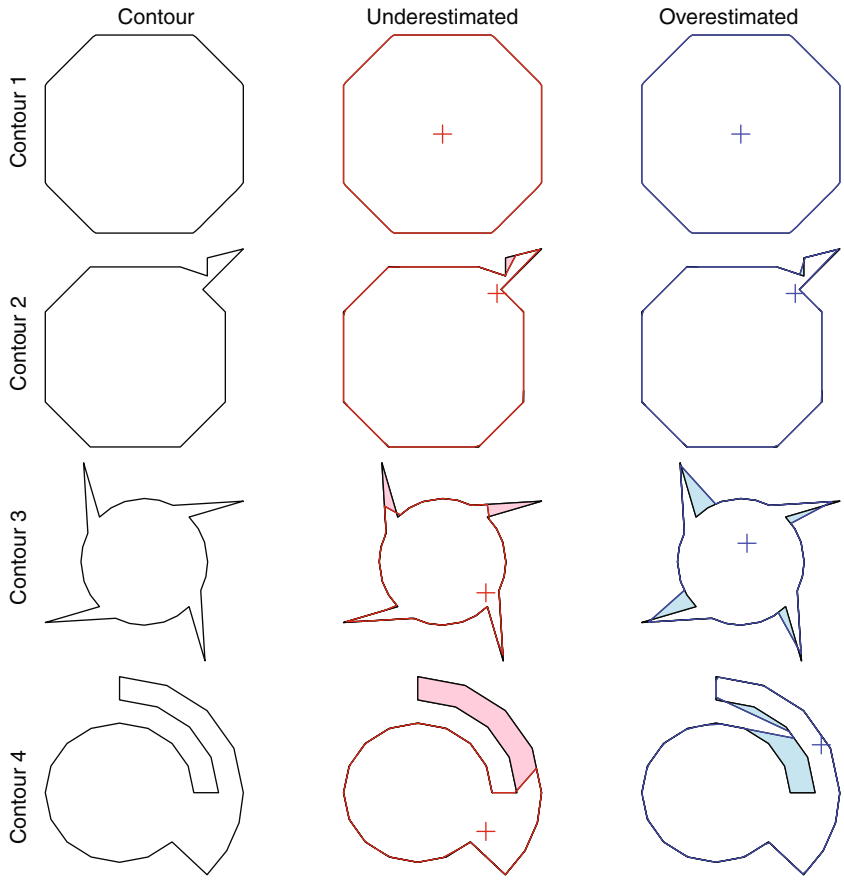

FIGURE 7 Four contours (left) with their underestimated star-shaped approximations, $\tilde{\boldsymbol{V}}_u(\hat{\boldsymbol{C}}_u, \boldsymbol{\theta}, \boldsymbol{S})$, (red, centre), and their overestimated star-shaped approximations, $\tilde{\boldsymbol{V}}_o(\hat{\boldsymbol{C}}_o, \boldsymbol{\theta}, \boldsymbol{S})$ (blue, right). The polygon in the top row is star-shaped, and the other three polygons are not. Pink and blue sections in the centre and right panel show the respective differing areas, $A_u(\hat{\boldsymbol{C}}_u, \boldsymbol{\theta}, \boldsymbol{S})$ and $A_o(\hat{\boldsymbol{C}}_o, \boldsymbol{\theta}, \boldsymbol{S})$. The red crosses in the central panel and the blue crosses in the right panel denote the estimated starting points, $\hat{\boldsymbol{C}}_u$ and $\hat{\boldsymbol{C}}_o$. The vector $\boldsymbol{\theta}$ contains 200 elements spaced evenly in the interval $[0, 2\pi]$. [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 4 The differing area for the under- and overestimated star-shaped approximations, $|A_u(\hat{\boldsymbol{C}}_u, \boldsymbol{\theta}, \boldsymbol{S})|$ and $|A_o(\hat{\boldsymbol{C}}_o, \boldsymbol{\theta}, \boldsymbol{S})|$, for the contours in Figure 7

	Underestimated	Overestimated
Contour 1	0.00	0.00
Contour 2	0.43	0.24
Contour 3	4.44	9.78
Contour 4	15.65	8.46

Notes: Differences in area are computed numerically and expressed as a percentage of the total area of the polygon. The vector $\boldsymbol{\theta}$ contains 200 elements spaced evenly in the interval $[0, 2\pi]$.

for the underestimated representation and

$$\hat{\boldsymbol{C}}_o = \underset{\boldsymbol{C}\in\boldsymbol{\mathcal{D}}}{\operatorname{argmin}} \{f(|A_u(\boldsymbol{C},\boldsymbol{\theta},\boldsymbol{S}_1)|, \ldots, |A_u(\boldsymbol{C},\boldsymbol{\theta},\boldsymbol{S}_N)|)\}, \tag{21}$$

for the overestimated representation. The values $\hat{\boldsymbol{C}}_u$ and $\hat{\boldsymbol{C}}_o$ denote the estimated starting points for the two approximations, respectively. As before, f is the mean function. Like in Section 3.2.1, we evaluate these functions of the area difference at a grid of possible locations, $\boldsymbol{\mathcal{D}}$. Unlike in Section 3.2.1 we cannot use the estimated intersection kernel to reduce the size of $\boldsymbol{\mathcal{D}}$, since the polygons are not exactly star-shaped. However, we note that the starting point must be common to all observed polygons, so we constrain $\boldsymbol{\mathcal{D}}$ to only cover areas in the intersection of all observed polygons, $\underline{\boldsymbol{S}}_1 \cap \ldots \cap \underline{\boldsymbol{S}}_N$. Equation (6) is also slightly changed so that

$$f(|A_u(\boldsymbol{C},\hat{\boldsymbol{\theta}},\boldsymbol{S}_1)|, \ldots, |A_u(\boldsymbol{C},\hat{\boldsymbol{\theta}},\boldsymbol{S}_N)|) < \frac{\delta}{N}\sum_{i=1}^{N}|\underline{\boldsymbol{S}}_i|, \tag{22}$$

for the underestimated representation and

$$f(|A_o(\boldsymbol{C},\hat{\boldsymbol{\theta}},\boldsymbol{S}_1)|, \ldots, |A_o(\boldsymbol{C},\hat{\boldsymbol{\theta}},\boldsymbol{S}_N)|) < \frac{\delta}{N}\sum_{i=1}^{N}|\underline{\boldsymbol{S}}_i|, \tag{23}$$

for the overestimated representation. The value f still represents the mean function and δ is still a proportion.

