Journal of Statistical Software

*December 2025, Volume 115, Issue 2.* *doi: 10.18637/jss.v115.i02*

# sdmTMB: An R Package for Fast, Flexible, and User-Friendly Generalized Linear Mixed Effects Models with Spatial and Spatiotemporal Random Fields

**Sean C. Anderson**
Pacific Biological Station

**Eric J. Ward**
Northwest Fisheries Science Center

**Philina A. English**
Pacific Biological Station

**Lewis A. K. Barnett**
Alaska Fisheries Science Center

**James T. Thorson**
Alaska Fisheries Science Center

**Abstract**

Geostatistical spatial or spatiotemporal data are common across scientific fields. However, appropriate models to analyze these data, such as generalized linear mixed effects models (GLMMs) with Gaussian Markov random fields (GMRFs), are computationally intensive and challenging for many users to implement. Here, we introduce the R package **sdmTMB**, which extends the flexible interface familiar to users of **lme4**, **glmmTMB**, and **mgcv** to include spatial and spatiotemporal latent GMRFs using the stochastic partial differential equation (SPDE) approach. SPDE matrices are constructed with **fmesher**, and estimation is conducted via maximum marginal likelihood with **TMB** or via Bayesian inference with **tmbstan** and **rstan**. We describe the model and explore case studies that illustrate **sdmTMB**'s flexibility in implementing penalized smoothers, non-stationary processes (time-varying and spatially varying coefficients), hurdle models, cross-validation, and anisotropy (directionally dependent spatial correlation). Finally, we compare the functionality, speed, and interfaces of related software, demonstrating that **sdmTMB** can be an order of magnitude faster than R-**INLA**. We hope **sdmTMB** will help open this useful class of models to more geostatistical analysts.

*Keywords*: Gaussian Markov random field, GLMM, R package, spatiotemporal model, SPDE, species distribution modeling, Template Model Builder.

# 1. Introduction

Data are often collected at particular locations in space or at spatial locations over time. While such data are a rich source of information across many fields, they are challenging to properly model – data closer in space and time are usually more similar to each other than data farther apart due to measured and unmeasured variables (Cressie 1993; Diggle and Ribeiro 2007; Cressie and Wikle 2011). While measured variables can be accounted for with predictors in a model (e.g., measuring and modeling temperature effects on species abundance), unmeasured variables can cause residual spatial correlation. Accounting for this residual correlation is important because doing so allows for valid statistical inference (Legendre and Fortin 1989; Dormann *et al.* 2007), can improve predictions (e.g., Shelton *et al.* 2014), and can provide useful spatial summary statistics (e.g., Thorson 2019b; Barnett *et al.* 2021).

Geostatistical generalized linear mixed effects models (GLMMs) with spatially correlated random effects constitute an appropriate class of models for such data (Rue and Held 2005; Diggle and Ribeiro 2007; Cressie and Wikle 2011; Thorson and Kristensen 2024). Just as random intercepts can account for correlation among groups, spatial or spatiotemporal random effects can account for unmeasured variables that cause observations to be correlated in space or in both space and time. A common approach to modeling these spatial effects is with Gaussian random fields (GRFs), where the random effects describing the spatial patterning are assumed to be drawn from a multivariate normal distribution, constrained by covariance functions such as the exponential or Matérn (Cressie 1993; Chilés and Delfiner 1999; Diggle and Ribeiro 2007).

Such models quickly become computationally challenging due to the need to invert large matrices to account for covariation when evaluating the multivariate normal density function. Many solutions have been proposed, such as predictive processes (Banerjee *et al.* 2008; Latimer *et al.* 2009), the stochastic partial differential equation (SPDE) approximation to GRFs (Lindgren *et al.* 2011), and nearest-neighbor Gaussian processes (Datta *et al.* 2016; Finley *et al.* 2022). These approaches aim to minimize the scale of the covariance estimation problem while providing a means to evaluate the data likelihood, thereby allowing fitting via Bayesian (Gelfand and Banerjee 2017) or maximum likelihood methods. This can greatly improve computational efficiency (e.g., Heaton *et al.* 2019). The SPDE approach has been widely adopted, especially via the R-**INLA** package (Rue *et al.* 2009; Lindgren *et al.* 2011; Lindgren and Rue 2015) and an implementation in **TMB** (Template Model Builder, Kristensen *et al.* 2016) that relies on R-**INLA** to create input matrices (Thorson *et al.* 2015c; Thorson 2019a; Osgood-Zimmerman and Wakefield 2023; Thorson and Kristensen 2024). This SPDE method approximates a Matérn correlation function as arising mechanistically from local diffusion in space and/or time (Lindgren *et al.* 2011) and results in a sparse precision matrix that permits efficient computation using existing sparse-matrix libraries (Rue and Held 2005).

Software packages designed for specifying statistical models that incorporate the SPDE, such as R-**INLA** and **TMB**, are flexible and powerful but can be challenging for many applied researchers. For example, **TMB** requires the user to program in a C++ template, and it can be slow to experiment with multiple models when writing bespoke model code. While packages such as **lme4** (Bates *et al.* 2015) and **glmmTMB** (Brooks *et al.* 2017) let users quickly iterate and explore statistical models – focusing on evaluating fit and comparing models – they lack built-in SPDE or spatiotemporal functionality. Packages such as R-**INLA**, **inlabru**

(Bachl *et al.* 2019), and **VAST** (Thorson 2019a) are powerful user interfaces to fit spatial models that use the SPDE, but they lack a modular interface familiar to those who have used **lme4** or **glmmTMB**, lack some functionality, or may be challenging to learn for some users.

Here, we introduce the R package **sdmTMB**, which implements geostatistical spatial and spatiotemporal GLMMs using **TMB** for model fitting and **fmesher** (Lindgren 2025) or R-**INLA** to set up SPDE matrices. Our aim is not to replace the above-mentioned statistical packages, but to provide a fast, flexible, and user-friendly interface that is familiar to users of **lme4**, **glmmTMB**, or **mgcv** (Wood 2017) for a specific class of spatial and spatiotemporal models. Many individual features of **sdmTMB** may be found in other software (Table 1), but to date this full suite of useful features has not been integrated into a single package. One common application in ecology is species distribution models (SDMs), hence the package name (i.e., species distribution modelling with **TMB**), although the package is widely applicable to other fields and any geostatistical data collected continuously in space and discretely indexed in time.

This paper describes the statistical models underlying **sdmTMB** (Section 2), explains how **sdmTMB** is designed and summarizes its software functionality (Section 3), illustrates its use through three case studies (Sections 4, 5, and 6), compares **sdmTMB** to other packages (Section 7) and concludes with a discussion of links to other packages and future development directions (Section 8).

# 2. Model description

## 2.1. A spatial Gaussian random field GLMM

A GLMM with spatial Gaussian random fields can be written as

$$\begin{aligned}
\mathbb{E}[y_{\boldsymbol{s}}] &= \mu_{\boldsymbol{s}}, \\
\mu_{\boldsymbol{s}} &= g^{-1}\left(\eta_{\boldsymbol{s}}\right), \\
\eta_{\boldsymbol{s}} &= \boldsymbol{X}_{\boldsymbol{s}}\boldsymbol{\beta} + \omega_{\boldsymbol{s}},
\end{aligned}$$

where the expected value $\mathbb{E}[\cdot]$ of an observation $y$ at coordinates in space $\boldsymbol{s}$ is defined as the mean $\mu_{\boldsymbol{s}}$. That mean $\mu_{\boldsymbol{s}}$ is the result of an inverse link function $g^{-1}$ applied to a linear predictor $\eta_{\boldsymbol{s}}$. In this case, that linear predictor is the combination of a model matrix $\boldsymbol{X}_{\boldsymbol{s}}$ multiplied by a vector of coefficients $\boldsymbol{\beta}$ and a value from a spatial random field $\omega_{\boldsymbol{s}}$. This spatial random field represents the effect of latent spatial variables that are not otherwise accounted for in the model. Alternatively, $\omega_{\boldsymbol{s}}$ can be thought of as representing spatially correlated noise arising from unmodeled processes. More simply, the vector $\boldsymbol{\omega}$ represents a "wiggly" surface with an expected value of zero that is added to the linear predictor in link space (e.g., Figure 1c). The vector $\boldsymbol{\omega}$ is assumed to be drawn from a multivariate normal distribution with a covariance matrix $\boldsymbol{\Sigma}_\omega$,

$$\boldsymbol{\omega} \sim \text{MVNormal}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_\omega\right),$$

constrained by some function that defines the rate at which spatial covariance decays with distance.

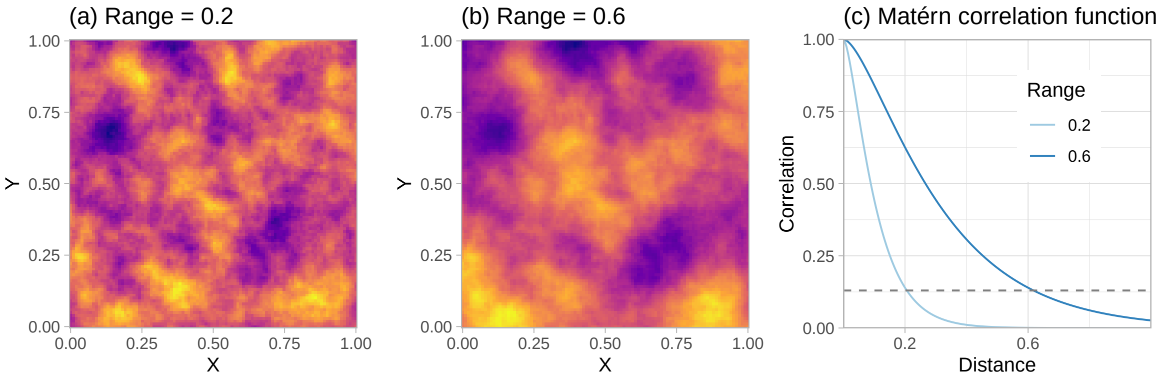


Figure 1: Example Gaussian random fields for two range values. The range describes the distance at which spatial correlation decays to ≈ 0.1 in coordinate units (i.e., the distance at which two points are effectively independent). Panel (a) shows a shorter range than panel (b), which results in a "wigglier" surface. Panel (c) shows the Matérn function for these two range values. The dashed horizontal line shows the correlation threshold of ≈ 0.1.

## 2.2. The Matérn covariance function

Various covariance functions are possible, but a popular and flexible choice is the Matérn (Whittle 1954; Matérn 1986); see Figure 1. Let $\boldsymbol{h} = \boldsymbol{s}_j - \boldsymbol{s}_k$ denote the spatial lag vector, and let $\|\boldsymbol{h}\|$ be its Euclidean norm (the distance between locations). We represent the Matérn covariance $\Phi$ as

$$\Phi\left(\boldsymbol{s}_j, \boldsymbol{s}_k\right) = \frac{\sigma_m^2}{2^{\nu-1}\Gamma(\nu)} \left(\kappa\|\boldsymbol{h}\|\right)^{\nu} K_{\nu}\left(\kappa\|\boldsymbol{h}\|\right).$$

The parameter $\sigma_m^2$ is the marginal variance (magnitude of the random field "wiggles"), $\Gamma$ represents the Gamma function, $K_\nu$ represents the modified Bessel function of the second kind, and $\kappa$ is a scale parameter. The parameter $\nu$ controls the smoothness of the covariance function. In practice, $\nu$ is challenging to estimate and is fixed here at $\nu = 1$ (Lindgren *et al.* 2011). A more interpretable derived parameter than $\kappa$ is the spatial range – the distance at which two points are nearly independent. A common definition is $\sqrt{8\nu}/\kappa$ (so if $\nu = 1$, range $= \sqrt{8}/\kappa$), which corresponds to the distance at which correlation decays to $\approx 0.1$ (Lindgren *et al.* 2011) (Figure 1c).

## 2.3. Geometric anisotropy

The assumption that correlation decays equally in all directions can be relaxed to allow for geometric anisotropy. Replacing $\|\boldsymbol{h}\|$ by $\|\boldsymbol{H}\boldsymbol{h}\|$ yields

$$\Phi\left(\boldsymbol{s}_j, \boldsymbol{s}_k\right) = \frac{\sigma_m^2}{2^{\nu-1}\Gamma(\nu)} \left(\kappa\|\boldsymbol{H}\boldsymbol{h}\|\right)^{\nu} K_{\nu}\left(\kappa\|\boldsymbol{H}\boldsymbol{h}\|\right).$$

Here, we introduce a linear transformation matrix $\boldsymbol{H}$ applied to the spatial lag $\boldsymbol{h} = \boldsymbol{s}_j - \boldsymbol{s}_k$, with two estimated parameters: the direction of the major axis of anisotropy and the ratio of the major to minor axes (Haskard 2007; Lindgren *et al.* 2011; Thorson *et al.* 2015b).

### 2.4. The SPDE approach

In practice, working with the dense covariance matrix $\boldsymbol{\Sigma}$ is computationally expensive and methods for working directly with its inverse, the precision matrix $\boldsymbol{Q}$, are more efficient ($\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1}$) (Rue and Held 2005; Simpson *et al.* 2012). One such approach is the SPDE approach, which approximates a mechanistic process of local diffusion using methods drawn from finite-element analysis. A full description of the SPDE approach is beyond the scope of this paper. Instead, we refer to the following literature: Lindgren *et al.* (2011) develop the approach. Lindgren and Rue (2015) and Bakka *et al.* (2018) summarize the SPDE approach for spatial modeling in the context of R-**INLA**. The second chapter of Krainski *et al.* (2018) provides an overview of the SPDE approach to spatial modeling with a focus on linking theory to code. Miller *et al.* (2020) summarize the approach and illustrate its equivalence to penalized smoothing approaches. Lindgren *et al.* (2022) provide a recent review of the approach and its applications over the last decade.

For a user of **sdmTMB**, the following are the important elements to understand. First, the SPDE approach links Gaussian random fields (GRFs) that have a Matérn covariance function to Gaussian Markov random fields (GMRFs) in such a way that a GMRF can be a good approximation to a GRF (Lindgren *et al.* 2011). This means that GRF models can be computationally approximated as GMRFs. By working with GMRFs, one can take advantage of theory developed to estimate their sparse precision matrix efficiently (Rue and Held 2005; Lindgren *et al.* 2011) and avoid inverting large dense matrices. Second, the SPDE approach involves piecewise linear basis functions that are defined by a triangulation over the spatial area of interest (Lindgren *et al.* 2011). Commonly, this is referred to as a "mesh" (a finite-element mesh). The properties of this mesh (e.g., its resolution and how far it extends beyond the data) affect the accuracy and computational efficiency of the SPDE approach (Lindgren *et al.* 2011). Third, working with the SPDE approach involves a precision matrix $\boldsymbol{Q}$ for the GMRF and introduces an alternative parameter $\tau$ that scales it. The precision matrix is constructed from three sparse matrices associated with the mesh ($\boldsymbol{C}$, $\boldsymbol{G}_1$, and $\boldsymbol{G}_2$) (Lindgren *et al.* 2011),

$$\boldsymbol{Q} = \tau^2 \left( \kappa^4 \boldsymbol{C} + 2\kappa^2 \boldsymbol{G}_1 + \boldsymbol{G}_2 \right),$$

where $\kappa$ is a scale parameter as before. For $\nu = 1$, the marginal variance of the Matérn random field is $\sigma_m^2 = \left(4\pi\tau^2\kappa^2\right)^{-1}$.

### 2.5. Adding spatiotemporal random fields

We can extend our spatial model to accommodate spatiotemporal data by adding Gaussian random fields for each time step $t$ (denoted $\boldsymbol{\epsilon}_t$).

$$\begin{aligned}
\mathbb{E}[y_{\boldsymbol{s},t}] &= \mu_{\boldsymbol{s},t}, \\
\mu_{\boldsymbol{s},t} &= g^{-1}\left(\boldsymbol{X}_{\boldsymbol{s},t}\boldsymbol{\beta} + \omega_{\boldsymbol{s}} + \epsilon_{\boldsymbol{s},t}\right), \\
\boldsymbol{\omega} &\sim \text{MVNormal}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_\omega\right), \\
\boldsymbol{\epsilon}_t &\sim \text{MVNormal}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_\epsilon\right).
\end{aligned}$$

In this case, we assume the spatiotemporal random fields are independent at each time step, but we could alternatively assume they are structured as a random walk or autoregressive process (demonstrated in Section 5.4). The spatiotemporal random fields represent latent variables causing spatial correlation that changes with each time step.