6.3 | Coverage metric for approximately star-shaped polygons

This metric introduced in Section 6.1 is essentially the same when applied to approximately star-shaped polygons. However, $R_{i,k}$, the intersection of an observed contour $\boldsymbol{S}_i$ and line ℓ_k, may now contain multiple points since polygon $\underline{\boldsymbol{S}}_i$ is only approximately star-shaped. A single point of intersection will still be more common when polygons are approximately star-shaped. Aside from this change, the definition of $W_{i,k} = \mathbb{1}[R_{i,k} \in I_{1-\alpha,k}]$ remains the same when $R_{i,k}$ is composed of multiple points. So, all subsequent definitions and properties in Section 6.1 hold as well.

6.4 | Simulation study of contours enclosing approximately star-shaped polygons

In these simulations, we assess GSCM performance for contours that enclose polygons that are only approximately star-shaped. We simulate contours that vary systematically in how much the polygons they enclose differ from being star-shaped. Figure 8 shows examples of the types of contours we will evaluate.

To obtain these contours, we first generate polygons that are star-shaped from a GSCM. We then append sections to these polygons that cause the polygons to no longer be star-shaped. The appended sections loop back around the outside of the initial polygon over some number of lines in the initial line set. The number of initial lines looped back over are selected randomly from some uniform distribution. Appended sections that loop around a larger number of lines

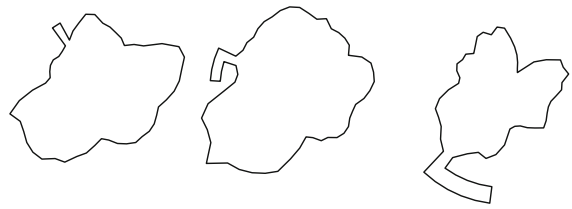

FIGURE 8 Three generated contours that enclose polygons that are only approximately star-shaped. From left to right, contours have increasing area that cannot be described with a star-shaped representation.

TABLE 5 Mean coverage values for 40 simulations of fitting the contour distribution for approximately star-shaped data

	Nominal	Unif(0, 1)	Unif(2, 3)	Unif(4, 5)
Random location	0.8	0.90 (0.04)	0.88 (0.07)	0.90 (0.05)
	0.9	0.96 (0.02)	0.94 (0.04)	0.93 (0.04)
	0.95	0.98 (0.02)	0.97 (0.02)	0.95 (0.03)
Fixed location	0.8	0.79 (0.08)	0.77 (0.12)	0.77 (0.16)
	0.9	0.88 (0.06)	0.87 (0.10)	0.87 (0.15)
	0.95	0.94 (0.04)	0.93 (0.07)	0.93 (0.13)

Notes: The number of initial lines for the appended section to loop back over are selected randomly from a Uniform(a, b) distribution. In each simulation, $N = 20$ simulated observed contours were sampled as training data and $M = 100$ evenly spaced test lines with $\theta_1^* = \pi/M$ were evaluated. The appended sections are located in a random position in the first three cases and a fixed location in the second three cases. SDs across the test lines are given in parentheses. Priors and Markov chain Monte Carlo details are given in Section 5.1.

are longer than appended sections that loop over a smaller number of lines. Longer appended sections result in more area that cannot be described with a star-shaped representation than shorter appended sections. How close the appended section is to the initial star-shaped polygon and the width of the appended section are selected randomly from uniform distributions.

We consider 40 evaluation runs for three different uniform distributions for the number of initial lines that the appended section loops back over. The initial GSCM is set to be *Shape A* with $\kappa = 2$ and $p = 50$. In each evaluation run, the GSCM is fit to $N = 20$ simulated contours. Rather than fixing δ to determine $\hat{p}$ and $\hat{\boldsymbol{C}}$ in these simulations, we set $\hat{p} = p = 50$. We then estimate $\hat{\boldsymbol{C}}$ conditional on $\hat{p}$. This choice simplifies the interpretation of our results. (Larger δ's would be needed as the area differing from a star-shaped polygon increases. So, it would be difficult to distinguish whether performance differences were due to changes in $\hat{p}$ or changes in how close to star-shaped the polygons are.) We initially run these simulations using a random location for the appended section and then repeat these simulations with a fixed location for the appended section. Results are reported in Table 5.

With a random location for the appended section, we find that applying GSCMs still results in reasonable coverage for the 90% and 95% credible intervals. For the 80% credible interval, performance is moderately degraded. Interestingly, average performance does not seem to be correlated with how large the appended section to the contour is. When an appended section is added to a fixed location, we find that the mean coverage across the test lines is relatively accurate; however, the SD is quite high, suggesting that coverage is actually poor in some parts of the contours.

FIGURE 9 Proportion of lines covered out of 40 evaluation runs plotted against the index of each of $M = 100$ evenly spaced test lines for the 90% credible interval for the contours enclosing approximately star-shaped polygons. The black line corresponds to when the appended sections are added to a random location and the blue lines corresponds to when the appended sections are added to a fixed location. The number of initial lines looped back over are selected randomly from a Uniform (4, 5) distribution. Nominal coverage is in red. Priors and Markov chain Monte Carlo details are given in Section 5.1. [Colour figure can be viewed at wileyonlinelibrary.com]

Figure 9 illustrates this variability in performance for the case with the number of initial lines looped over distributed Uniform(4, 5). We plot the proportion of evaluation runs covered by the 90% credible intervals for each test line individually.

The location of the fixed appended section is under-covered. In contrast, no obvious patterns are seen when the location of the appended section is random. These results indicate that contours that enclose polygons that modestly differ from star-shaped contours can be modelled with GSCMs. However, if areas differing from the star-shaped representation occur in the same location repeatedly, additional modelling of these areas may be needed to avoid systematic errors in coverage.

7 | APPLYING GSCMS TO THE ARCTIC SEA ICE EDGE CONTOUR

We now illustrate how GSCMs can be used to model the Arctic sea ice edge. We focus on the ice edge in September, the month when the ice-covered area in the Arctic is at its annual minimum. September holds particular interest for maritime planning, since vessel traffic is typically highest when there is the least sea ice.

We focus on modelling the sea ice edge over a short set of recent years. These types of models can be used to make statements about how probable sea ice edges will be in the near future and/or to describe plausible sea ice edges that align with the oceanic and meteorological conditions in this time period. The GSCM can be applied to generate plausible sea ice edge contours and corresponding credible intervals for the ice edge contour. Note that with climate change, the Arctic area covered by sea ice (Comiso et al., 2008; Stroeve et al., 2012) has reduced on average over time. So, observed ice edge contours from decades past are not similar to ice edges observed recently. However, on short time scales, the change in average meteorological and oceanic conditions is small. So, assuming that each observed September sea ice edge is an independent draw from a stationary distribution is appropriate. Additionally, the correlation of sea ice from 1 month to the next decays rapidly, so observations from one year apart can be treated as independent.

The data used in fitting a sea ice edge model is a monthly average observational product produced by the National Aeronautics and Space Administration satellites Nimbus-7 SMMR and DMSP SSM/I-SSMIS and downloaded from the National Snow and Ice Data Center (Comiso, 2017). The data are composed of gridded fields reporting the sea ice concentration, or percent of ice-covered area in each grid box. Following a convention used by sea ice researchers, grid boxes with at least 15% concentration are treated as containing sea ice and grid boxes with less than 15% concentration are treated as open water. This thresholding is needed because satellite estimates for very low concentrations are not considered reliable. The transition from complete sea ice cover to open water occurs over a narrow spatial range, so the ice edge is not particularly sensitive to the exact threshold selected. Each month's gridded field is converted to an ordered sequence of points as described in Section 2. We refer to the ordered sequences of points associated with each month of data as the observed ice edge in that month.

To assess the sea ice edges generated by the GSCM, we perform a leave-one-out cross-validation experiment on the September sea ice edge contour in a region in the central Arctic for 10 recent years. For each year j from 2008 to 2017, we fit the GSCM using the contours observed in the other 9 years. We then try to 'predict' the distribution of possible ice edges that would have been plausible in year j. Data have been re-scaled as described in Section 3.2.4 with

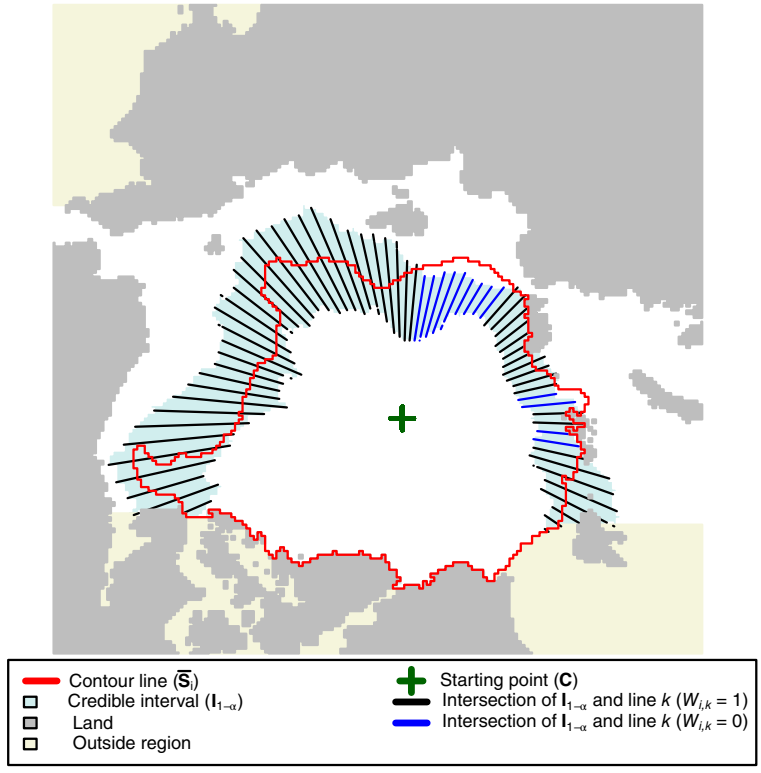

FIGURE 10 The September 2017 sea ice edge contour for the central Arctic region (red) with an 80% credible region fitted from the Gaussian star-shaped contour model with data from 2008 to 2016. Line segments, $I_{0.8,k}$, corresponding to the intersection of the $\boldsymbol{I}_{0.8}$ credible region and line ℓ_k are coloured black when they cover the contour and blue otherwise. The starting point of the evaluation is denoted by a green cross sign. Note that at the bottom of the figure, the sea ice touches the land consistently. We do not evaluate these line segments, since there is zero variance. [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 6 Mean coverage for a leave-one-out cross-validation study for the 2008–2017 September sea ice edge

Nominal	Mean
0.80	0.72 (0.13)
0.90	0.83 (0.12)
0.95	0.91 (0.09)

Notes: $M = 68$ evenly spaced test lines were evaluated with $\theta_1^* = \pi/M$. SDs across the test lines are given in parentheses.

$\epsilon = 0.1$. We exclude from fitting a section of the ice edge contours that always borders land, since there is no variability in the associated $\boldsymbol{y}$ values. We use the overestimated approximation, since for some maritime planning applications, predicting too much sea ice is less dangerous than predicting too little sea ice.

As shown in Section 5.2, 9 years of data may not be sufficient for fitting a GSCM well. However, we have considerable information that we can incorporate into the prior to augment the observations. In particular, the region we are focused often follows land boundaries, restricting where sea ice in the region could go. So, the lines in the line set $\mathcal{L} = (\ell_1, \ldots, \ell_p)$ are also bounded. We leverage their fixed maximal lengths in setting the priors. Specifically, we let $\mu_{0,i} = 0.5\ell_i$ and $\beta_{0,\sigma_i} = (\ell_i/2)/\Phi^{-1}(.995)$. The latter corresponds to the standard deviation of a normal distribution with 99% of its mass falling in the interval $(0, \|\ell_i\|)$ where $\|\ell_i\|$ is the length of ℓ_i. We also set Λ_0 be a diagonal matrix with 0.5^2 on the diagonal and $\beta_{\kappa_0} = 10$. We use MCMC for fitting with 50,000 iterations, of which 25,000 are omitted as burn-in.

We evaluate coverage as in Section 4 using the approximately optimal starting point selected using the method in Section 6.2 with $\delta = 0.03$ and $p = 100$. We exclude from evaluation test lines that always border land and have no variability. This leaves 68 test lines. We plot in Figure 10 the 80% credible interval estimated for the 2017 sea ice edge fit with data from the other nine years. As in Section 4 lines extending outward from the estimated starting point are coloured blue if they do not cover the left-out contour and black otherwise. Table 6 reports the mean coverage of the credible intervals averaged over the 10 years and 68 testing lines. Three nominal α-levels are considered. The observed and nominal coverage values are similar, suggesting that the GSCM has appropriate coverage. In other words, GSCMs perform well for capturing the variability of the sea ice edge contour.