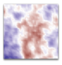
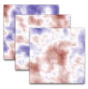
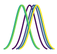
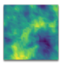
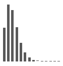
| Model component | Illustration | Description | Example | Notation | Example code |
|---|---|---|---|---|---|
| Main effects | | Linear, smoother, or breakpoint effects | Linear temperature, spline of depth, or breakpoint effect of oxygen on abundance | $\boldsymbol{X}_{\boldsymbol{s},t}^{\text{main}}\boldsymbol{\beta}$ | `formula = y ~ x`<br>`formula = y ~ s(x)`<br>`formula = y ~ breakpt(x)` |
| Spatial random effects |  | All spatially correlated effects from variables that are constant in time but are omitted from the model (or a model without a time element) | Depth, latitude, or substrate effects if omitted from model | $\omega_{\boldsymbol{s}} \sim \text{MVN}(\boldsymbol{0}, \boldsymbol{\Sigma}_\omega)$ | `spatial = ‘on’`<br>`spatial = ‘off’` |
| Spatiotemporal random effects |  | All spatially correlated effects from variables that change through time but are omitted from the model | Temperature, oxygen, prey abundance effects if omitted from the model | $\epsilon_{\boldsymbol{s},t} \sim \text{MVN}(\boldsymbol{0}, \boldsymbol{\Sigma}_\epsilon)$ | `spatiotemporal = ‘iid’`<br>`spatiotemporal = ‘ar1’`<br>`spatiotemporal = ‘rw’`<br>`spatiotemporal = ‘off’` |
| IID random intercepts | | Group-level effects that are constrained by normal distributions | Transect ID, vessel ID | $\alpha_g \sim \text{N}(0, \sigma_\alpha^2)$ | `formula = y ~ (1\|g)` |
| Time-varying effects |  | Effects that vary through time | Relationship between depth and fish abundance changes through time | $\boldsymbol{X}_{\boldsymbol{s},t}^{\text{tvc}}\gamma_t$<br>$\gamma_t \sim \text{N}(\gamma_{t-1}, \sigma_\gamma^2)$ | `time_varying = ~ 0 + x`<br>`time_varying = ~ 1` |
| Spatially varying effects |  | Effects (‘slopes’) that vary in space | (1) Local trends in abundance over time;<br>(2) spatial distribution depends on a climate index | $\boldsymbol{X}_{\boldsymbol{s},t}^{\text{svc}}\zeta_{\boldsymbol{s}}$<br>$\zeta_{\boldsymbol{s}} \sim \text{MVN}(\boldsymbol{0}, \boldsymbol{\Sigma}_\zeta)$ | `spatial_varying = ~ x` |
| Observation error |  | Error from observing or sampling the process | Counting birds or fish in a survey; recording presence/ absence in a quadrat | | `family = binomial(link = "logit")`<br>`family = nbinom2(link = "log")`<br>`family = tweedie(link = "log")` |

Figure 2: Components of an **sdmTMB** model with illustrations, descriptions, examples, notation, and example code. An **sdmTMB** model can be built from any combination of the process components (first six rows) plus an observation component (last row). The examples are from an ecology context, but the model can be fit to any spatially referenced point data. Notation: We refer to design matrices as $\boldsymbol{X}$. The indices $\boldsymbol{s}$, $t$, and $g$ index spatial coordinates, time, and group, respectively. The $\sigma$ and $\boldsymbol{\Sigma}$ symbols represent standard deviations and covariance matrices, respectively. All other symbols refer to the described model components (e.g., $\boldsymbol{\beta}$ and $\boldsymbol{\omega}$ refer to a vector of main effects and spatial random field deviations, respectively). Note that `s()` denotes a smoother as in **mgcv** (Wood 2017), `breakpt()` denotes a breakpoint "hockey-stick" shape (e.g., Barrowman and Myers 2000), `(1|g)` denotes a random intercept by group `g`, and `~0` is used in an R formula to omit the intercept.

## 2.6. Additional model components

In practice, the above models can become considerably more complex by, for example, allowing coefficients to vary through time, allowing coefficients to vary through space (Hastie and Tibshirani 1993), or adding random intercepts or slopes by grouping factors (Figure 2). All these components enter additively on the linear predictor (link) scale. Adopting the notation "main" for main effects, "tvc" for time-varying coefficients, and "svc" for spatially varying coefficients, a more complex model could be written as

$$\mu_{\boldsymbol{s},t} = g^{-1}\left(\boldsymbol{X}_{\boldsymbol{s},t}^{\text{main}}\boldsymbol{\beta} + \boldsymbol{X}_{\boldsymbol{s},t}^{\text{tvc}}\boldsymbol{\gamma}_t + \boldsymbol{X}_{\boldsymbol{s},t}^{\text{svc}}\boldsymbol{\zeta}_{\boldsymbol{s}} + \alpha_g + O_{\boldsymbol{s},t} + \omega_{\boldsymbol{s}} + \epsilon_{\boldsymbol{s},t}\right).$$

where each $\boldsymbol{X}$ represents a model matrix, $\boldsymbol{\gamma}_t$ represents a vector of coefficients that are constrained to vary through time as random walks or AR(1) processes, $\boldsymbol{\zeta}_{\boldsymbol{s}}$ represents a vector of spatially varying coefficients that follow a random field, $\alpha_g$ represents IID random intercepts by group $g$ ($\alpha_g \sim \text{Normal}\left(0, \sigma_\alpha^2\right)$), $O_{\boldsymbol{s},t}$ represents an offset variable (McCullagh and Nelder 1989, p. 206) (e.g., log sampling effort), and $\omega_{\boldsymbol{s}}$ and $\epsilon_{\boldsymbol{s},t}$ represent spatial and spatiotemporal intercept random fields as before (Figure 2). We demonstrate these model components in Sections 4, 5, and 6.

## 2.7. Delta models

So far, we have described models with one linear predictor and one family (common terminology in R packages for the combination of an observation likelihood and link). Frequently, data are better represented with two-part "delta" or "hurdle" models, which include linear predictors and observation distributions for two processes: whether observations are non-zero, and the magnitude of positive observations (Aitchison 1955). Here, we describe two types of delta models, dropping the space and time subscripts for simplicity.

### *Standard delta models*

Using $p$ to denote the probability of a non-zero observation and $r$ to represent the expected rate for positive data, we can construct two linear predictors ($\eta_1$ and $\eta_2$) in link space

$$\begin{aligned} \operatorname{logit}(p) &= \eta_1, \\ \log(r) &= \eta_2. \end{aligned}$$

We can relate $p$ and $r$ to the data via a Bernoulli distribution and a positive-valued distribution (e.g., lognormal or gamma with $\phi$ representing a generic dispersion parameter). We use $\mathrm{I}(\cdot)$ to denote an indicator function, which is 1 if $y > 0$ and 0 if $y = 0$.

$$\begin{aligned} \mathrm{I}(y > 0) &\sim \text{Bernoulli}\left(p\right), \\ y|y > 0 &\sim \text{Positive-distribution}\left(r, \phi\right). \end{aligned}$$

The expectation for a new data point is then the probability of a non-zero event multiplied by the positive rate: $p \cdot r$.

### *Poisson-link delta models*

An additional delta model is possible that has several advantages over the logit-log delta model (Thorson 2018). The primary advantage is that both linear predictors use a log link, so they can be added in link space and the partial effect of a coefficient from both can be combined. In these models, the linear predictors represent a theoretical group number $n$ and a theoretical weight (e.g., mass) per group $w$ (Thorson 2018),

$$\begin{aligned} \log(n) &= \eta_1, \\ \log(w) &= \eta_2. \end{aligned}$$

Note that the first linear predictor has a log link, *not* a logit link, and the linear predictor predicts group number $n$, *not* positive observation probability $p$. These are transformed (Thorson 2018) via

$$\begin{aligned} p &= 1 - \exp(-\exp(\log n)), \\ r &= \frac{nw}{p}. \end{aligned}$$

The first part is the complementary-log-log inverse link (McCullagh and Nelder 1989, p. 31) (which has its roots in a Poisson process), but the group number $n$ also enters into the expected positive rate $r$. These probabilities $p$ and positive rates $r$ are then used in a Bernoulli and a positive-valued distribution as before

$$\begin{aligned} \mathrm{I}(y > 0) &\sim \text{Bernoulli}\left(p\right), \\ y|y > 0 &\sim \text{Positive-distribution}\left(r, \phi\right). \end{aligned}$$

# 3. Software design and user interface

The design goal of **sdmTMB** was to develop a flexible, modular, and intuitive interface for fast maximum likelihood inference (or full Bayesian inference) with the SPDE approach to spatial and spatiotemporal GLMMs with random fields. The package gathers functionality not found combined in other packages that is particularly useful to species distribution modeling, but is applicable beyond ecology to any field that encounters geostatistical data that are continuously referenced in space and (optionally) discretely indexed by time. **sdmTMB** relies on several well-established R packages to construct and fit models (Figure 3).

The **sdmTMB** package is designed to be both modular and familiar to users of widely used R packages (e.g., **glmmTMB**, **lme4**, **mgcv**). The user starts by constructing a triangulation mesh for the SPDE approach with `make_mesh()` (Figure 3). `make_mesh()` is a wrapper for **fmesher** (Lindgren 2025) triangulation mesh functions, and users can also construct any mesh with **fmesher** or R-**INLA** and pass it to `make_mesh()`.

Fitting is accomplished with `sdmTMB()`, which has arguments similar to **glmmTMB**'s `glmmTMB()` but with additional arguments for how any spatial and spatiotemporal random fields are structured, what column defines time, any time-varying formulas, and any spatially varying formulas. Observation distributions and links are specified with standard R family functions (e.g., `binomial()`) plus several **sdmTMB**-specific families not available in the R **stats** package (e.g., `nbinom2()`, `delta_lognormal()`, see `?sdmTMB::Families`).

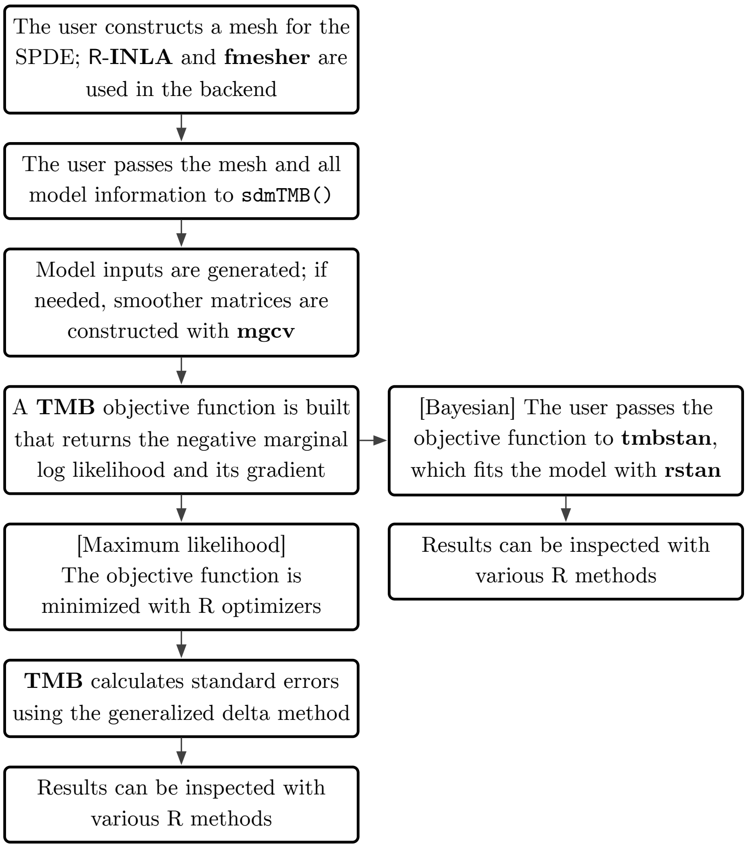

The user constructs a mesh for the SPDE; R-INLA and fmesher are used in the backend
The user passes the mesh and all model information to sdmTMB()
Model inputs are generated; if needed, smoother matrices are constructed with mgcv
A TMB objective function is built that returns the negative marginal log likelihood and its gradient
[Bayesian] The user passes the objective function to tmbstan, which fits the model with rstan
Results can be inspected with various R methods
[Maximum likelihood] The objective function is minimized with R optimizers
TMB calculates standard errors using the generalized delta method
Results can be inspected with various R methods

Figure 3: Description of the model fitting procedure in **sdmTMB**.

The **sdmTMB** formula syntax (`formula` argument) follows a standard format that is similar to **glmmTMB**, **lme4**, and **mgcv**. In addition to standard main effects, the user can include random intercepts and slopes (e.g., `+ (1 + x | group)`), threshold-shaped hockey-stick models `breakpt()` (Barrowman and Myers 2000), logistic functions `logistic()`, and penalized smoothers `s()` for generalized additive models, GAMs (Wood 2017). Penalized smoothers use the same `s()` and `t2()` syntax as in **mgcv** (Wood 2017). Supported functionality includes bivariate smoothers `s(x, y)`, smoothers varying by continuous or categorical variables `s(x1, by = x2)`, cyclical smoothers `s(x, bs = "cc")`, and smoothers with specified basis dimensions `s(x, k = 4)` (Wood 2017). Beyond the main formula, `sdmTMB()` accepts one-sided formulas for coefficients that should vary through time (`time_varying`) according to a random walk or AR(1) process (`time_varying_type`) or vary through space as random fields (`spatial_varying`).

Once the user makes a call to `sdmTMB()`, input data structures for a **TMB** model are constructed internally (Figure 3). If needed, data structures required to implement penalized smoothers are constructed using `smooth2random()` from **mgcv** (Wood 2017), as in **gamm4** (Wood and Scheipl 2025) and **brms** (Bürkner 2017). `sdmTMB()` formats data, establishes parameter starting values, and constructs an objective function with derivatives based on a compiled C++ template written for **TMB**. The objective function returns the marginal log likelihood and its gradient, integrating over random effects with the Laplace approximation (Kristensen *et al.* 2016) and efficiently using sparse-matrix computation provided by **Matrix** (Bates *et al.* 2025) as an interface to the **Eigen** library in C++. The negative marginal log likelihood is minimized via the non-linear optimization routine `stats::nlminb()` (Gay 1990) in R, with optional additional Newton-Raphson iterations (one by default) using the Hessian computed via `stats::optimHess()` (R Core Team 2025); updated parameters are accepted only if they improve the marginal log likelihood. Random effects are returned at values that maximize the likelihood conditional on the fixed effects at their maximum marginal likelihood (i.e., plug-in or empirical Bayes estimates); however, it is also possible to apply an "epsilon" estimator (Thorson and Kristensen 2016), which corrects for bias arising from the variance and skewness of random effects when calculating an estimator as a non-linear transformation of random effects (see Section 5.7 for an example). Standard errors on all parameters and derived quantities – including those involving random effects – are calculated using the generalized delta method (Kristensen *et al.* 2016; Zheng and Cadigan 2021). After rapid model exploration with maximum likelihood, one can optionally pass an **sdmTMB** model to the R package **rstan** (Carpenter *et al.* 2017; Stan Development Team 2025) via **tmbstan** (Monnahan and Kristensen 2018) to sample from the joint posterior distribution for Bayesian inference.

A fitted model summary can be viewed with `print()` or `summary()` and a set of basic "sanity" checks can be run with `sanity()`. `tidy()` returns parameter estimates in standard data frame formats similar to **broom** (Robinson *et al.* 2025). Other standard methods are also available such as `fixef()`, `confint()`, and `vcov()`.

Prediction on fitted or new data is accomplished with `predict()` (`?predict.sdmTMB`). In this paragraph, we include relevant `predict()` arguments in parentheses. The predict method is flexible and includes the option to specify a new data frame (`newdata`), whether to return predictions on the link or response scale (`type`), whether to return standard errors (`se_fit`, which can be slow if conditioned on random fields), whether to condition on the random fields (`re_form`), whether to condition on the random intercepts and slopes (`re_form_iid`), which delta model linear predictor to use (`model`), whether to take draws from the joint parameter precision matrix (`nsim`), and whether to use MCMC samples from a **tmbstan** model fit (`mcmc_samples`).

A variety of model evaluation tools are available. A `residuals()` method calculates various types of residuals (`?residuals.sdmTMB`). The default is a form of randomized quantile (Dunn and Smyth 1996) or probability integral transform (Smith 1985) residuals. New observations can be simulated from a fitted model with `simulate()` (`?simulate.sdmTMB`) or data can be simulated from scratch (without fitting a model) with `simulate_new()`. `sdmTMB_cv()` facilitates cross-validation.