8 | DISCUSSION

We have introduced the GSCM for modelling contours that enclose polygons that are star-shaped or approximately star-shaped and we have applied it to ice edge contours. Analysis of September Arctic sea ice showed how GSCMs can be applied to model the ice-covered area in the Arctic. Simulation studies also illustrated how GSCMs provide accurate coverage in different scenarios. We conclude this paper with a discussion of how GSCMs relate to other contour models, other methods' appropriateness in modelling the sea ice edge, and directions for future research.

8.1 | Other approaches

A large body of research addresses contours in other contexts. GSCMs are applied when multiple contour boundaries are directly observed and the distribution of possible contours is the primary

object of inference. Other methods for contours and boundaries may be appropriate for other applications.

Much of the existing research on boundaries relates to level exceedances, also called excursions. The classic level exceedance problem refers to inferring the contour enclosing regions where some latent process exceeds a certain level u. Inference is based on measurements taken at various spatial points on a random spatial field. Early work by Polfeldt (1999) considers how to make statements about the accuracy of contour maps in this context. Lindgren and Rychlik (1995) first define contour uncertainty regions using unions of crossing intervals, or line sections where transitions from below and above u occur. More recently, Bolin and Lindgren (2015) introduce a method for inferring exceedance levels with irregularly-spatial measurements when the latent spatial field is Gaussian. Their approach provides a way to make global statements about the uncertainty of the full contour. Bolin and Lindgren (2017) then extend this method to estimate the uncertainty of multiple contours to produce contour maps with appropriate uncertainty estimates. Both methods leverage Integrated Nested Laplace Approximations for efficient computation and can be implemented with the `excursions` R package (Bolin & Lindgren, 2018).

French (2014) provides an alternate simulation-based method for making global statements about the location of the contour. Methods for identifying the exceedance region are also explored from both Bayesian and frequentist perspectives (French & Hoeting, 2016; French & Sain, 2013). Level exceedance methods and GSCMs both focus on inferring contours and their uncertainty. However, in the former, boundaries and their uncertainty are inferred with measurements of a continuous process made at spatially referenced points while in the latter distributions of plausible contours are inferred from direct observations of the contour boundaries themselves.

Wombling methods also focus on spatial boundaries. First considered by Womble (1951), these methods typically apply bilinear interpolation to spatially-referenced data points. The gradients of the interpolated functions are used to infer boundaries. Jacquez et al. (2000) summarise early research on primarily deterministic methods for identifying these boundaries. Principled Bayesian statistical methods have since been developed (Banerjee & Gelfand, 2006; Gelfand & Banerjee, 2015) and recent research has introduced Wombling methods for areal data (Li et al., 2015; Lu & Carlin, 2005) and point processes (Liang et al., 2009). Wombling boundaries are inferred from spatially referenced or areal data. The proposed GSCM differs from Wombling techniques, since it is targeted to be applied to repeated directly observed contour boundaries.

Statistical shape analysis (e.g., Dryden & Mardia, 2016; Srivastava & Klassen, 2016) describes features and variation around boundaries. Shapes typically have consistent and definable features. In these types of applications, location and rotational effects are often ignored in describing the distribution around the shape. As an example, shape analysis is often applied to biological imaging research. Deformable templates were developed to describe distributions around shapes with definable features such as the parts of a hand (Amit et al., 1991; Grenander & Keenan, 1993; Grenander & Miller, 1998). A key difference between our application and those more typical of shape analysis is that the sea ice edge's physical location if of primary importance. As such, establishing correspondence (Procrustes analysis) is not appropriate. We are interested in the specific location of the ice edge on Earth, not the general shapes that form ice edges. As such, translating, rotating, or scaling observed points to align one ice edge to another does not make sense. Furthermore, no discernible features are present in the ice edge to align.

Many image analysis methods have been developed to segment images or identify edges. Mathematical morphology (Haralick et al., 1987; Lee et al., 1987), watershed segmentation (e.g., Gauch, 1999), Bayesian models (Li & Ghosal, 2017) and more recently deep learning have all been applied to identify or sharpen the uncertainty of a single observed boundary in an image.

The goals of these methods again differ from the goal of GSCM to define variability over multiple observations of boundaries.

Another alternative to GSCMs would be to model directly whether the points on a lattice are inside or outside a contour boundary. However, methods for modelling binary data on a lattice such as the autologistic (Besag, 1974), centred autologistic (Caragea & Kaiser, 2009) and the spatial generalised linear mixed model (Besag et al., 1991; Diggle et al., 1998; Hughes & Haran, 2013) are not structured to guarantee that all the grid boxes inside the contour form in a contiguous section. Hence, these methods are not designed directly for modelling contour boundaries.

Extensive research in the remote sensing community has focused on how best to translate the sea ice backscatter from Synthetic Aperture Radar into sea ice classifications. Approaches include thresholding of the backscatter coefficient for particular ice types or thickness, traditional statistical approaches and machine learning methods (see Zakhvatkina et al., 2019 for a thorough review). This research estimates the location of the ice edge given the corresponding satellite data for that time point. In other words, this research, like much of the statistical research on contours, focuses on finding the best estimate and uncertainty of individual contours at individual time points given the data at those time points. Instead, the GSCM allows one to infer the distribution of the ice edge given samples of ice edges observed at different times. This provides a distribution of plausible ice edges and information about the expected ice edge location and its variability.

Other sea ice research has examined the variability of the sea ice edge over time by parameterising the ice edge as a curve and considering deviations from the mean parameterised curve (Divine & Dick, 2006; Shapiro et al., 2003). This approach is reasonable for characterising past variability, but may not be optimal for generating a distribution of contours. Particularly if variability is high, a generated boundary may not enclose a single polygon or may intersect itself often.

8.2 | Fractal contours and GSCMs

We have treated contours as connected sequences of points, but many contours have fractal-like properties. We now discuss how a fractal contour could be converted to a connected sequence of points for modelling with a GSCM. A true fractal contour, represented by a set F, could be approximately represented by a smaller set of points $\boldsymbol{S}$. In a 2-D Euclidean space, $\mathbb{R}^2$, consider a countable or finite set $\{U\}$ of circles of radius δ. We say $\{U\}$ covers F if $F \subset \cup_{i=1}^{\infty} U_i$ and we refer to the set $\{U\}$ as a δ-cover of F. The value $N_\delta(F)$ is the smallest number of circles of radius δ that could be used to cover F (Falconer, 2004, pp. 27-28). Let $\{U^*\}$ denote one covering that contains $N_\delta(F)$ circles. Since the contour F is assumed to be finite, the number of circles in $\{U^*\}$ will also be finite. In Figure 11, we plot a δ-cover over a visualisation of a fractal contour F with several finite self-similar layers. The δ-cover plotted is for visualisation and may not contain exactly $N_\delta(F)$ circles.

We can define the elements in the sequence of points forming the contour, $\boldsymbol{S}$, to be the starting points, $\{M^*\}$, of the circles in $\{U^*\}$. The starting points should be arranged in the order in which they would be touched if one were to trace over the fractal contour line, F. Since the contour is a closed loop, where and in what direction to start tracing the fractal contour, F, only affects the indexing of the starting points and not the contour line formed by connecting these points. With this procedure the distance from any point F to a point in $\boldsymbol{S}$ is no more than δ. In general there are multiple δ-covers that contains $N_\delta(F)$ circles; therefore, a criterion needs to be specified as to which set $\{U^*\}$ is used to define $\boldsymbol{S}$. For example, the set $\{U^*\}$ could be selected to be the δ-cover with $N_\delta(F)$ circles that has the circle centre closest to the highest x- and highest y-coordinate in the domain, that is, the top right corner of the domain.

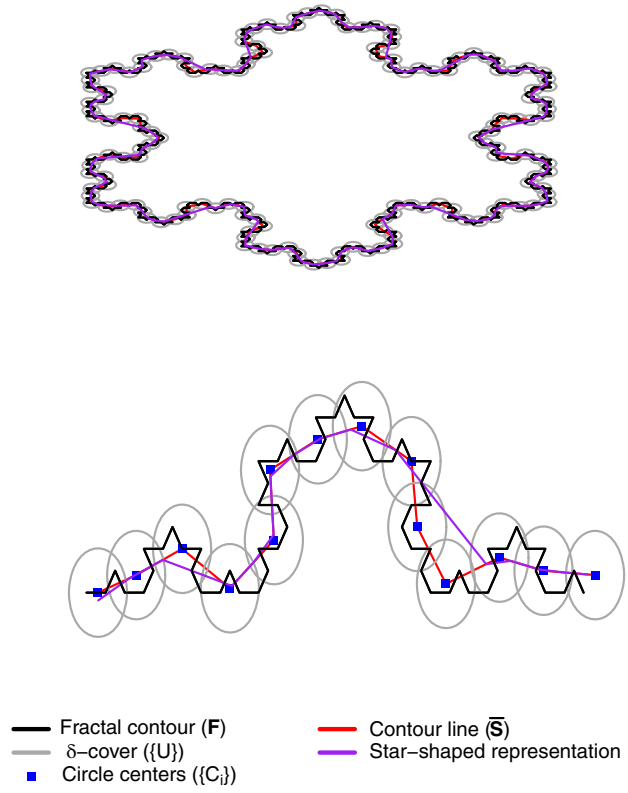

FIGURE 11 Top: A visualisation of a fractal contour, $\boldsymbol{F}$, with several self-similar layers shown (black line), a δ-cover of $\boldsymbol{F}$ (grey circles), and a line connecting the starting points of the δ-cover (red line). Bottom: Magnified section of the top figure with circle centres, $\boldsymbol{S}$, (blue squares), added. Note that the δ-cover shown is just a visualisation and may not contain $N_\delta(F)$ elements. [Colour figure can be viewed at wileyonlinelibrary.com]

Once $\boldsymbol{S}$ has been obtained, GSCMs can be used. In particular, the under- or overestimated star-shaped representation to any fractal contour can be found for $\boldsymbol{S}$ as described in Definitions 10 and 11. A star-shaped representation can represent the area contained within some fractal contours fairly well. For example, the visualised F and the overestimated star-shaped representation for $\boldsymbol{S}$ obtained using the visualised δ-cover are similar. The differing area represents only 7.4% of the total area of the visualised fractal contour when $p = 200$ lines are used in the line set.

8.3 | Limitations and extensions

A limitation of GSCMs is that they can be applied only to contours that enclose polygons that are star-shaped or approximately star-shaped. Using multiple starting points would be one way to relax this assumption. From each starting point a different star-shaped contour could be generated. These individual contours could be combined to produce an overall contour that encloses a polygon that is not star-shaped as in the 'divide and conquer strategy' of Li and Ghosal (2017). Another promising direction for relaxing the star-shaped assumption is to develop a method for appending sections to an initial contour that encloses star-shaped polygons. Extending the way contours were generated in the simulation in Section 6.4 might provide an initial approach to do this.

Another limitation is that the GSCM assumes that all generated line lengths $\boldsymbol{y}$ are drawn from a Gaussian distribution. While GSCMs can cover a range of shapes, in some applications the observed y may be skewed, bounded above, or otherwise differ from being distributed approximately normally. Therefore, exploring alternatives to GSCMs that allow for more flexible distributions of $\boldsymbol{y}$ would be valuable.

Our development of the GSCM assumes a single data source defining the contours. However, in many applications multiple data sources may provide information about the ice edge contour or sections of the ice edge contour. For example, the satellite data we used in testing GSCMs is not the only source of sea ice data. Multiple satellites operating over different locations and at different spatial and temporal scales track sea ice behaviour. Determining how to combine these data sources into a probabilistic contour model could well increase applications to which the GSCM could be applied and improve the accuracy of existing models.

All models introduced in this paper assume constant fractal properties, that is, the smoothness (roughness) of the contour is the same for the whole contour. However, smoothness can vary spatially. Extending the GSCM to account for this type of non-stationarity would broaden the range of applications. Research exploring multi-fractal star-shaped objects with locally varying parameters (Emery, & Alegría, 2020) could provide a path to making this extension.

Overall, the GSCM provides a promising avenue for modelling data composed of multiple observed contours. Representing contours with sequences of points, which are of lower dimension than spatial fields, could enable more detailed and efficient modelling of contour boundaries.