**sdmTMB** models can include penalized likelihoods by assigning priors (penalties) to model parameters through the `sdmTMB() prior` argument (`?sdmTMBpriors`). These priors may be useful in cases where estimation is difficult because of identifiability issues or relatively flat likelihood surfaces, or to impart prior information or to achieve regularization. Following other recent SPDE implementations in **TMB** (Breivik *et al.* 2021; Osgood-Zimmerman and Wakefield 2023), penalized complexity (PC) priors (Simpson *et al.* 2017; Fuglstad *et al.* 2019) (`?pc_matern`) can constrain the spatial range and variance parameters. These priors or penalties are available both with maximum likelihood estimation and with MCMC sampling. If a model will be passed to **tmbstan** with priors, the `bayesian` logical argument should be set to `TRUE` to enable Jacobian adjustments for change of variables (priors applied to parameters that are internally constrained or transformed, Carpenter *et al.* 2017).

### 3.1. Installation

**sdmTMB** can be installed from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=sdmTMB`

```
R> install.packages("sdmTMB")
```

Users who wish to automatically install suggested packages as well may wish to use

```
R> install.packages("sdmTMB", dependencies = TRUE)
```

As an alternative to the CRAN version, the development version can be installed with

```
R> install.packages("pak")
R> pak::pak("sdmTMB/sdmTMB", dependencies = TRUE)
```

Development code is hosted at `https://github.com/sdmTMB/sdmTMB`.

Additional utilities, which require heavier package dependencies (such as R-**INLA** and **rstan**) and are used by only a subset of users, are maintained in the **sdmTMBextra** package at `https://github.com/sdmTMB/sdmTMBextra`

Users willing to replace the default R **BLAS** (Basic Linear Algebra Subprograms, Blackford *et al.* 2002) library with an optimized version (e.g., **openBLAS**, Xianyi *et al.* 2012) can expect up to an order of magnitude increase in model fitting speed for complex models. Suggestions are included in the package README file.

# 4. Example: Spatial species distribution modeling

We begin with a simple species distribution model of encounter probability of Pacific Cod (*Gadus macrocephalus*) from a trawl survey conducted in Queen Charlotte Sound, British Columbia, Canada. The purpose of our example is to illustrate the need for spatial random fields. This survey is conducted by Fisheries and Oceans Canada and follows a depth-stratified random sampling design, resulting in a georeferenced dataset. The data frame `pcod` is available as package data in **sdmTMB**. Relevant columns include latitude, longitude, Universal Transverse Mercator (UTM) coordinates, bottom depth, and encounter (`present = 1`) vs. non-encounter (`present = 0`) of Pacific Cod for a given survey sample.

```
R> library("sdmTMB")
R> library("dplyr")
R> library("ggplot2")
R> select(pcod, lat, lon, X, Y, depth, present)
```

```
# A tibble: 2,143 x 6
    lat   lon     X     Y depth present
  <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
1  52.3 -130.  446. 5793.   201       1
2  52.3 -130.  446. 5800.   212       1
3  52.4 -130.  449. 5802.   220       0
# i 2,140 more rows
```

## 4.1. Adding UTM columns

An **sdmTMB** model requires a data frame that contains a response column, columns for any predictors, and columns for spatial coordinates. Usually it makes sense to convert the spatial coordinates to a projected coordinate system such as UTM so distances are approximately preserved within the study region (e.g., using `sf::st_transform()`, Pebesma 2018). Here we use the helper function `add_utm_columns()` to add UTM coordinates with km units (so our estimated spatial range parameter is not too big or small). By default, the function will guess the UTM zone and create new columns `X` and `Y`. Since our example data already has these UTM columns, we can skip running this code.

```
R> pcod <- add_utm_columns(pcod, c("lon", "lat"), units = "km")
```

## 4.2. SPDE mesh creation

We then create a mesh object that contains triangulation and projection matrices needed to apply the SPDE approach using `make_mesh()`. The argument `cutoff` defines the minimum allowed distance between mesh vertices in the units of `X` and `Y` (km). We could create a basic mesh specifying this:

```
R> mesh_pcod <- make_mesh(pcod, xy_cols = c("X", "Y"), cutoff = 8)
```

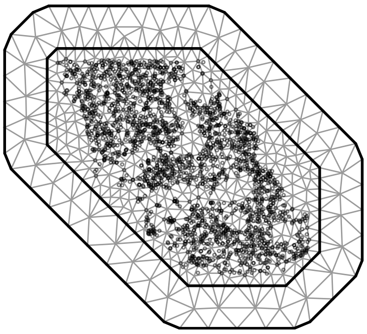


Figure 4: SPDE mesh (lines) combined with the trawl survey observations (points). The locations where lines intersect are referred to as "vertices" or "knots". Finer meshes will be slower to fit but generally increase the accuracy of the SPDE approximation, to a point. A greater degree of control over the mesh construction can be achieved by using **fmesher** or R-**INLA** directly and supplying the object to `make_mesh()`.

We can also specify additional arguments, in this case passed to `fmesher::fm_mesh_2d_inla()`: maximum triangle edge (`max.edge`) lengths of 10 km and 40 km for the inner and outer mesh, respectively, and an `offset` width of 10 km and 40 km for the inner and outer mesh borders. For more irregularly shaped areas, we could have specified a non-convex hull with `convex` and `concave` arguments. The triangle edge length should be several times smaller than the range and the outer boundary should extend at least as far as the range to avoid edge effects. See Krainski *et al.* (2018, Chapters 2.6 and 2.7) for guidance on mesh construction.

We can visualize the mesh object with the associated plotting method (Figure 4). Our mesh has 563 (`mesh_pcod2$mesh$n`) vertices. Mesh complexity has a large influence on the speed of fitting these models. Meshes generated with **fmesher** may differ slightly across CPU architectures and compiler settings due to platform-specific floating-point arithmetic, which can lead to minor differences in parameter estimates. Results shown here were produced with R 4.5.2 on Ubuntu 22.04.5 LTS (x86_64-pc-linux-gnu).

```
R> mesh_pcod2 <- make_mesh(pcod, xy_cols = c("X", "Y"),
+    fmesher_func = fmesher::fm_mesh_2d_inla,
+    cutoff = 8, max.edge = c(10, 40), offset = c(10, 40))
R> plot(mesh_pcod2)
```

## 4.3. Fitting the model

We will fit a logistic regression of encounter probability with and without spatial random fields to illustrate the importance of accounting for spatial correlation. In addition to the spatial random field, we include an intercept and a quadratic effect of depth on the probability of encounter. Our model can be written as

$$
\begin{aligned}
y_{\boldsymbol{s}} &\sim \operatorname{Bernoulli}\left(\mu_{\boldsymbol{s}}\right), \\
\mu_{\boldsymbol{s}} &= \operatorname{logit}^{-1}\left(\boldsymbol{X}_{\boldsymbol{s}}^{\text{main}}\boldsymbol{\beta} + \omega_{\boldsymbol{s}}\right), \\
\boldsymbol{\omega} &\sim \operatorname{MVNormal}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_{\omega}\right),
\end{aligned}
$$

where $\boldsymbol{X}_{\boldsymbol{s}}^{\text{main}}$ represents a matrix of main effect covariates (intercept and quadratic effects of depth), $\boldsymbol{\beta}$ represents a vector of estimated main effect coefficients, and $\omega_{\boldsymbol{s}}$ represents the estimated spatial random field at location $\boldsymbol{s}$.

We can implement this model with `sdmTMB()`:

```
R> fit_bin_rf <- sdmTMB(present ~ poly(log(depth), 2), data = pcod,
+    mesh = mesh_pcod2, spatial = "on", family = binomial(link = "logit"))
```

We can also fit a version that omits the spatial random field by setting `spatial = "off"`. We will use the `update()` method to refit the model while updating any specified arguments:

```
R> fit_bin <- update(fit_bin_rf, spatial = "off")
```

## 4.4. Inspecting the model

We can run some basic checks on our model with the `sanity()` function:

```
R> sanity(fit_bin_rf)

✓ Non-linear minimizer suggests successful convergence
✓ Hessian matrix is positive definite
✓ No extreme or very small eigenvalues detected
✓ No gradients with respect to fixed effects are >= 0.001
✓ No fixed-effect standard errors are NA
✓ No fixed-effect standard errors look unreasonably large
✓ No sigma parameters are < 0.01
✓ No sigma parameters are > 100
✓ Range parameter doesn't look unreasonably large
```

This does not flag any issues. `sanity()` is checking that the `nlminb()` optimizer reported successful convergence, that the Hessian matrix is positive definite, that no extreme or small eigenvalues are detected, that absolute values of log-likelihood gradients with respect to fixed effects are all $< 0.001$, that no fixed-effect standard errors are unreasonably large ($< 100$ by default), that random field marginal standard deviations are not unexpectedly small ($< 0.01$) or large ($> 100$), and that the random field Matérn range parameter does not look unreasonably large ($\geq 1.5$ times the largest distance from a bounding box around the observations).

We can get a summary of our model fit:

```
R> summary(fit_bin_rf)

Spatial model fit by ML ['sdmTMB']
Formula: present ~ poly(log(depth), 2)
Mesh: mesh_pcod2 (isotropic covariance)
Data: pcod
Family: binomial(link = 'logit')
```

```
Conditional model:
                     coef.est coef.se
(Intercept)             -0.44    0.42
poly(log(depth), 2)1   -75.72    9.30
poly(log(depth), 2)2  -109.79    9.27

Matérn range: 41.43
Spatial SD: 1.69
ML criterion at convergence: 1034.744

See ?tidy.sdmTMB to extract these values as a data frame.
```

The output indicates our model was fit by maximum (marginal) likelihood (ML). We also see the formula, mesh, fitted data, and family. Next, we see any estimated main effects, the Matérn range distance, the spatial random field standard deviation, and the negative log likelihood at convergence.

We can use the tidy() function to obtain a data frame with parameter estimates (standard methods such as fixef(), confint(), and vcov() are also available). The standard errors on our fixed effects have increased with the spatial random field:

```
R> tidy(fit_bin_rf, conf.int = TRUE)

# A tibble: 3 x 5
  term                 estimate std.error conf.low conf.high
  <chr>                   <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept)            -0.439     0.424    -1.27     0.393
2 poly(log(depth), 2)1  -75.7       9.30    -94.0    -57.5
3 poly(log(depth), 2)2 -110.        9.27   -128.     -91.6

R> tidy(fit_bin, conf.int = TRUE)

# A tibble: 3 x 5
  term                 estimate std.error conf.low conf.high
  <chr>                   <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept)            -0.426    0.0573   -0.538    -0.314
2 poly(log(depth), 2)1  -31.7      3.03    -37.6     -25.8
3 poly(log(depth), 2)2  -66.9      4.09    -74.9     -58.9
```

By setting effects = "ran_pars", tidy() will return random field parameters, where sigma_O is the marginal standard deviation of the spatial random field $\boldsymbol{\omega}$ (“O” for “Omega”).

```
R> tidy(fit_bin_rf, effects = "ran_pars", conf.int = TRUE)

# A tibble: 2 x 5
  term    estimate std.error conf.low conf.high
  <chr>      <dbl>     <dbl>    <dbl>     <dbl>
1 range      41.4      9.38     26.6      64.6
2 sigma_O     1.69     0.233     1.29      2.22
```

### 4.5. Checking the effect of including a random field

We can test for spatial autocorrelation with a visual inspection or a statistical test of the residuals. Here, we demonstrate an approach using an implementation of Moran's I from the **ape** package (Gittleman and Kot 1990; Paradis and Schliep 2019). We set `type = "mle-mvn"` to denote setting fixed effects at their maximum likelihood estimate (MLE) but taking a single draw from the approximate (multivariate normal) distribution of the random effects (Waagepetersen 2006; Thygesen *et al.* 2017).

```
R> inv_dist_matrix <- 1 / as.matrix(dist(pcod[, c("X", "Y")]))
R> diag(inv_dist_matrix) <- 0
R> set.seed(1)
R> r_bin <- residuals(fit_bin, type = "mle-mvn")
R> set.seed(1)
R> r_bin_rf <- residuals(fit_bin_rf, type = "mle-mvn")
R> ape::Moran.I(r_bin, weight = inv_dist_matrix)$p.value

[1] 0

R> ape::Moran.I(r_bin_rf, weight = inv_dist_matrix)$p.value

[1] 0.8817066
```

We see strong evidence for spatial autocorrelation for the model without a random field ($p < 0.01$) but a lack of evidence for spatial correlation after including a random field, suggesting that residual spatial autocorrelation is alleviated by including the random field. The specific p-value is dependent on the seed due to the randomization in the randomized quantile residuals.

We can also see that the marginal Akaike information criterion (AIC) (Akaike 1974) of the model with spatial random fields is lower:

```
R> AIC(fit_bin_rf, fit_bin)

           df      AIC
fit_bin_rf  5 2079.487
fit_bin     3 2392.070
```

Caution is warranted in performing model selection via marginal AIC for models involving penalized smoothing (spatial or otherwise) (e.g., Greven and Kneib 2010; Säfken *et al.* 2021). In marginal AIC calculation, the degrees of freedom are based on the number of fixed effects and do not account for the degree of random effect penalization as conditional AIC would. Methods to estimate effective degrees of freedom for similar models were recently demonstrated in Thorson (2024) and are included in the function `cAIC()`.

### 4.6. Comparing models with cross-validation

As an alternative to AIC, we can conduct model comparison with cross-validation. **sdmTMB** includes the helper function `sdmTMB_cv()` to facilitate this. We will do 10-fold cross-validation with the folds constructed randomly. We will set the seed each time to ensure the folds are consistent. Using the `fold_ids` argument, we could supply our own folds and conduct spatially blocked cross-validation (Roberts *et al.* 2017). If we set a parallel plan with the **future** package (Bengtsson 2021), our folds will be fit in parallel.

```
R> library("future")
R> plan(multisession)
R> set.seed(12928)
R> cv_bin_rf <- sdmTMB_cv(present ~ poly(log(depth), 2), data = pcod,
+    mesh = mesh_pcod, spatial = "on", family = binomial(), k_folds = 10)
R> set.seed(12928)
R> cv_bin <- sdmTMB_cv(present ~ poly(log(depth), 2), data = pcod,
+    mesh = mesh_pcod, spatial = "off", family = binomial(), k_folds = 10)
```

We can then calculate any performance metric of interest for comparison. A common metric is the log score or log predictive density (lpd) of the left-out data (Geisser and Eddy 1979; Vehtari *et al.* 2017),

$$\text{lpd} = \sum_{i=1}^{n} \log \mathcal{L}(y_i|\hat{y}_i),$$

where $n$ represents the number of left-out data points, $\log \mathcal{L}$ denotes the log likelihood, $y_i$ represents left-out data point $i$, and $\hat{y}_i$ represents the prediction for left-out data point $i$ with the fixed effects at their MLEs and random effects at their empirical Bayes estimates.

Indeed, the log predictive density for the left-out data is considerably higher for the model that includes random fields, indicating better out-of-sample predictions:

```
R> cv_bin_rf$sum_loglik

[1] -995.4446

R> cv_bin$sum_loglik

[1] -1196.493
```

In practice, we would repeat this procedure several times to ensure the rank order is insensitive to the randomly chosen folds and, if it is sensitive, consider averaging across multiple folds or increasing the number of folds.

### 4.7. Making predictions on new data

To visualize our model, we can make predictions with the `predict()` method (`?predict.sdmTMB`) and optionally use the `newdata` argument to predict on a new data frame containing locations and all predictor columns. Here, we will predict on a 2 × 2 km grid (`qcs_grid`) that covers the entire region of interest so we can visualize the predictions spatially. The grid contains spatial covariate columns and all predictors used in the model set to values for which we want to predict. Some covariates might be fixed at a specified value for all predictions, such that we are predicting the expected value for samples conditional on

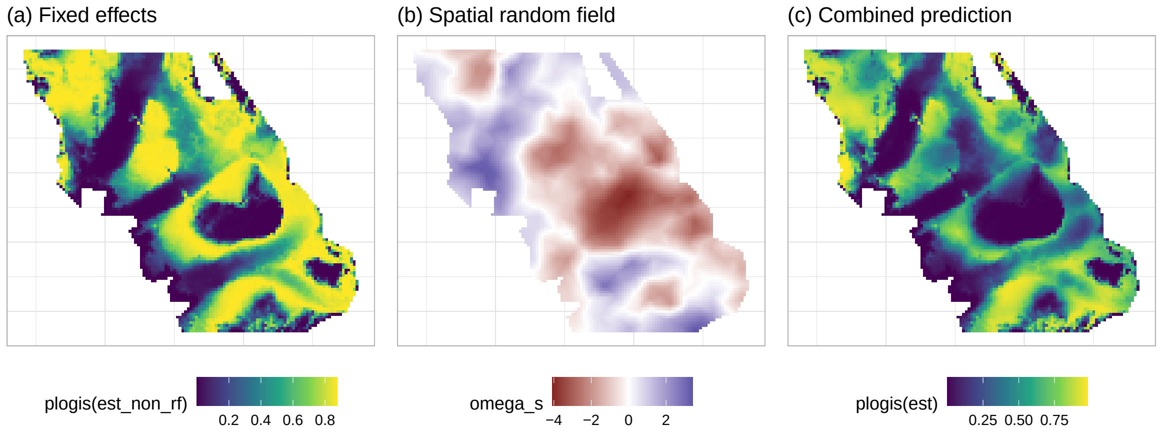


Figure 5: Prediction components from the binomial species distribution model of Pacific Cod. Shown are (a) the quadratic effect of bottom depth, (b) the spatial random field in link (logit) space, and (c) the overall prediction, which here is the combination of panels a and b. The spatial random field represents spatially correlated latent effects not accounted for by the fixed effects. Note the difference between predictions from depth alone (a) and predictions including a spatial random field (c).

those specified values. In the context of fish or animal surveys, these are sometimes called *detectability* or *catchability* covariates, given the model is predicting the target variable while controlling for the additional influence of these covariates (Thorson 2019a). The output of `predict()` is a data frame containing overall estimates in link space (`est`), estimates from the non-random-field components (`est_non_rf`; here, intercept and depth), and estimates from the individual random field components (here, `omega_s` – the spatial field). We can plot these with `geom_raster()` or `geom_tile()` from the **ggplot2** (Wickham 2016) package (Figure 5).

```
R> p <- predict(fit_bin_rf, newdata = qcs_grid)
R> select(p, X, Y, depth, est, est_non_rf, omega_s) |>
+    as_tibble() |>
+    head(n = 2)

# A tibble: 2 x 6
      X     Y depth    est est_non_rf omega_s
  <dbl> <dbl> <dbl>  <dbl>      <dbl>   <dbl>
1   456  5636  347. -6.52       -6.23  -0.291
2   458  5636  223. -0.861      -0.600 -0.261
```

## 5. Example: Spatiotemporal species distribution modeling

As a second example, we will construct a spatiotemporal model of catch rates of Pacific Spiny Dogfish (*Squalus suckleyi*) from a trawl survey off the west coast of Vancouver Island, Canada. This example extends the spatial model described above by including (1) spatiotemporal fields, allowing unique spatially correlated latent effects each year; (2) a time-varying intercept as an AR(1) process, allowing year effects to vary but remain autocorrelated; (3) a smooth effect of depth, allowing catch rates to vary non-linearly with depth; and (4) spatial anisotropy, allowing spatial correlation to be directionally dependent. Since catch rates are positive, continuous, and contain zeros, we begin by specifying the response family as a Tweedie distribution (Tweedie 1984) with a log link. We then compare alternative families, spatiotemporal random field structures, and the exclusion of anisotropy to illustrate the flexibility of **sdmTMB**.

## 5.1. Adding UTM columns and creating a mesh

The dataset includes spatial coordinates, year, dogfish catch weight in kg, area swept in km$^2$, and bottom depth:

```
R> dat <- select(dogfish, lon = longitude, lat = latitude, year,
+    catch_weight, area_swept, depth)
R> dat

# A tibble: 1,458 x 6
    lon   lat  year catch_weight area_swept depth
  <dbl> <dbl> <int>        <dbl>      <dbl> <dbl>
1 -125.  48.7  2004         22.7      0.103    73
2 -126.  48.2  2004          0        0.103   455
3 -126.  48.3  2004          0        0.116   171
# i 1,455 more rows
```

We add UTM zone 9 columns, create a log depth column for convenience, and create a basic mesh:

```
R> dat <- add_utm_columns(dat, c("lon", "lat"),
+    units = "km", utm_crs = 32609)
R> dat$log_depth <- log(dat$depth)
R> mesh <- make_mesh(dat, xy_cols = c("X", "Y"), n_knots = 200)
```

## 5.2. Fitting the model

We can then specify our model. We include an offset (McCullagh and Nelder 1989, p. 206) for the effort variable (log area swept) such that we are effectively modeling density and our predictions will be for an area swept of 1 km$^2$.

Our model can be written as

$$\begin{aligned}
\mathbb{E}[y_{\boldsymbol{s},t}] &= \mu_{\boldsymbol{s},t}, \\
\mu_{\boldsymbol{s},t} &= \exp\left(\boldsymbol{X}^{\mathrm{main}}_{\boldsymbol{s},t}\boldsymbol{\beta} + O_{\boldsymbol{s},t} + \boldsymbol{X}^{\mathrm{tvc}}_{\boldsymbol{s},t}\gamma_t + \omega_{\boldsymbol{s}} + \epsilon_{\boldsymbol{s},t}\right), \\
\boldsymbol{\omega} &\sim \mathrm{MVNormal}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_\omega\right), \\
\boldsymbol{\epsilon}_t &\sim \mathrm{MVNormal}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_\epsilon\right),
\end{aligned}$$

where $\boldsymbol{\beta}$ are coefficients associated with the main effects, $O_{\boldsymbol{s},t}$ represents the offset (here, log area swept), $\gamma_t$ represents the time-varying coefficients, $\omega_{\boldsymbol{s}}$ is a value from a spatial field (representing constant latent spatial effects), and $\epsilon_{s,t}$ is a value from a spatiotemporal field (representing latent spatial effects that vary by year). The temporally varying intercepts $\gamma_t$ are modeled as a stationary AR(1) process,

$$\gamma_{t=1} \sim \text{Normal}\left(0, \sigma^2_\gamma\right),$$
$$\gamma_{t>1} \sim \text{Normal}\left(\rho_\gamma \gamma_{t-1}, (1-\rho^2_\gamma)\sigma^2_\gamma\right),$$

where $\rho_\gamma$ represents the correlation between intercepts $\gamma$ at time $t-1$ and $t$, and $\sigma^2_\gamma$ represents the marginal variance of this process. We include the argument `extra_time`, which represents all time steps to include in the latent process (and all time steps for which we may wish to predict), such that autoregressive processes are applied to equally spaced annual time steps.

We can fit this model as:

```
R> fit_tw <- sdmTMB(catch_weight ~ s(log_depth), data = dat, mesh = mesh,
+    family = tweedie(), offset = log(dat$area_swept), time = "year",
+    time_varying = ~ 1, time_varying_type = "ar1", spatial = "on",
+    spatiotemporal = "iid", anisotropy = TRUE,
+    extra_time = seq(min(dat$year), max(dat$year)), silent = FALSE)
```

### 5.3. Exploring delta model alternative families

We next explore four alternative families that may better represent the data. Each alternative family uses a delta model formulation as described in Section 2.7.

```
R> fit_dg <- update(fit_tw, family = delta_gamma())
R> fit_dl <- update(fit_tw, family = delta_lognormal())
R> fit_dpg <- update(fit_tw, family = delta_gamma(type = "poisson-link"))
R> fit_dpl <- update(fit_tw, family = delta_lognormal(type = "poisson-link"))
```

We can then compare the models via AIC:

```
R> AIC(fit_tw, fit_dg, fit_dl, fit_dpg, fit_dpl) |>
+    mutate(delta_AIC = AIC - min(AIC)) |>
+    arrange(delta_AIC)
```

```
        df      AIC delta_AIC
fit_dpl 19 11043.63   0.00000
fit_dpg 19 11112.30  68.66264
fit_dl  19 11133.52  89.89018
fit_dg  19 11208.90 165.26222
fit_tw  12 11356.57 312.93803
```

We find that the Poisson-link delta-lognormal model (Thorson 2018) is favored by marginal AIC. In an applied situation, we would inspect the distribution of the residuals and consider comparing models with cross-validation.

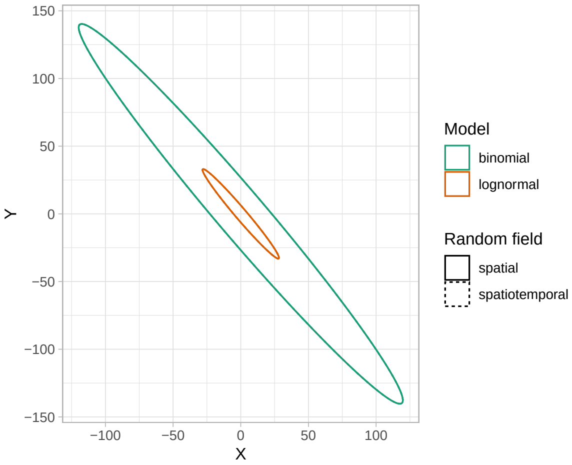


Figure 6: A visualization of anisotropy from the function `plot_anisotropy()`. Ellipses are centered at coordinates of zero in the units that the X-Y coordinates are modeled. The ellipses show the spatial and spatiotemporal range (distance at which correlation is nearly independent) in any direction from the center (zero).

## 5.4. Adding AR(1) random fields and comparing isotropic correlation

We next test two additional model formulations: making the spatial correlation isotropic (the default) instead of anisotropic, and structuring the spatiotemporal random fields as AR(1) to allow spatiotemporal patterns to partially persist from year to year.

The AR(1) fields can be represented as

$$\begin{aligned}
&\boldsymbol{\epsilon}_{t=1} \sim \mathrm{MVNormal}(\mathbf{0}, \boldsymbol{\Sigma}_{\epsilon}),\\
&\boldsymbol{\epsilon}_{t>1} = \rho \boldsymbol{\epsilon}_{t-1} + \sqrt{1-\rho^2}\boldsymbol{\delta}_t,\ \boldsymbol{\delta}_t \sim \mathrm{MVNormal}\left(\mathbf{0}, \boldsymbol{\Sigma}_{\epsilon}\right),
\end{aligned}$$

where $\rho$ represents the estimated autoregressive parameter allowing the spatial field at time $t$ to be correlated with the spatial field at time $t-1$ with deviations created by $\boldsymbol{\delta}_t$, which are themselves independent random fields each year. This is equivalent to a separable GMRF model with precision arising from the Kronecker product of the spatial precision and an AR(1) temporal precision.

```
R> fit_dpl_iso <- update(fit_dpl, anisotropy = FALSE)
R> fit_dpl_ar1 <- update(fit_dpl, spatiotemporal = "ar1")
R> AIC(fit_dpl_ar1, fit_dpl, fit_dpl_iso)
```

```
            df      AIC
fit_dpl_ar1 21 11038.88
fit_dpl     19 11043.63
fit_dpl_iso 17 11069.41
```

We find that the anisotropic AR(1) is favored. This makes sense given the elongated shape of the continental shelf and the rapid transition to deeper water. We can use `plot_anisotropy()` to visually inspect the anisotropy (Figure 6).

As an example of estimation time for these complex spatiotemporal models, the Tweedie model (`fit_tw`), Poisson-link delta-lognormal model (`fit_dpl`), and Poisson-link delta-lognormal model with autoregressive random fields (`fit_dpl_ar1`) took approximately 10, 30, and 90 seconds to fit, respectively, on an Apple MacBook Pro with an M2 Pro processor and Apple's **vecLib** implementation of **BLAS** in R 4.4.0.

```
R> plot_anisotropy(fit_dpl_ar1)
```

### 5.5. Inspecting the model

We save our chosen model to the object `fit` to simplify subsequent code, run the `sanity()` check (suppressed for brevity), and inspect `summary()`:

```
R> fit <- fit_dpl_ar1
R> sanity(fit)
R> summary(fit)

Spatiotemporal model fit by ML ['sdmTMB']
Formula: catch_weight ~ s(log_depth)
Mesh: mesh (anisotropic covariance)
Time column: year
Data: dat
Family: delta_lognormal(link1 = 'log', link2 = 'log', type = 'poisson-link')

Delta/hurdle model 1: -----------------------------------
Family: binomial(link = 'log')
Conditional model:
            coef.est coef.se
(Intercept)     2.35    0.53
slog_depth      0.20    0.67

Smooth terms:
                 Std. Dev.
sd__s(log_depth)      3.23

Time-varying parameters:
                 coef.est coef.se
(Intercept)-2004     0.34    0.61
(Intercept)-2005     0.39    0.65
(Intercept)-2006     0.45    0.68
(Intercept)-2007     0.35    0.61
(Intercept)-2008     0.26    0.55
(Intercept)-2009     0.25    0.54
(Intercept)-2010     0.25    0.52
(Intercept)-2011     0.08    0.47
(Intercept)-2012    -0.09    0.49
(Intercept)-2013    -0.10    0.47
```

```
(Intercept)-2014    -0.10    0.45
(Intercept)-2015    -0.05    0.43
(Intercept)-2016     0.01    0.43
(Intercept)-2017    -0.01    0.45
(Intercept)-2018    -0.04    0.44
(Intercept)-2019    -0.15    0.47
(Intercept)-2020    -0.26    0.53
(Intercept)-2021    -0.38    0.60
(Intercept)-2022    -0.30    0.53
rho-(Intercept)      0.85    0.28

Spatiotemporal AR1 correlation (rho): 0.74
Matérn anisotropic range (spatial): 17.5 to 183.6 at 130 deg.
Spatial SD: 0.82
Spatiotemporal marginal AR1 SD: 1.34

Delta/hurdle model 2: -----------------------------------
Family: lognormal(link = 'log')
Conditional model:
            coef.est coef.se
(Intercept)     2.20    0.20
slog_depth      0.11    0.24

Smooth terms:
                 Std. Dev.
sd__s(log_depth)      0.59

Time-varying parameters:
                 coef.est coef.se
(Intercept)-2004     0.43    0.27
(Intercept)-2005     0.25    0.31
(Intercept)-2006     0.13    0.24
(Intercept)-2007     0.03    0.30
(Intercept)-2008    -0.06    0.26
(Intercept)-2009     0.20    0.30
(Intercept)-2010     0.50    0.26
(Intercept)-2011     0.30    0.30
(Intercept)-2012     0.18    0.23
(Intercept)-2013     0.10    0.29
(Intercept)-2014     0.04    0.23
(Intercept)-2015    -0.10    0.29
(Intercept)-2016    -0.26    0.23
(Intercept)-2017    -0.24    0.30
(Intercept)-2018    -0.27    0.23
(Intercept)-2019    -0.26    0.35
(Intercept)-2020    -0.30    0.35
(Intercept)-2021    -0.41    0.24
```

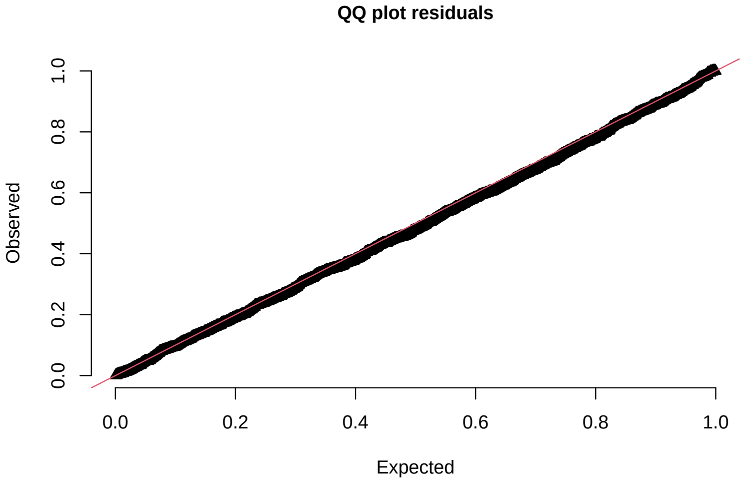


Figure 7: Simulation-based randomized quantile residuals from the **DHARMa** package.

```
(Intercept)-2022    -0.20    0.25
rho-(Intercept)      0.62    0.40

Dispersion parameter: 1.10
Spatiotemporal AR1 correlation (rho): 0.22
Matérn anisotropic range (spatial): 4.1 to 43.3 at 130 deg.
Spatial SD: 0.41
Spatiotemporal marginal AR1 SD: 0.82

ML criterion at convergence: 5498.439

See ?tidy.sdmTMB to extract these values as a data frame.
See ?plot_anisotropy to plot the anisotropic range.
```

The output is more complex than our binomial spatial model. We now have two model components (linear predictors), which are shown one after the other. Starting with the binomial component, we have output from the smoother, which includes a linear parameter (`slog_depth`) and the standard deviation on the smoother weights (`sds(log_depth)`). The smoother summary follows the format used in the **brms** package (Bürkner 2017). Next, we have the time-varying intercepts and information on our anisotropic spatial correlation. We then have the second model component (lognormal) with a similar summary structure and the addition of a dispersion parameter, the AR(1) correlation of the spatiotemporal random fields, and a spatiotemporal random field marginal standard deviation.