ACKNOWLEDGEMENTS

We thank the Editor, Associate Editor, and two anonymous referees for helpful comments. We thank Cecilia Bitz, Donald Percival, and Daniel Pollack for helpful discussions. Results were generated using the *ContouR* R package (Director & Raftery, 2020) and scripts accessible at https://github.com/hdirector/contourPaperScripts. Contributions by Hannah M. Director and Adrian E. Raftery were supported by NOAA's Climate Program Office, Climate Variability and Predictability Program through grant NA15OAR4310161. Hannah M. Director's contribution to this work was also based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1256082. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

DATA AVAILABILITY STATEMENT

The Bootstrap Sea Ice Concentration Data that support the findings of this study are openly available in the NASA National Snow and Ice Data Center Distributed Active Archive Center at doi.org/10.5067/7Q8HCCWS4I0R (Comiso, 2017).

ORCID


Hannah M. Director https://orcid.org/0000-0002-1874-5102

REFERENCES

Amit, Y., Grenander, U. & Piccioni, M. (1991) Structural image restoration through deformable templates. *Journal of the American Statistical Association*, 86, 376–387.

Banerjee, S. & Gelfand, A.E. (2006) Bayesian Wombling: curvilinear gradient assessment under spatial process models. *Journal of the American Statistical Association*, 101, 1487–1501.

Besag, J. (1974) Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36, 192–225.

Besag, J., York, J. & Mollié, A. (1991) Bayesian image restoration, with two applications in spatial statistics. *Annals of the Institute of Statistical Mathematics*, 43, 1–20.

Bolin, D. & Lindgren, F. (2015) Excursion and contour uncertainty regions for latent Gaussian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77, 85–106.

Bolin, D. & Lindgren, F. (2017) Quantifying the uncertainty of contour maps. *Journal of Computational and Graphical Statistics*, 26, 513–524.

Bolin, D. & Lindgren, F. (2018) Calculating probabilistic excursion sets and related quantities using excursions. *Journal of Statistical Software*, 86, 1–20.

Caragea, P.C. & Kaiser, M.S. (2009) Autologistic models with interpretable parameters. *Journal of Agricultural, Biological, and Environmental Statistics*, 14, 281.

Comiso, J. (2017) *Bootstrap sea ice concentrations from Nimbus-7 SMMR and DMSP SSM/I-SSMIS*. Version 3.

Comiso, J.C., Parkinson, C.L., Gersten, R. & Stock, L. (2008) Accelerated decline in the Arctic sea ice cover. *Geophysical Research Letters*, 35. https://doi.org/10.1029/2007GL031972.

Diggle, P.J., Tawn, J. & Moyeed, R.A. (1998) Model-based geostatistics. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 47, 299–350.

Director, H. M. & Raftery, A. E. (2020) ContouR: implementing Gaussian star-shaped contour models (GSCMs). https://github.com/hdirector/ContouR.

Divine, D.V. & Dick, C. (2006) Historical variability of sea ice edge position in the nordic seas. *Journal of Geophysical Research: Oceans*, 111. https://doi.org/10.1029/2004JC002851.

Dryden, I.L. & Mardia, K.V. (2016) *Statistical shape analysis: with applications in R*, 2nd edition. Boca Raton: John Wiley & Sons.

Emery, X. & Alegría, A. (2020) A spectral algorithm to simulate nonstationary random fields on spheres and multifractal star-shaped random sets. *Stochastic Environmental Research and Risk Assessment*, 34, 2301–2311.

Falconer, K. (2004) *Fractal geometry: mathematical foundations and applications*. Boca Raton: John Wiley & Sons.

French, J.P. (2014) Confidence regions for the level curves of spatial data. *Environmetrics*, 25, 498–512.

French, J.P. & Hoeting, J.A. (2016) Credible regions for exceedance sets of geostatistical data. *Environmetrics*, 27, 4–14.

French, J.P. & Sain, S.R. (2013) Spatio-temporal exceedance locations and confidence regions. *The Annals of Applied Statistics*, 7, 1421–1449.

Gauch, J.M. (1999) Image segmentation and analysis via multiscale gradient watershed hierarchies. *IEEE Transactions on Image Processing*, 8, 69–79.

Gelfand, A.E. & Banerjee, S. (2015) Bayesian Wombling: finding rapid change in spatial maps. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7, 307–315.

Gneiting, T. (2013) Strictly and non-strictly positive definite functions on spheres. *Bernoulli*, 19, 1327–1349.

Gneiting, T., Ševčíková, H. and Percival, D.B. (2012) Estimators of fractal dimension: assessing the roughness of time series and spatial data. *Statistical Science*, 27, 247–277.

Grenander, U. & Keenan, D.M. (1993) On the shape of plane images. *SIAM Journal on Applied Mathematics*, 53, 1072–1094.

Grenander, U. & Miller, M.I. (1998) Computational anatomy: an emerging discipline. *Quarterly of Applied Mathematics*, 56, 617–694.

Hansen, L.V., Thorarinsdottir, T.L., Ovcharov, E., Gneiting, T. & Richards, D. (2015) Gaussian random particles with flexible Hausdorff dimension. *Advances in Applied Probability*, 47, 307–327.

Haralick, R.M., Sternberg, S.R. & Zhuang, X. (1987) Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9, 532–550.

Hughes, J. & Haran, M. (2013) Dimension reduction and alleviation of confounding for spatial generalized linear mixed models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75, 139–159.

Jacquez, G.M., Maruca, S. & Fortin, M.-J. (2000) From fields to objects: a review of geographic boundary analysis. *Journal of Geographical Systems*, 2, 221–241.

Lee, D. & Preparata, F.P. (1979) An optimal algorithm for finding the kernel of a polygon. *Journal of the Association for Computing Machinery*, 26, 415–421.

Lee, J., Haralick, R. & Shapiro, L. (1987) Morphologic edge detection. *IEEE Journal on Robotics and Automation*, 3, 142–156.
Li, M. & Ghosal, S. (2017) Bayesian detection of image boundaries. *The Annals of Statistics*, 45, 2190–2217.
Li, P., Banerjee, S., Hanson, T.A. & McBean, A. (2015) Bayesian models for detecting difference boundaries in areal data. *Statistica Sinica*, 25, 385.
Liang, S., Banerjee, S. & Carlin, B.P. (2009) Bayesian Wombling for spatial point processes. *Biometrics*, 65, 1243–1253.
Lindgren, G. & Rychlik, I. (1995) How reliable are contour curves? Confidence sets for level contours. *Bernoulli*, 1, 301–319.
Lu, H. & Carlin, B.P. (2005) Bayesian areal Wombling for geographical boundary analysis. *Geographical Analysis*, 37, 265–285.
Mandelbrot, B. (1967) How long is the coast of Britain? Statistical self-similarity and fractional dimension. *Science*, 156, 636–638.
Polfeldt, T. (1999) On the quality of contour maps. *Environmetrics*, 10, 785–790.
Preparata, F.P. & Shamos, M.I. (1985) *Computational geometry: an introduction*. New York: Springer Science & Business Media.
Shamos, M. I. (1975) Geometric complexity. *Proceedings of the 7th Annual ACM Symposium on Theory of Computing*, New York: ACM, pp. 224–233.
Shapiro, I., Colony, R. & Vinje, T. (2003) April sea ice extent in the Barents Sea, 1850–2001. *Polar Research*, 22, 5–10.
Srivastava, A. & Klassen, E.P. (2016) *Functional and shape data analysis*. New York: Springer.
Stroeve, J.C., Serreze, M.C., Holland, M.M., Kay, J.E., Malanik, J. & Barrett, A.P. (2012) The Arctic's rapidly shrinking sea ice cover: a research synthesis. *Climatic Change*, 110, 1005–1027.
Womble, W.H. (1951) Differential systematics. *Science*, 114, 315–322.
Zakhvatkina, N., Smirnov, V. & Bychkova, I. (2019) Satellite SAR data-based sea ice classification: an overview. *Geosciences*, 9, 152.

How to cite this article: Director, H.M. & Raftery, A.E. (2022) Contour models for physical boundaries enclosing star-shaped and approximately star-shaped polygons. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 71(5), 1688–1720. Available from: https://doi.org/10.1111/rssc.12592

APPENDIX A. PROOFS

A.1 Proof of Theorem 1

Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)$ with $\theta_i < \theta_{i+1}$ and $\theta_i \in (0, 2\pi)$ for all i. For a star-shaped polygon $\underline{\boldsymbol{S}}$ there exists $\boldsymbol{\theta}$ and $\boldsymbol{y}$ such that $\tilde{\boldsymbol{V}}(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{y}) = \boldsymbol{S}$ for any $\boldsymbol{C} \in \mathcal{K}(\underline{\boldsymbol{S}})$.

Proof. Consider the point $\boldsymbol{s}_i \in \boldsymbol{S}$. Since $\boldsymbol{C} \in \mathcal{K}(\underline{\boldsymbol{S}})$, there exists a line segment, $\overline{\boldsymbol{C}\boldsymbol{s}_i}$, from $\boldsymbol{C}$ to $\boldsymbol{s}_i$ that is entirely contained within the polygon $\underline{\boldsymbol{S}}$. Let $\theta_i = \text{atan2}(\boldsymbol{s}_{i,y} - C_y, \boldsymbol{s}_{i,x} - C_x)$ where atan2(b, a) is the two-quadrant arctangent representing the angle between the positive x-axis and the line segment from the origin to point (b, a). Then the corresponding ℓ_i in the line set covers the line segment $\overline{\boldsymbol{C}\boldsymbol{s}_i}$. By construction, the line ℓ_i intersects $\boldsymbol{s}_i$. The line ℓ_i also cannot intersect any other points on $\overline{\boldsymbol{S}}$, since the existence of such points would violate the assumption that $\boldsymbol{S}$ is star-shaped. Define $y_i = \sqrt{(\boldsymbol{s}_{i,y} - C_y)^2 + (\boldsymbol{s}_{i,x} - C_x)^2}$, then $\boldsymbol{s}_i = \hat{\boldsymbol{v}}_i$. Repeat this construction of θ_i, ℓ_i and y_i for all $\boldsymbol{s}_i$. Then $\boldsymbol{s}_i = \hat{\boldsymbol{v}}_i$ for all i, hence $\tilde{\boldsymbol{V}}(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{y}) = \boldsymbol{S}$.

A.1.1 Proof of Corollary 1

Let ℓ_θ denote the line that extends infinitely outward from $\boldsymbol{C}$ at angle $\theta \in (0, 2\pi)$ and that intersects $\overline{\boldsymbol{S}}$. For any $\theta \in (0, 2\pi)$, the line ℓ_θ is distinct, that is, $\ell_\theta \neq \ell_{\theta'}$ for any θ, θ' such that $\theta \neq \theta'$.

Proof. Let $\theta_i < \theta < \theta_{i+1}$. The line ℓ_θ intersects $\overline{\boldsymbol{S}}$ in the line segment $\overline{\boldsymbol{s}_i\boldsymbol{s}_{i+1}}$, but not at either $\boldsymbol{s}_i, \boldsymbol{s}_{i+1}$. Since this fact holds for any θ_i, θ_{i+1}, each line ℓ_θ must be distinct.

A.2 Proof of Theorem 2

Let $\underline{\boldsymbol{S}}$ be a set of N star-shaped polygons, let $\boldsymbol{C}$ be the true starting point, and let $\hat{\boldsymbol{\mathcal{K}}}(\underline{\boldsymbol{S}}) = \mathcal{K}(\underline{\boldsymbol{S}}_1) \cap \ldots \cap \mathcal{K}(\underline{\boldsymbol{S}}_N)$ denote the intersection of the kernels of all polygons. Then, $\boldsymbol{C} \in \hat{\boldsymbol{\mathcal{K}}}(\underline{\boldsymbol{S}})$.

Proof. The starting point $\boldsymbol{C}$ will be in every $\mathcal{K}(\underline{\boldsymbol{S}}_i)$ by the definition of a kernel. So, any point p that is in $\mathcal{K}(\underline{\boldsymbol{S}}_i)$ but not $\mathcal{K}(\underline{\boldsymbol{S}}_j)$ for any i, j is not $\boldsymbol{C}$. Hence, $\boldsymbol{C}$ must be in $\hat{\boldsymbol{\mathcal{K}}}(\underline{\boldsymbol{S}})$.

A.3 Proof of Theorem 3

For any polygon $\underline{\boldsymbol{S}}$ that is not star-shaped $|A_u(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})| > 0$ and $|A_o(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})| > 0$ for any $\boldsymbol{C}$ and $\boldsymbol{\theta}$.