We can check simulation-based randomized quantile residuals from our chosen model via the **DHARMa** package (Hartig 2024). To do that, we simulate from our model with the `simulate.sdmTMB()` method and pass those simulations to a helper function `dharma_residuals()`.

```
R> set.seed(123)
R> s <- simulate(fit, nsim = 500, type = "mle-mvn")
R> dharma_residuals(s, fit)
```

The quantile-quantile plot suggests that under the model assumptions, the (transformed) residuals are reasonably consistent with an independent uniform(0, 1) distribution.

### 5.6. Visualizing model predictions

Similarly to the first example, we can visualize model predictions on a grid covering the area of interest. Because this is a spatiotemporal model, we first need to replicate our grid for each year we will predict on. Since this is a common operation, we include the function `replicate_df()` to replicate a data frame. We then ensure our data frame contains all the predictors used in the model (here `log_depth`).

```
R> grid <- replicate_df(wcvi_grid, "year", time_values = unique(dat$year))
R> grid$log_depth <- log(grid$depth)
R> head(grid, n = 2)
```

```
         X        Y depth year log_depth
1 733.9852 5346.334   447 2004  6.102559
2 735.9852 5346.334   442 2004  6.091310
```

```
R> pred <- predict(fit, newdata = grid, type = "response")
R> names(pred)
```

```
 [1] "X"            "Y"            "depth"        "year"         "log_depth"
 [6] "est1"         "est2"         "est_non_rf1"  "est_non_rf2"  "est_rf1"
[11] "est_rf2"      "omega_s1"     "omega_s2"     "epsilon_st1"  "epsilon_st2"
[16] "est"
```

Our prediction data frame is similar to the binomial spatial model, but includes columns for the two delta model linear predictors (labelled with suffixes `1` and `2`) and adds an `epsilon_st` column for spatiotemporal random effects. We can easily generate plots from this data frame using **ggplot2** code with `geom_raster()` similarly to our spatial example with Pacific Cod (Figure 8). We suppress that code for brevity.

We can visualize the conditional effect of the bottom depth smoother by predicting across a sequence of depths and holding other variables at reference values (Figure 9). Here, we pick the last year, specify to include both delta model components (`model = NA`), omit the random fields (`re_form = NA`), and return standard errors (`se_fit = TRUE`). Alternatively, we could produce a similar plot using **ggeffects** (Lüdecke 2018) with `ggeffects::ggpredict()`.

```
R> nd <- data.frame(log_depth =
+      seq(min(dat$log_depth), max(dat$log_depth), length.out = 100),
+    year = max(dat$year))
R> pred_depth <- predict(fit, newdata = nd,
+    model = NA, re_form = NA, se_fit = TRUE)
```

### 5.7. Calculating an area-weighted index

We can generate an area-weighted population index (e.g., a relative or absolute index of abundance or biomass) that is independent of sampling locations by predicting from the

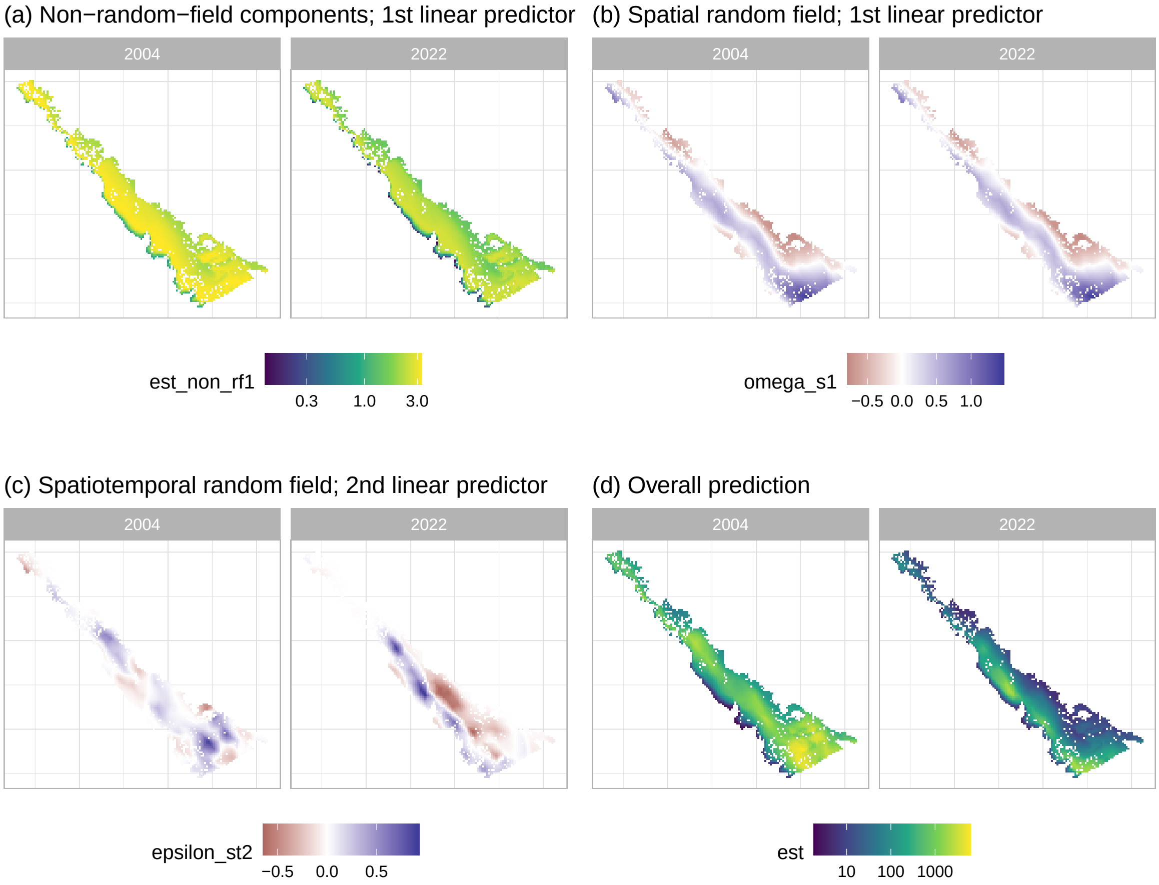

(a) Non−random−field components; 1st linear predictor
2004
2022
est_non_rf1
0.3
1.0
3.0
(b) Spatial random field; 1st linear predictor
2004
2022
omega_s1
−0.5
0.0
0.5
1.0
(c) Spatiotemporal random field; 2nd linear predictor
2004
2022
epsilon_st2
−0.5
0.0
0.5
(d) Overall prediction
2004
2022
est
10
100
1000

Figure 8: Example prediction elements from the spatiotemporal model of Pacific Spiny Dogfish biomass density. Throughout, two example years are shown. (a) `est_non_rf1` refers to the prediction from all non-random-field elements (here, a smoother for bottom depth and the time-varying year effect) from the first linear predictor, (b) `omega_s1` refers to the spatial random field from the first linear predictor, (c) `epsilon_st2` refers to spatiotemporal random fields from the second linear predictor, and (d) `est` refers to the overall prediction estimate combining all effects. The spatial random field is constant through time (i.e., the two panels in b are identical) and represents static biotic or abiotic features not included as covariates (e.g., habitat). The spatiotemporal random fields are different at each time step and here are constrained to follow an AR(1) process. They represent temporal variability in the spatial patterning of Pacific Spiny Dogfish (e.g., resulting from movement or local changes in population density).

model on a grid covering the area of interest and summing the predicted biomass with the `get_index()` function (Figure 10). We supply the grid cell area (4 km$^2$) to the `area` argument and specify `bias_correct = TRUE` to enable a bias correction needed due to the non-linear transformation of the random effects (Thorson and Kristensen 2016).

```
R> grid$area <- 4
R> pred2 <- predict(fit, newdata = grid, return_tmb_object = TRUE)
R> ind <- get_index(pred2, bias_correct = TRUE, area = grid$area)
```

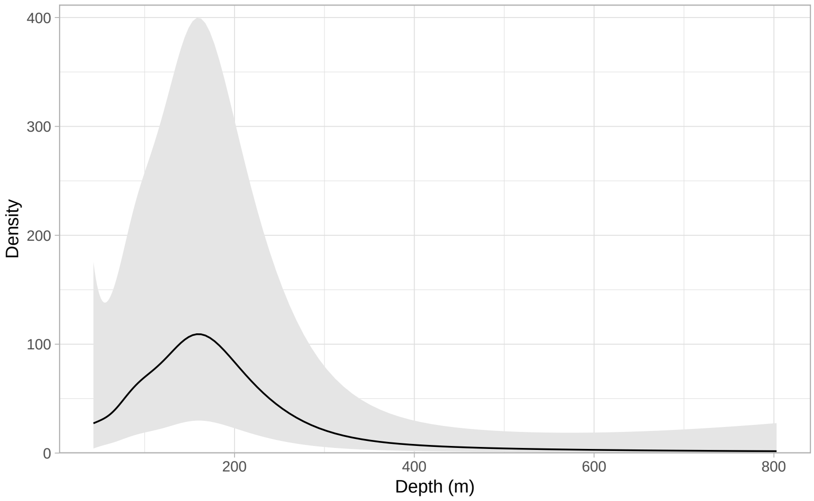


Figure 9: The conditional effect of ocean bottom depth on Pacific Spiny Dogfish population density. The line and shaded ribbon represent the mean and 95% confidence interval, respectively. Other fixed effects are held at constant values and the random fields are set to their expected value (zero).

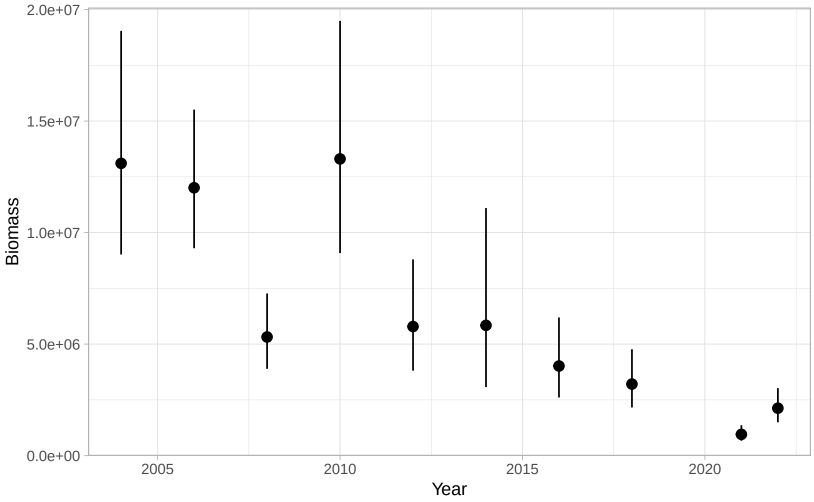


Figure 10: Area-weighted index of relative biomass over time for Pacific Spiny Dogfish. Dots and line segments represent means and 95% confidence intervals.

## 6. Example: Spatially varying coefficients

In our final example, we demonstrate a model with spatially varying coefficient (SVC) effects and illustrate combining uncertainty from parameters by working with draws from the joint parameter precision matrix. SVC models are a class of models in which coefficients are allowed to vary spatially constrained by some smooth function (Hastie and Tibshirani 1993; Thorson *et al.* 2023).

Snowy Owls (*Bubo scandiacus*) breed on the Arctic tundra and are irruptive migrants, meaning that they appear across the mid-latitudes of North America in much greater numbers in some winters than others. The reasons for this interannual variation in the number of individuals migrating south are not well understood but seem to be related to high abundances of food during the breeding season and therefore sharp increases in breeding ground population densities (Robillard *et al.* 2016). The North Atlantic Oscillation Index (NAO) has been linked to the productivity of both owls and their prey in Europe (Millon *et al.* 2014). Because both productivity and the choice of wintering location could be influenced by climate, we model an SVC effect of the annual mean NAO index on early winter abundance across the southern boundary of their winter distribution. Annual mean NAO captures the preceding winter's conditions, combined with breeding season and early winter climate.

To do this, we use counts of Snowy Owls observed by the annual Christmas Bird Counts (National Audubon Society 2021) from all locations where they have been recorded and for which there were at least three counts conducted from 1979 to 2020. Our data are provided in the supplementary materials and contain columns for spatial coordinates (in an Albers projection for North America and divided by 100000 to give units of 100 km), year, year as a factor, the annual NAO value, and owl count:

```
R> select(snow, X, Y, year, year_f, nao, count) |>
+    head()
```

```
# A tibble: 6 x 6
      X     Y  year year_f   nao count
  <dbl> <dbl> <int> <fct>  <dbl> <dbl>
1 -12.0  14.2  1979 1979    0.19     1
2 -11.1  16.9  1979 1979    0.19     3
3 -11.7  17.1  1979 1979    0.19     0
# i 3 more rows
```

## 6.1. Fitting the model

We will fit counts using a negative binomial (NB2, Hilbe 2011) distribution, random intercepts for year, spatial and spatiotemporal random fields, and an SVC associated with the NAO. Centering and scaling variables (e.g., by their mean and standard deviation) can be helpful to reduce the correlation between the SVC and other random fields that are included in the model when estimating SVCs. Here, NAO is an index with a mean near zero and an SD not too far from 1, so we will leave it as is. However, we also include NAO as a main effect since the SVC random field is drawn from a multivariate normal distribution with mean zero.

We can write the model as

$$
\begin{aligned}
y_{\boldsymbol{s},t} &\sim \mathrm{NB2}\left(\mu_{\boldsymbol{s},t}, \phi\right), \\
\mu_{\boldsymbol{s},t} &= \exp\left(\boldsymbol{X}^{\mathrm{main}}_{\boldsymbol{s},t}\boldsymbol{\beta} + \alpha_t + \boldsymbol{X}^{\mathrm{svc}}_{\boldsymbol{s},t}\zeta_{\boldsymbol{s}} + \omega_{\boldsymbol{s}} + \epsilon_{\boldsymbol{s},t}\right), \\
\alpha_t &\sim \mathrm{Normal}(0, \sigma^2_\alpha), \\
\boldsymbol{\zeta} &\sim \mathrm{MVNormal}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_\zeta\right), \\
\boldsymbol{\omega} &\sim \mathrm{MVNormal}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_\omega\right), \\
\boldsymbol{\epsilon}_t &\sim \mathrm{MVNormal}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_\epsilon\right),
\end{aligned}
$$

where three types of random fields are now included: spatially varying NAO coefficients ($\zeta_{\boldsymbol{s}}$), a spatial intercept ($\omega_{\boldsymbol{s}}$), and spatiotemporal variation ($\epsilon_{\boldsymbol{s},t}$). The NB2 distribution is specified with a mean $\mu_{\boldsymbol{s},t}$ and size parameter $\phi$. The observation variance scales quadratically with the mean: $\mathrm{Var}[y] = \mu + \mu^2/\phi$ (Hilbe 2011). The $\alpha_t$ represent IID random intercepts by year. We can then fit this model:

```
R> mesh_snow <- make_mesh(snow, xy_cols = c("X", "Y"), cutoff = 1.5)
R> fit_owl <- sdmTMB(count ~ 1 + nao + (1 | year_f),
+    spatial_varying = ~ 0 + nao, time = "year", data = snow,
+    mesh = mesh_snow, family = nbinom2(link = "log"),
+    spatial = "on", spatiotemporal = "iid", silent = FALSE)
```

## 6.2. Inspecting the model

`summary()` prints standard model information:

```
R> summary(fit_owl)

Spatiotemporal model fit by ML ['sdmTMB']
Formula: count ~ 1 + nao + (1 | year_f)
Mesh: mesh_snow (isotropic covariance)
Time column: year
Data: snow
Family: nbinom2(link = 'log')

Random intercepts and/or slopes:

Conditional model:
     Groups        Name    Variance    Std.Dev.
     year_f (Intercept)        0.09        0.30

Conditional model:
            coef.est coef.se
(Intercept)    -3.23    5.44
nao             0.36    0.22

Dispersion parameter: 0.46
Matérn range: 25.21
Spatial SD: 6.97
Spatially varying coefficient SD (nao): 0.17
Spatiotemporal IID SD: 0.79
ML criterion at convergence: 17512.366

See ?tidy.sdmTMB to extract these values as a data frame.
```

In addition to the output seen for other models, we now have a section for random intercepts and a standard deviation for the SVC random field. Given our model specification, all random fields are sharing a single Matérn range. We can also check the confidence intervals (CIs) on the main effect of NAO and see that they overlap zero.

```
R> tidy(fit_owl, conf.int = TRUE)
```

```
# A tibble: 2 x 5
  term        estimate std.error conf.low conf.high
  <chr>          <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept)   -3.23      5.44  -13.9         7.42
2 nao            0.356     0.224  -0.0828     0.795
```

However, given that this is only part of the effect of NAO that we include in our model, we next assess whether this variable has a meaningful effect in some locations, even if not overall.