Proof. We show that $|A_u(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})| > 0$. The proof for $|A_o(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})| > 0$ is analogous. The quantity $|A_u(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})| = 0$ only if $(\underline{\boldsymbol{S}}^c \cap \underline{\tilde{\boldsymbol{V}}_u}) = \emptyset$ and $(\underline{\boldsymbol{S}} \cap \underline{\tilde{\boldsymbol{V}}_u}^c) = \emptyset$. These sets are both empty only if $\tilde{\boldsymbol{V}}_u = \boldsymbol{S}$. Hence, we need show only that no $\boldsymbol{\theta}$, $\boldsymbol{C}$, $\boldsymbol{y}$ combination exists that allows $\tilde{\boldsymbol{V}}_u = \boldsymbol{S}$. Polygon $\underline{\boldsymbol{S}}$ is not star-shaped. So, there exists at least one point $s^* \in \overline{\boldsymbol{S}}$ such that for any point $\boldsymbol{C} \in \underline{\boldsymbol{S}}$, the line $\overline{\boldsymbol{C}s^*}$ goes outside polygon $\underline{\boldsymbol{S}}$. Since the line $\overline{\boldsymbol{C}s^*}$ exits $\underline{\boldsymbol{S}}$ before reaching s^*, there is at least one additional intersection point between line $\overline{\boldsymbol{C}s^*}$ and contour line $\overline{\boldsymbol{S}}$. Let s^{**} denote this intersection. The lines $\overline{\boldsymbol{C}s^*}$ and $\overline{\boldsymbol{C}s^{**}}$ are at the same angle, denoted θ^*. For either s^* or s^{**} to be $\tilde{\boldsymbol{V}}_u$, we must put a a line ℓ^* in the line set that extends at angle $\theta^* = \text{atan2}(s_y^* - C_y, s_x^* - C_x) = \text{atan2}(s_y^{**} - C_y, s_x^{**} - C_x)$ where atan2(b, a) is the two-quadrant arctangent representing the angle between the positive x-axis and the line segment from the origin to point (a, b). However, any corresponding selection of y^* that results in s^* in $\tilde{\boldsymbol{V}}_u$ ensures that s^{**} is not in $\tilde{\boldsymbol{V}}_u$, since only one point in $\tilde{\boldsymbol{V}}_u$ is created for each line in the line set. Hence, there is no $\boldsymbol{y}$ that would allow $\tilde{\boldsymbol{V}}_u \neq \boldsymbol{S}$, so $|A_u(\boldsymbol{C}, \boldsymbol{\theta}, \boldsymbol{S})| > 0$.

APPENDIX B. COMPUTATION OF $\hat{\boldsymbol{\mathcal{K}}}(\underline{\boldsymbol{S}})$

Computation of $\hat{\boldsymbol{\mathcal{K}}}(\underline{\boldsymbol{S}})$ is simple. Each $\mathcal{K}(\underline{\boldsymbol{S}}_i)$ is the intersection of a set of interior half-planes with one half-plane defined by each edge $\overline{\boldsymbol{S}}_i$. For each edge, the plane is divided with the line that intersects the edge. The half-plane that is on the same side of the dividing lines as the interior of the polygon is used. The kernel of $\boldsymbol{S}_i$, $\mathcal{K}(\underline{\boldsymbol{S}}_i)$, is the intersection of all these interior half-planes (Shamos, 1975). Intersecting all the individual polygons, $\mathcal{K}(\underline{\boldsymbol{S}}_i)$, produces $\hat{\boldsymbol{\mathcal{K}}}(\underline{\boldsymbol{S}})$. For any $\underline{\boldsymbol{S}}_i$ with n edges, $\mathcal{K}(\underline{\boldsymbol{S}}_i)$ can be found as described in $O(n \log n)$ time (Shamos, 1975). An alternative algorithm can find $\mathcal{K}(\underline{\boldsymbol{S}}_i)$ in $O(n)$ time (Lee & Preparata, 1979).

APPENDIX C. SIMULATION SHAPE PARAMETERS

This section gives a list of the parameter values used in simulations. All values are rounded to the nearest thousandth. For *Shape A*, *Shape B* and *Shape C*, all the following parameters are shared.

$$\kappa = 2$$
$$\boldsymbol{C} = (0.5, 0.5)$$
$$\boldsymbol{\theta} = (0.063, 0.188, 0.314, 0.440, 0.565, 0.691, 0.817, 0.942, 1.068, 1.194, 1.319, 1.445, 1.571, 1.696, 1.822, 1.948, 2.073, 2.199, 2.325, 2.450, 2.576, 2.702, 2.827, 2.953, 3.079, 3.204, 3.330, 3.456, 3.581, 3.707, 3.833, 3.958, 4.084, 4.210, 4.335, 4.461, 4.587, 4.712, 4.838, 4.964, 5.089, 5.215, 5.341, 5.466, 5.592, 5.718, 5.843, 5.969, 6.095, 6.220)$$
$$\boldsymbol{\sigma} = (0.035, 0.044, 0.053, 0.062, 0.071, 0.080, 0.080, 0.073, 0.065, 0.058, 0.050, 0.042, 0.035, 0.035, 0.035, 0.035, 0.035, 0.035, 0.035, 0.035, 0.035, 0.035, 0.035, 0.035, 0.035, 0.035, 0.044, 0.053, 0.062, 0.071, 0.080, 0.080, 0.073, 0.065, 0.058, 0.050, 0.042, 0.035, 0.035, 0.035, 0.035, 0.035, 0.035, 0.035, 0.035, 0.035, 0.035, 0.035, 0.035, 0.035)$$

The mean parameters for *Shape A*, *Shape B* and *Shape C* are

$$\boldsymbol{\mu}_A = (0.294, 0.304, 0.306, 0.290, 0.264, 0.241, 0.220, 0.213, 0.219, 0.239, 0.259, 0.282, 0.298, 0.300, 0.276, 0.254, 0.236, 0.216, 0.211, 0.217, 0.212, 0.206, 0.200, 0.192, 0.195, 0.239, 0.287, 0.313, 0.318, 0.321, 0.322, 0.321, 0.319, 0.316, 0.312, 0.308, 0.280, 0.240, 0.197, 0.197, 0.214, 0.231, 0.247, 0.264, 0.263, 0.256, 0.249, 0.244, 0.270, 0.283)$$
$$\boldsymbol{\mu}_B = (0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3)$$
$$\boldsymbol{\mu}_C = (0.300, 0.263, 0.225, 0.188, 0.150, 0.150, 0.187, 0.225, 0.262, 0.300, 0.290, 0.243, 0.197, 0.150, 0.150, 0.200, 0.250, 0.300, 0.300, 0.274, 0.248, 0.222, 0.196, 0.170, 0.170, 0.200, 0.230, 0.260, 0.290, 0.320, 0.320, 0.298, 0.275, 0.253, 0.230, 0.150, 0.200, 0.250, 0.300, 0.350, 0.360, 0.312, 0.265, 0.218, 0.170, 0.170, 0.203, 0.235, 0.268, 0.300).$$