### 6.3. Extracting the spatially varying effects with uncertainty

The spatially varying effect at any point in space is the combination of the main effect and SVC random effect for `nao`. Mean estimates of the SVC random effect are available in the output of `predict.sdmTMB()` in a column starting with `zeta_s` (in this case, `zeta_s_nao`). However, we might wish to combine the fixed and random components of a spatially varying effect and assess the uncertainty of these combined predictions. We illustrate a way of accomplishing this by simulating from the fixed and random effects while assuming that parametric uncertainty is well approximated using a multivariate normal distribution and the joint precision matrix. We do this by specifying a non-null number of simulation draws to `nsim` in `predict.sdmTMB()`. By default, `nsim > 0` will return a matrix of draws for the overall prediction. Here, we instead specify that we want to return draws for the `zeta_s` ($\zeta_s$) random field, which is the SVC random field (`sims_var = "zeta_s"`). This returns a matrix where each row matches a row of `newdata` and each column is a simulation draw. We then use the function `spread_sims()` to draw 300 simulations for the parameters themselves. Because the simulations are stored in different dimensions, the random field draws must be transposed `t()` before combining the vector of main effect draws (`sims$nao`) with the random field values `zeta_s`. Next, we can calculate the median, and upper and lower quantiles for each column of data, which correspond to the rows in the data provided. For this example, we use thresholds of 0.025 and 0.975 representing a 95% CI.

```
R> set.seed(42)
R> zeta_s <- predict(fit_owl, newdata = snow,
+    nsim = 300, sims_var = "zeta_s")
R> dim(zeta_s)
```

```
[1] 30392   300
```

```
R> sims <- spread_sims(fit_owl, nsim = 300)
R> dim(sims)
```

```
[1] 300   9
```

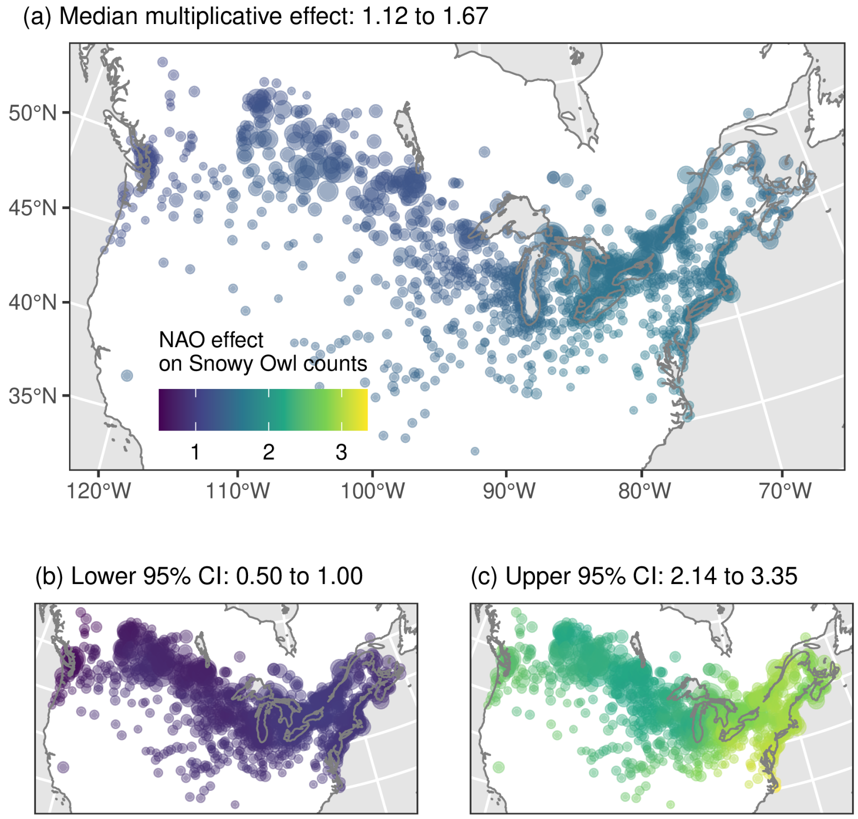

(a) Median multiplicative effect: 1.12 to 1.67
50°N
45°N
40°N
35°N
120°W
110°W
100°W
90°W
80°W
70°W
NAO effect
on Snowy Owl counts
1
2
3
(b) Lower 95% CI: 0.50 to 1.00
(c) Upper 95% CI: 2.14 to 3.35

Figure 11: Spatially varying effect of mean annual NAO (North Atlantic Oscillation) on counts of Snowy Owls observed on annual Christmas Bird Counts from 1979–2020 in Canada and the US. The effect is multiplicative on owl count per NAO unit. In the west, the lower bound of values overlaps 1, implying no effect, whereas in the southeast the effect becomes positive. Point size is scaled to the mean counts in each location.

```
R> combined <- sims$nao + t(zeta_s)
R> snow$nao_effect <- exp(apply(combined, 2, median))
R> snow$nao_effect_lwr <- exp(apply(combined, 2, quantile, probs = 0.025))
R> snow$nao_effect_upr <- exp(apply(combined, 2, quantile, probs = 0.975))
```

We can make a basic plot using the following code. A more elaborate version including separate panels for each of the CIs is shown in Figure 11.

```
R> ggplot(snow, aes(X, Y)) + geom_point(aes(colour = nao_effect))
```

Overall, we find a weak average positive effect of annual mean NAO on overall counts with a southeast to northwest gradient in the intensity of the effect (Figure 11). At some locations, the lower CI on the exponentiated effect is above 1. This result is consistent with owls closest to the Atlantic coast and those migrating the furthest south being the most affected by the NAO.

# 7. Package comparisons

There are many R packages capable of fitting geostatistical spatial or spatiotemporal models (e.g., Heaton *et al.* 2019). **sdmTMB**, **VAST**, **tinyVAST**, R-**INLA**/**inlabru**, and **spaMM** (Rousset and Ferdy 2014) are the most closely related, as they all provide a user interface to SPDE-based GMRF models. In our software comparison (Table 1), we also include **mgcv** as it can be adapted to use the SPDE (Miller *et al.* 2020) and **spBayes** (Finley *et al.* 2007, 2015) since it is a prominent package that can fit related predictive-process models without the SPDE. **sdmTMB**, **VAST**, **tinyVAST**, and **mgcv** can estimate anisotropic covariance whereas R-**INLA**/**inlabru** and **spBayes** are currently limited to isotropic covariance. To our knowledge, **VAST** and **tinyVAST** are the only packages to implement spatial (Thorson *et al.* 2015a) and spatial dynamic factor analysis (Thorson *et al.* 2016) and spatial empirical orthogonal function (EOF) regression (Thorson *et al.* 2020). Of these packages, only **sdmTMB** and **inlabru** can currently fit threshold (e.g., hockey-stick) covariate relationships. To our understanding, **spaMM** is limited to a spatial random field (i.e., does not fit spatiotemporal fields) and **spBayes** implements spatiotemporal fields, but only as a random walk. There is considerable variability in the available observation likelihoods across packages (Table 1).

We ran a simple speed comparison between **sdmTMB**, R-**INLA**/**inlabru**, **spaMM**, and **mgcv** for fitting an SPDE spatial random field model to 1,000, 10,000, or 100,000 data points with Gaussian error across a range of mesh resolutions (Figure 12, Appendix A). Our test was restricted to one core and default R algebra libraries using R 4.4.0 and **Matrix** version 1.7.0. With up to 10,000 rows of data, **sdmTMB** was fastest at approximately a three- to 13-fold speed increase over R-**INLA**/**inlabru** (Figure 12a–b). At larger sample sizes, **inlabru** was more affected by mesh resolution than **sdmTMB** (Figure 12c). **mgcv** was most affected by mesh resolution (Figure 12) – timing was on par or faster than the other packages at low ($\approx$ 200) mesh resolutions but much slower as mesh complexity grew into commonly used ranges. The **mgcv** model was fit with the `bam()` function instead of the standard `gam()`. The `bam()` function here uses numerical methods optimized for large datasets, including a discretization of covariate values (Wood *et al.* 2015, 2017; Li and Wood 2020). The standard `gam()` function was approximately 60 times slower than `bam()` for the largest dataset with the most complex mesh (not shown). **spaMM** scaled well to large datasets with high mesh resolutions (Figure 12c), but to our knowledge can only fit spatial GMRFs. At low mesh resolutions, a large portion of the differences in timing is likely a result of initial model processing and not estimation itself. Speed increases can allow for more rapid and thorough model exploration and experimentation with this class of computationally intensive models. For users ultimately interested in Bayesian inference, the approximate Bayesian inference offered by R-**INLA**/**inlabru** is likely to be faster than passing the same model from **sdmTMB**/**VAST** to **tmbstan** for full MCMC-based Bayesian inference.

# 8. Discussion

How does one choose among the related packages mentioned in this paper to fit SPDE-based geostatistical GLMMs? Assuming a given package can fit the model of interest (Table 1), we suggest the primary differences are the user interface and speed. We think users familiar with `stats::glm()`, **lme4**, **mgcv**, or **glmmTMB** will find **sdmTMB** most approachable. Users familiar with R-**INLA** will find **inlabru** approachable. Users familiar with **mgcv** can

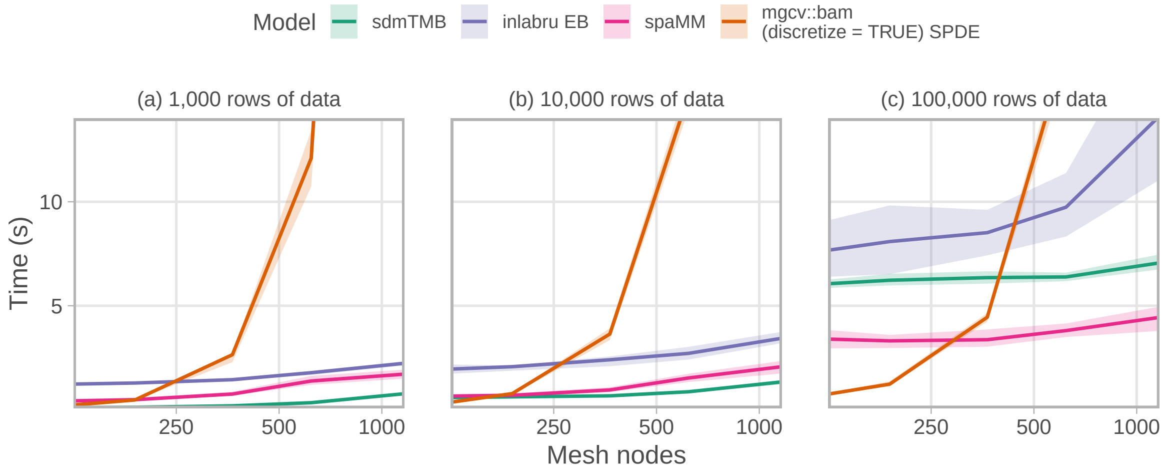


Figure 12: Comparison of time to fit an SPDE spatial random field model with an intercept, one fixed-effect predictor, Gaussian error, and a sequence of SPDE resolutions to three dataset sizes. Lines represent means and ribbons represent 90% quantiles across 50 random iterations. **VAST** and **tinyVAST** should be similar to **sdmTMB** and so are not shown. **inlabru** (version 2.10.1) used the empirical Bayes integration strategy and Gaussian approximation with `bru_max_iter = 1`, and the `like()` formulation. **mgcv** (version 1.9.1) used `bam()`, `method = "fREML"`, and discretized covariates (Miller *et al.* 2020). Note that **spaMM** (version 4.4.16) only fits spatial, not spatiotemporal, models. All platforms were restricted to one core and could be faster with parallel computation or optimized algebra libraries. See Figure 13 for a version with log-log axes.

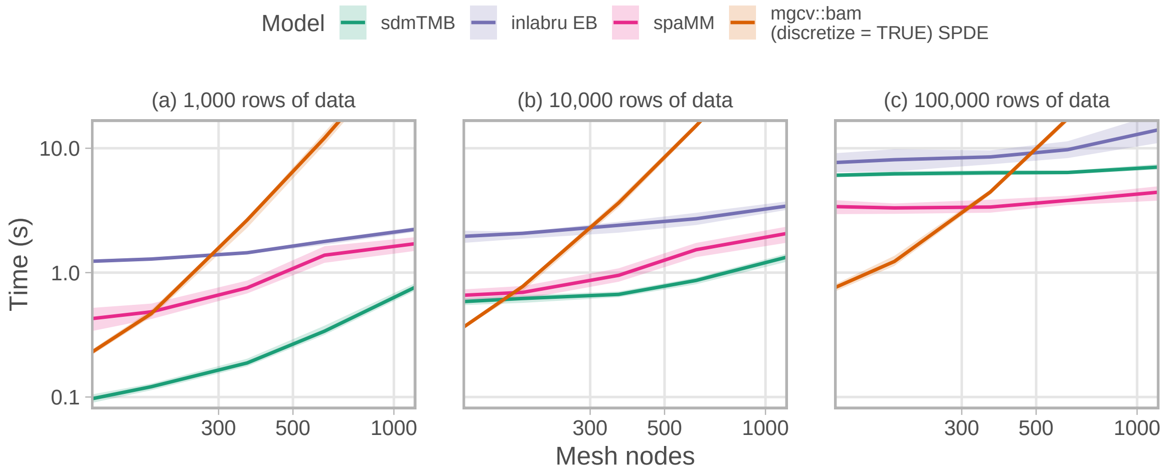


Figure 13: Same as Figure 12 but with both axes on a log scale.

| | **sdmTMB** | **VAST** | R-**INLA**/**inlabru** | **mgcv** | **spBayes** | **spaMM** |
|---|---|---|---|---|---|---|
| Time-varying coefficients | ✓ | –[1] | ✓ | ✓ | ✓ | – |
| Spatially varying coefficients (SVC) | ✓ | ✓ | ✓ | ✓ | ✓ | – |
| GAMs[2] | ✓ | – | ✓ | ✓ | – | – |
| Threshold covariates | ✓ | – | ✓[3] | – | – | – |
| Offsets | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Spatiotemporal fields | ✓ | ✓ | ✓ | ✓ | ✓[4] | – |
| Spatial + spatiotemporal fields | ✓ | ✓ | ✓ | ✓ | – | – |
| Anisotropy | ✓ | ✓ | – | ✓ | – | – |
| Correlation barriers | ✓ | ✓ | ✓ | ✓ | – | – |
| Separate range parameters for fields | ✓ | – | ✓ | ✓ | – | – |
| Share range parameters across fields | ✓ | ✓ | ✓ | – | – | – |
| SPDE-based | ✓ | ✓ | ✓ | ✓[5] | –[6] | ✓ |
| NB1 distribution | ✓ | – | ✓ | ✓ | – | ✓ |
| NB2 distribution | ✓ | ✓[7] | ✓ | ✓ | – | ✓ |
| Zero-truncated distributions | ✓ | – | ✓ | – | – | ✓ |
| Zero-inflated distributions | ✓ | ✓ | ✓ | – | – | ✓ |
| Tweedie distribution | ✓ | ✓ | ✓ | ✓[8] | – | – |
| Student-*t* distribution | ✓ | – | ✓ | ✓ | – | – |
| Censored Poisson distribution | ✓ | – | ✓ | – | – | – |
| Log Gaussian Cox processes | –[9] | –[9] | ✓ | –[9] | –[9] | –[9] |
| Multivariate responses | – | ✓ | ✓ | ✓ | ✓ | ✓ |
| Built-in delta/hurdle models | ✓ | ✓ | ✓ | –[10] | – | ✓ |
| Poisson-link delta models | ✓ | ✓ | ✓ | – | – | – |
| Likelihood weights | ✓ | – | ✓ | ✓ | ✓ | ✓ |
| Maximum/marginal likelihood | ✓ | ✓ | – | ✓ | – | – |
| Bayesian/optionally Bayesian | ✓ | ✓ | ✓ | ✓ | ✓ | – |
| Priors/penalties | ✓ | – | ✓ | – | ✓ | – |
| Matérn PC priors | ✓ | – | ✓ | – | – | – |
| Spatial (or spatial dynamic) factor analysis | – | ✓ | – | – | – | – |
| Empirical Orthogonal Function (EOF) analysis | – | ✓ | – | – | – | – |
| Built-in area-weighted index standardization | ✓ | ✓ | – | – | – | – |
| Built-in cross-validation | ✓ | – | – | – | – | – |

Table 1: Comparison of functionality between several R packages that can fit geostatistical GLMMs. At the time of writing, the feature set of **tinyVAST** is rapidly evolving and so is not shown here. Notes: [1]Technically possible but non-trivial. [2]Penalized smoother GAMs that determine "wiggliness". [3]**inlabru** but not R-**INLA**. [4]Spatiotemporal fields as random walk only. [5]SPDE approach as in Miller *et al.* (2020). [6]Does have predictive process knots. [7]Zero-inflated NB2 only. [8]Tweedie power parameter fixed for `mgcv::gamm()`. [9]Possible as log-linked Poisson GLMM with aggregated data. [10]Hurdle models possible by fitting components separately.

adapt **mgcv** to fit similar models with custom code (Miller *et al.* 2020). R-**INLA**/**inlabru** and **mgcv** are also general-purpose modeling packages. **VAST** and **tinyVAST** are the sole options for fitting some multivariate models; alternatively, because these packages focus on multivariate delta models and fisheries applications, users fitting "simple" univariate spatial/spatiotemporal GLMMs will likely find **sdmTMB** more straightforward. Users looking for calculations of derived variables with uncertainty, such as area-weighted population indices, may favor **sdmTMB** or **VAST** (although such quantities can be calculated using other packages post hoc). Overall, **sdmTMB** unites functionality useful in many applied settings into a single package.

While our examples focused on applications to ecological data, the SPDE approach and the functionality of **sdmTMB** have applications in many other fields. Examples include spatial models of disease spread (Moraga *et al.* 2021), spatial econometric models of quantities such as housing prices (Bivand *et al.* 2014), analyzing medical imaging data such as MRI scans (Parisa Naseri and Tabatabaei 2022), and geophysical models of seismic waves following earthquakes (Zhang *et al.* 2015). The **sdmTMB** model is also relevant to spatial (Elhorst 2010; Lee and Yu 2010) and dynamic spatial panel data models (Elhorst 2012) in econometrics. These examples underscore the versatile nature of the SPDE approach and the potential uses of **sdmTMB** across various scientific and industrial sectors.

The GLMMs underpinning **sdmTMB** models are spatially explicit and derived from a mechanistic diffusion process – they estimate interpretable parameters of a spatial covariance function: parameters defining the magnitude of spatial variation and the rate of correlation decay with distance. In contrast, non-parametric approaches (e.g., **randomForest**, Liaw and Wiener 2002; **MaxEnt**, Phillips *et al.* 2006) and most smooths in **mgcv** (Wood 2017) do not estimate spatial covariance functions. Approaches such as conditional autoregressive models (CAR) are applicable to areal data, where the spatial domain is discretized into a set of vertices or polygons. Data aggregated to polygon or grid level may be analyzed using other spatial models, including CAR models (e.g., Ver Hoef *et al.* 2018); **sdmTMB** can also fit models with areal data if each polygon has an associated centroid. A benefit of the geostatistical approach over CAR or similar models is that the parameters describing spatial covariance can be more easily interpreted (Wall 2004).

Additional functionality in **sdmTMB** not already mentioned includes interpolating across missing time slices and forecasting, the barrier SPDE model (Bakka *et al.* 2019), and time-varying spatiotemporal covariance parameters (Ward *et al.* 2022). There are several planned future additions to the **sdmTMB** model structure. A subset of features to be added includes (1) the ability to specify observation-specific likelihoods to integrate different data types (e.g., Grüss and Thorson 2019), (2) inclusion of a dispersion formula similar to **glmmTMB**, (3) covariate diffusion to model non-local covariate effects (Lindmark *et al.* 2025), and (4) integration with the **RTMB** (Kristensen 2025) package so that the model code base is written in R rather than C++.

# Acknowledgements

**sdmTMB** would not be possible without the **TMB** (Kristensen *et al.* 2016) and R-**INLA** (Rue *et al.* 2009; Lindgren *et al.* 2011; Lindgren and Rue 2015) R packages. **sdmTMB** is heavily inspired by and in some places code has been adapted from both the **VAST** (Thorson 2019a) and **glmmTMB** (Brooks *et al.* 2017) R packages (as described in the `DESCRIPTION` and `inst/COPYRIGHTS` files). Smoother support was possible thanks to **mgcv** (Wood 2017). We thank the authors of all these packages. We thank S. Kotwicki, M. Lindmark, M. Martin, C.C. Monnahan, P.M. Regular, J. Watson, two anonymous reviewers, and the editor for helpful comments that substantially improved the manuscript. We thank the many individuals who have contributed to collecting the trawl survey data with Fisheries and Oceans Canada that was used in the Pacific Cod and Pacific Spiny Dogfish examples. Christmas Bird Count data for the Snowy Owl example were provided by National Audubon Society and through the generous efforts of Bird Studies Canada and countless volunteers across the Western Hemisphere.

# References

Aitchison J (1955). "On the Distribution of a Positive Random Variable Having a Discrete Probability Mass at the Origin." *Journal of the American Statistical Association*, **50**(271), 901. doi:10.2307/2281175.

Akaike H (1974). "A New Look at the Statistical Model Identification." *IEEE Transactions on Automatic Control*, **19**(6), 716–723. doi:10.1109/tac.1974.1100705.

Bachl FE, Lindgren F, Borchers DL, Illian JB (2019). "**inlabru**: An R Package for Bayesian Spatial Modelling from Ecological Survey Data." *Methods in Ecology and Evolution*, **10**(6), 760–766. doi:10.1111/2041-210x.13168.

Bakka H, Rue H, Fuglstad GA, Riebler A, Bolin D, Illian J, Krainski E, Simpson D, Lindgren F (2018). "Spatial Modeling with R-**INLA**: A Review." *Wiley Interdisciplinary Reviews: Computational Statistics*, **10**(6), e1443. doi:10.1002/wics.1443.

Bakka H, Vanhatalo J, Illian J, Simpson D, Rue H (2019). "Non-Stationary Gaussian Models with Physical Barriers." *Spatial Statistics*, **29**, 268–288. doi:10.1016/j.spasta.2019.01.002.

Banerjee S, Gelfand AE, Finley AO, Sang H (2008). "Gaussian Predictive Process Models for Large Spatial Data Sets." *Journal of the Royal Statistical Society B*, **70**(4), 825–848. doi:10.1111/j.1467-9868.2008.00663.x.

Barnett LAK, Ward EJ, Anderson SC (2021). "Improving Estimates of Species Distribution Change by Incorporating Local Trends." *Ecography*, **44**(3), 427–439. doi:10.1111/ecog.05176.

Barrowman N, Myers RA (2000). "Still More Spawner-Recruitment Curves: The Hockey Stick and Its Generalizations." *Canadian Journal of Fisheries and Aquatic Sciences*, **57**, 665–676. doi:10.1139/f99-282.

Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using **lme4**." *Journal of Statistical Software*, **67**(1), 1–48. doi:10.18637/jss.v067.i01.

Bates D, Mächler M, Jagan M (2025). ***Matrix**: Sparse and Dense Matrix Classes and Methods.* doi:10.32614/CRAN.package.Matrix. R package version 1.7-4.

Bengtsson H (2021). "A Unifying Framework for Parallel and Distributed Processing in R Using Futures." *The R Journal*, **13**(2), 208–227. doi:10.32614/rj-2021-048.

Bivand RS, Gómez-Rubio V, Rue H (2014). "Approximate Bayesian Inference for Spatial Econometrics Models." *Spatial Statistics*, **9**, 146–165. doi:10.1016/j.spasta.2014.01.002.

Blackford LS, Petitet A, Pozo R, Remington K, Whaley RC, Demmel J, Dongarra J, Duff I, Hammarling S, Henry G, *et al.* (2002). "An Updated Set of Basic Linear Algebra Subprograms (BLAS)." *ACM Transactions on Mathematical Software*, **28**(2), 135–151. doi:10.1145/567806.567807.

Breivik ON, Aanes F, Søvik G, Aglen A, Mehl S, Johnsen E (2021). "Predicting Abundance Indices in Areas Without Coverage with a Latent Spatio-Temporal Gaussian Model." *ICES Journal of Marine Science*, p. fsab073. doi:10.1093/icesjms/fsab073.

Brooks ME, Kristensen K, Van Benthem KJ, Magnusson A, Berg CW, Nielsen A, Skaug HJ, Maechler M, Bolker BM (2017). "**glmmTMB** Balances Speed and Flexibility among Packages for Zero-Inflated Generalized Linear Mixed Modeling." *The R Journal*, **9**(2), 378–400. doi:10.32614/rj-2017-066.

Bürkner PC (2017). "**brms**: An R Package for Bayesian Multilevel Models Using Stan." *Journal of Statistical Software*, **80**(1), 1–28. doi:10.18637/jss.v080.i01.

Carpenter B, Gelman A, Hoffman MD, Lee D, Goodrich B, Betancourt M, Brubaker M, Guo J, Li P, Riddell A (2017). "Stan: A Probabilistic Programming Language." *Journal of Statistical Software*, **76**(1), 1–32. doi:10.18637/jss.v076.i01.

Chilés JP, Delfiner P (1999). *Geostatistics: Modeling Spatial Uncertainty.* John Wiley & Sons.

Cressie N (1993). *Statistics for Spatial Data.* John Wiley & Sons.

Cressie NAC, Wikle CK (2011). *Statistics for Spatio-Temporal Data.* Wiley Series in Probability and Statistics. John Wiley & Sons, Hoboken.

Datta A, Banerjee S, Finley AO, Gelfand AE (2016). "Hierarchical Nearest-Neighbor Gaussian Process Models for Large Geostatistical Datasets." *Journal of the American Statistical Association*, **111**(514), 800–812. doi:10.1080/01621459.2015.1044091.

Diggle PJ, Ribeiro PJ (2007). *Model-Based Geostatistics.* Springer-Verlag. doi:10.1007/978-0-387-48536-2.

Dormann CF, McPherson JM, Araújo MB, Bivand R, Bolliger J, Carl G, Davies RG, Hirzel A, Jetz W, Kissling WD, Kühn I, Ohlemüller R, Peres?Neto PR, Reineking B, Schröder B, Schurr FM, Wilson R (2007). "Methods to Account for Spatial Autocorrelation in the Analysis of Species Distributional Data: A Review." *Ecography*, **30**(5), 609–628. doi:10.1111/j.2007.0906-7590.05171.x.

Dunn PK, Smyth GK (1996). "Randomized Quantile Residuals." *Journal of Computational and Graphical Statistics*, **5**(3), 236–244. doi:10.1080/10618600.1996.10474708.

Elhorst JP (2010). "Spatial Panel Data Models." In MM Fischer, A Getis (eds.), *Handbook of Applied Spatial Analysis: Software Tools, Methods and Applications*, pp. 377–407. Springer-Verlag, Berlin. doi:10.1007/978-3-642-03647-7_19.

Elhorst JP (2012). "Dynamic Spatial Panels: Models, Methods, and Inferences." *Journal of Geographical Systems*, **14**(1), 5–28. doi:10.1007/s10109-011-0158-4.

Finley AO, Banerjee S, Carlin BP (2007). "**spBayes**: An R Package for Univariate and Multivariate Hierarchical Point-Referenced Spatial Models." *Journal of Statistical Software*, **19**(4). doi:10.18637/jss.v019.i04.

Finley AO, Banerjee S, E Gelfand A (2015). "**spBayes** for Large Univariate and Multivariate Point-Referenced Spatio-Temporal Data Models." *Journal of Statistical Software*, **63**(13). doi:10.18637/jss.v063.i13.

Finley AO, Datta A, Banerjee S (2022). "**spNNGP** R Package for Nearest Neighbor Gaussian Process Models." *Journal of Statistical Software*, **103**(5). doi:10.18637/jss.v103.i05.

Fuglstad GA, Simpson D, Lindgren F, Rue H (2019). "Constructing Priors That Penalize the Complexity of Gaussian Random Fields." *Journal of the American Statistical Association*, **114**(525), 445–452. doi:10.1080/01621459.2017.1415907.

Gay DM (1990). "Usage Summary for Selected Optimization Routines." *Computing Science Technical Report*, **153**, 1–21. doi:10.1007/978-1-349-20837-1_8.

Geisser S, Eddy WF (1979). "A Predictive Approach to Model Selection." *Journal of the American Statistical Association*, **74**(365), 153–160. doi:10.1080/01621459.1979.10481632.

Gelfand AE, Banerjee S (2017). "Bayesian Modeling and Analysis of Geostatistical Data." *Annual Review of Statistics and Its Application*, **4**(1), 245–266. doi:10.1146/annurev-statistics-060116-054155.

Gittleman JL, Kot M (1990). "Adaptation: Statistics and a Null Model for Estimating Phylogenetic Effects." *Systematic Biology*, **39**(3), 227–241. doi:10.2307/2992183.

Greven S, Kneib T (2010). "On the Behaviour of Marginal and Conditional AIC in Linear Mixed Models." *Biometrika*, **97**(4), 773–789. doi:10.1093/biomet/asq042.

Grüss A, Thorson JT (2019). "Developing Spatio-Temporal Models Using Multiple Data Types for Evaluating Population Trends and Habitat Usage." *ICES Journal of Marine Science*, **76**(6), 1748–1761. doi:10.1093/icesjms/fsz075.

Hartig F (2024). ***DHARMa**: Residual Diagnostics for Hierarchical (Multi-Level/Mixed) Regression Models*. doi:10.32614/CRAN.package.DHARMa. R package version 0.4.7.

Haskard KA (2007). *An Anisotropic Matern Spatial Covariance Model: REML Estimation and Properties*. Ph.D. thesis, The University of Adelaide.

Hastie T, Tibshirani R (1993). "Varying-Coefficient Models." *Journal of the Royal Statistical Society B*, **55**(4), 757–796. doi:10.1111/j.2517-6161.1993.tb01939.x.

Heaton MJ, Datta A, Finley AO, Furrer R, Guinness J, Guhaniyogi R, Gerber F, Gramacy RB, Hammerling D, Katzfuss M, Lindgren F, Nychka DW, Sun F, Zammit-Mangion A (2019). "A Case Study Competition among Methods for Analyzing Large Spatial Data." *Journal of Agricultural, Biological and Environmental Statistics*, **24**(3), 398–425. doi:10.1007/s13253-018-00348-w.

Hilbe JM (2011). *Negative Binomial Regression*. Cambridge University Press.

Krainski E, Gómez-Rubio V, Bakka H, Lenzi A, Castro-Camilo D, Simpson D, Lindgren F, Rue H (2018). *Advanced Spatial Modeling with Stochastic Partial Differential Equations Using R and **INLA***. Chapman & Hall/CRC. doi:10.1201/9780429031892.

Kristensen K (2025). ***RTMB**: R Bindings for **TMB***. doi:10.32614/CRAN.package.RTMB. R package version 1.8.

Kristensen K, Nielsen A, Berg CW, Skaug H, Bell BM (2016). "**TMB**: Automatic Differentiation and Laplace Approximation." *Journal of Statistical Software*, **70**(5), 1–21. doi:10.18637/jss.v070.i05.

Latimer AM, Banerjee S, Sang Jr H, Mosher ES, Silander Jr JA (2009). "Hierarchical Models Facilitate Spatial Analysis of Large Data Sets: A Case Study on Invasive Plant Species in the Northeastern United States." *Ecology Letters*, **12**(2), 144–154. doi:10.1111/j.1461-0248.2008.01270.x.

Lee LF, Yu J (2010). "Estimation of Spatial Autoregressive Panel Data Models with Fixed Effects." *Journal of Econometrics*, **154**(2), 165–185. doi:10.1016/j.jeconom.2009.08.001.

Legendre P, Fortin MJ (1989). "Spatial Pattern and Ecological Analysis." *Vegetatio*, **80**(2), 107–138. doi:10.1007/bf00048036.

Li Z, Wood SN (2020). "Faster Model Matrix Crossproducts for Large Generalized Linear Models with Discretized Covariates." *Statistics and Computing*, **30**(1), 19–25. doi:10.1007/s11222-019-09864-2.

Liaw A, Wiener M (2002). "Classification and Regression by **randomForest**." *R News*, **2**(3), 18–22.

Lindgren F (2025). ***fmesher**: Triangle Meshes and Related Geometry Tools*. doi:10.32614/CRAN.package.fmesher. R package version 0.5.0.

Lindgren F, Bolin D, Rue H (2022). "The SPDE Approach for Gaussian and Non-Gaussian Fields: 10 Years and Still Running." *Spatial Statistics*, **50**, 100599. doi:10.1016/j.spasta.2022.100599.

Lindgren F, Rue H (2015). "Bayesian Spatial Modelling with R-**INLA**." *Journal of Statistical Software*, **63**(1), 1–25. doi:10.18637/jss.v063.i19.

Lindgren F, Rue H, Lindström J (2011). "An Explicit Link between Gaussian Fields and Gaussian Markov Random Fields: The Stochastic Partial Differential Equation Approach." *Journal of the Royal Statistical Society B*, **73**(4), 423–498. doi:10.1111/j.1467-9868.2011.00777.x.

Lindmark M, Anderson SC, Thorson JT (2025). "Estimating Scale-Dependent Covariate Responses Using Two-Dimensional Diffusion Derived from the Stochastic Partial Differential Equation Method." *Methods in Ecology and Evolution*, **17**, 207–218. doi:10.1111/2041-210X.70177.

Lüdecke D (2018). "**ggeffects**: Tidy Data Frames of Marginal Effects from Regression Models." *Journal of Open Source Software*, **3**(26), 772. doi:10.21105/joss.00772.

Matérn B (1986). *Spatial Variation*. 2nd edition. Springer-Verlag, New York.

McCullagh P, Nelder JA (1989). *Generalized Linear Models*. Chapman & Hall. doi:10.1007/978-1-4899-3242-6.

Miller DL, Glennie R, Seaton AE (2020). "Understanding the Stochastic Partial Differential Equation Approach to Smoothing." *Journal of Agricultural, Biological and Environmental Statistics*, **25**, 1–19. doi:10.1007/s13253-019-00377-z.

Millon A, Petty SJ, Little B, Gimenez O, Cornulier T, Lambin X (2014). "Dampening Prey Cycle Overrides the Impact of Climate Change on Predator Population Dynamics: A Long-Term Demographic Study on Tawny Owls." *Global Change Biology*, **20**(6), 1770–1781. doi:10.1111/gcb.12546.

Monnahan CC, Kristensen K (2018). "No-U-Turn Sampling for Fast Bayesian Inference in **ADMB** and **TMB**: Introducing the **adnuts** and **tmbstan** R Packages." *PLOS One*, **13**(5). doi:10.1371/journal.pone.0197954.

Moraga P, Dean C, Inoue J, Morawiecki P, Noureen SR, Wang F (2021). "Bayesian Spatial Modelling of Geostatistical Data Using INLA and SPDE Methods: A Case Study Predicting Malaria Risk in Mozambique." *Spatial and Spatio-Temporal Epidemiology*, **39**, 100440. doi:10.1016/j.sste.2021.100440.

National Audubon Society (2021). "The Christmas Bird Count Historical Results." URL https://www.christmasbirdcount.org/.

Osgood-Zimmerman A, Wakefield J (2023). "A Statistical Review of Template Model Builder: A Flexible Tool for Spatial Modelling." *International Statistical Review*, **91**(2), 318–342. doi:10.1111/insr.12534.

Paradis E, Schliep K (2019). "**ape** 5.0: An Environment for Modern Phylogenetics and Evolutionary Analyses in R." *Bioinformatics*, **35**, 526–528. doi:10.1093/bioinformatics/bty633.

Parisa Naseri HAM, Tabatabaei SM (2022). "Integrated Nested Laplace Approximation Method for Hierarchical Bayesian Inference of Spatial Model with Application to Functional Magnetic Resonance Imaging Data." *Communications in Statistics – Theory and Methods*, **51**(8), 2414–2437. doi:10.1080/03610926.2020.1776327.

Pebesma E (2018). "Simple Features for R: Standardized Support for Spatial Vector Data." *The R Journal*, **10**(1), 439–446. doi:10.32614/rj-2018-009.

Phillips SJ, Anderson RP, Schapire RE (2006). "Maximum Entropy Modeling of Species Geographic Distributions." *Ecological Modelling*, **190**(3), 231–259. doi:10.1016/j.ecolmodel.2005.03.026.

R Core Team (2025). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Roberts DR, Bahn V, Ciuti S, Boyce MS, Elith J, Guillera-Arroita G, Hauenstein S, Lahoz-Monfort JJ, Schröder B, Thuiller W, Warton DI, Wintle BA, Hartig F, Dormann CF (2017). "Cross-Validation Strategies for Data with Temporal, Spatial, Hierarchical, or Phylogenetic Structure." *Ecography*, **40**(8), 913–929. doi:10.1111/ecog.02881.

Robillard A, Therrien JF, Gauthier G, Clark KM, Bêty J (2016). "Pulsed Resources at Tundra Breeding Sites Affect Winter Irruptions at Temperate Latitudes of a Top Predator, the Snowy Owl." *Oecologia*, **181**(2), 423–433. doi:10.1007/s00442-016-3588-3.

Robinson D, Hayes A, Couch S (2025). ***broom**: Convert Statistical Objects into Tidy Tibbles*. doi:10.32614/CRAN.package.broom. R package version 1.0.10.

Rousset F, Ferdy JB (2014). "Testing Environmental and Genetic Effects in the Presence of Spatial Autocorrelation." *Ecography*, **37**(8), 781–790. doi:10.1111/ecog.00566.

Rue H, Held L (2005). *Gaussian Markov Random Fields: Theory and Applications*. Chapman & Hall/CRC.

Rue H, Martino S, Chopin N (2009). "Approximate Bayesian Inference for Latent Gaussian Models by Using Integrated Nested Laplace Approximations." *Journal of the Royal Statistical Society B*, **71**(2), 319–392. doi:10.1111/j.1467-9868.2008.00700.x.

Säfken B, Rügamer D, Kneib T, Greven S (2021). "Conditional Model Selection in Mixed-Effects Models with **cAIC4**." *Journal of Statistical Software*, **99**(8). doi:10.18637/jss.v099.i08.

Shelton AO, Thorson JT, Ward EJ, Feist BE (2014). "Spatial Semiparametric Models Improve Estimates of Species Abundance and Distribution." *Canadian Journal of Fisheries and Aquatic Sciences*, **71**(11), 1655–1666. doi:10.1139/cjfas-2013-0508.

Simpson D, Lindgren F, Rue H (2012). "In Order to Make Spatial Statistics Computationally Feasible, We Need to Forget about the Covariance Function." *Environmetrics*, **23**(1), 65–74. doi:10.1002/env.1137.

Simpson D, Rue H, Riebler A, Martins TG, Sørbye SH (2017). "Penalising Model Component Complexity: A Principled, Practical Approach to Constructing Priors." *Statistical Science*, **32**(1), 1–28. doi:10.1214/16-sts576.

Smith JQ (1985). "Diagnostic Checks of Non-Standard Time Series Models." *Journal of Forecasting*, **4**(3), 283–291. doi:10.1002/for.3980040305.

Stan Development Team (2025). ***rstan**: The R Interface to Stan*. doi:10.32614/CRAN.package.rstan. R package version 2.32.7.

Thorson JT (2018). "Three Problems with the Conventional Delta-Model for Biomass Sampling Data, and a Computationally Efficient Alternative." *Canadian Journal of Fisheries and Aquatic Sciences*, **75**(9), 1369–1382. doi:10.1139/cjfas-2017-0266.

Thorson JT (2019a). "Guidance for Decisions Using the Vector Autoregressive Spatio-Temporal (**VAST**) Package in Stock, Ecosystem, Habitat and Climate Assessments." *Fisheries Research*, **210**, 143–161. doi:10.1016/j.fishres.2018.10.013.

Thorson JT (2019b). "Measuring the Impact of Oceanographic Indices on Species Distribution Shifts: The Spatially Varying Effect of Cold-Pool Extent in the Eastern Bering Sea." *Limnology and Oceanography*, **64**(6), 2632–2645. doi:10.1002/lno.11238.

Thorson JT (2024). "Measuring Complexity for Hierarchical Models Using Effective Degrees of Freedom." *Ecology*, **105**(7), e4327. doi:10.1002/ecy.4327.

Thorson JT, Barnes CL, Friedman ST, Morano JL, Siple MC (2023). "Spatially Varying Coefficients Can Improve Parsimony and Descriptive Power for Species Distribution Models." *Ecography*, p. e06510. doi:10.1111/ecog.06510.

Thorson JT, Cheng W, Hermann AJ, Ianelli JN, Litzow MA, O'Leary CA, Thompson GG (2020). "Empirical Orthogonal Function Regression: Linking Population Biology to Spatial Varying Environmental Conditions Using Climate Projections." *Global Change Biology*, **26**(8), 4638–4649. doi:10.1111/gcb.15149.

Thorson JT, Ianelli JN, Larsen EA, Ries L, Scheuerell MD, Szuwalski C, Zipkin EF (2016). "Joint Dynamic Species Distribution Models: A Tool for Community Ordination and Spatio-Temporal Monitoring." *Global Ecology and Biogeography*, **25**(9), 1144–1158. doi:10.1111/geb.12464.

Thorson JT, Kristensen K (2016). "Implementing a Generic Method for Bias Correction in Statistical Models Using Random Effects, with Spatial and Population Dynamics Examples." *Fisheries Research*, **175**, 66–74. doi:10.1016/j.fishres.2015.11.016.

Thorson JT, Kristensen K (2024). *Spatio-Temporal Models for Ecologists.* Chapman & Hall/CRC, Boca Raton.

Thorson JT, Scheuerell MD, Shelton AO, See KE, Skaug HJ, Kristensen K (2015a). "Spatial Factor Analysis: A New Tool for Estimating Joint Species Distributions and Correlations in Species Range." *Methods in Ecology and Evolution*, **6**(6), 627–637. doi:10.1111/2041-210x.12359.

Thorson JT, Shelton AO, Ward EJ, Skaug HJ (2015b). "Geostatistical Delta-Generalized Linear Mixed Models Improve Precision for Estimated Abundance Indices for West Coast Groundfishes." *ICES Journal of Marine Science: Journal Du Conseil*, **72**(5), 1297–1310. doi:10.1093/icesjms/fsu243.

Thorson JT, Skaug HJ, Kristensen K, Shelton AO, Ward EJ, Harms JH, Benante JA (2015c). "The Importance of Spatial Models for Estimating the Strength of Density Dependence." *Ecology*, **96**(5), 1202–1212. doi:10.1890/14-0739.1.

Thygesen UH, Albertsen CM, Berg CW, Kristensen K, Nielsen A (2017). "Validation of Ecological State Space Models Using the Laplace Approximation." *Environmental and Ecological Statistics*, **24**(2), 317–339. doi:10.1007/s10651-017-0372-4.

Tweedie MCK (1984). "An Index Which Distinguishes between Some Important Exponential Families." In JK Ghosh, J Roy (eds.), *Statistics: Applications and New Directions. Proceedings of the Indian Statistical Institute Golden Jubilee International Conference*, pp. 579–604. Indian Statistical Institute, Calcutta.

Vehtari A, Gelman A, Gabry J (2017). "Practical Bayesian Model Evaluation Using Leave-One-Out Cross-Validation and WAIC." *Statistics and Computing*, **27**(5), 1413–1432. doi:10.1007/s11222-016-9696-4.

Ver Hoef JM, Peterson EE, Hooten MB, Hanks EM, Fortin MJ (2018). "Spatial Autoregressive Models for Statistical Inference from Ecological Data." *Ecological Monographs*, **88**(1), 36–59. doi:10.1002/ecm.1283.

Waagepetersen R (2006). "A Simulation-Based Goodness-of-Fit Test for Random Effects in Generalized Linear Mixed Models." *Scandinavian Journal of Statistics*, **33**(4), 721–731. doi:10.1111/j.1467-9469.2006.00504.x.

Wall MM (2004). "A Close Look at the Spatial Structure Implied by the CAR and SAR Models." *Journal of Statistical Planning and Inference*, **121**(2), 311–324. doi:10.1016/s0378-3758(03)00111-3.

Ward EJ, Barnett LAK, Anderson SC, Commander CJC, Essington TE (2022). "Incorporating Non-Stationary Spatial Variability into Dynamic Species Distribution Models." *ICES Journal of Marine Science*, **79**(9), 2422–2429. doi:10.1093/icesjms/fsac179.

Whittle P (1954). "On Stationary Processes in the Plane." *Biometrika*, **41**(3/4), 434. doi:10.2307/2332724.

Wickham H (2016). ***ggplot2**: Elegant Graphics for Data Analysis*. Springer-Verlag. doi:10.1007/978-0-387-98141-3.

Wood S, Scheipl F (2025). ***gamm4**: Generalized Additive Mixed Models Using **mgcv** and **lme4***. doi:10.32614/CRAN.package.gamm4. R package version 0.2-7.

Wood SN (2017). *Generalized Additive Models: An Introduction with R*. 2nd edition. Chapman & Hall/CRC.

Wood SN, Goude Y, Shaw S (2015). "Generalized Additive Models for Large Data Sets." *Journal of the Royal Statistical Society C*, **64**(1), 139–155. doi:10.1111/rssc.12068.

Wood SN, Li Z, Shaddick G, Augustin NH (2017). "Generalized Additive Models for Gigadata: Modeling the U.K. Black Smoke Network Daily Data." *Journal of the American Statistical Association*, **112**(519), 1199–1210. doi:10.1080/01621459.2016.1195744.

Xianyi Z, Qian W, Yunquan Z (2012). "Model-Driven Level 3 BLAS Performance Optimization on Loongson 3A Processor." In *2012 IEEE 18th International Conference on Parallel and Distributed Systems*, pp. 684–691. doi:10.1109/icpads.2012.97.

Zhang R, Czado C, Sigloch K (2015). "Bayesian Spatial Modelling for High Dimensional Seismic Inverse Problems." *Journal of the Royal Statistical Society C*, **65**(2), 187–213. doi:10.1111/rssc.12118.

Zheng N, Cadigan N (2021). "Frequentist Delta-Variance Approximations with Mixed-Effects Models and **TMB**." *Computational Statistics & Data Analysis*, **160**, 107227. doi:10.1016/j.csda.2021.107227.

# A. Speed testing related packages

Here, we describe the methods underlying the speed testing in Figure 12. We generated a mesh that was consistent across simulated data sets for a given mesh resolution (Figure 14). We did this by setting the `max.edge` argument, which controls the largest allowed triangle edge length. We tested values of `max.edge` of 0.06, 0.075, 0.1, 0.15, and 0.2. In Figure 12, we report on the x-axis the number of mesh nodes ("vertices") that result from each of these meshes.

We simulated 1,000, 10,000, or 100,000 spatial observations with both x and y coordinates from uniform(0, 1) distributions (Figure 15). Each iteration generated unique coordinates, predictor data, GMRF values, and observation error. The GMRF was parameterized with a range of 0.5 and marginal standard deviation of 0.2. The model included an intercept with a value of 0.2 and a normal(0, 1) predictor with an associated coefficient of -0.4. The observation error was Gaussian with a standard deviation of 0.3.

We conducted 50 simulation iterations per model and mesh resolution and show timing results for the mean, 10%, and 90% quantile values. The models were fit on a 2023 16-inch M2 MacBook Pro with an Apple M2 Pro 10-core CPU and 32 GB of RAM in R 4.4.0 and the default **BLAS** library packaged with R (R Core Team 2025).

## A.1. Illustration of generating the INLA mesh

We will illustrate with a `max.edge` of 0.06:

```
R> max_edge <- 0.06
R> loc_bnd <- matrix(c(0, 0, 1, 0, 1, 1, 0, 1), 4, 2, byrow = TRUE)
R> segm_bnd <- INLA::inla.mesh.segment(loc_bnd)
R> mesh <- INLA::inla.mesh.2d(boundary = segm_bnd,
+    max.edge = c(max_edge, 0.2), offset = c(0.1, 0.05))
```

This mesh has 1165 (`mesh$n`) vertices.

## A.2. Illustration of simulating data

```
R> set.seed(123)
R> n_obs <- 1000
R> predictor_dat <- data.frame(X = runif(n_obs), Y = runif(n_obs),
+    a1 = rnorm(n_obs))
R> mesh_sdmTMB <- make_mesh(predictor_dat, xy_cols = c("X", "Y"),
+    mesh = mesh)
R> sim_dat <- simulate_new(formula = ~ 1 + a1, data = predictor_dat,
+    mesh = mesh_sdmTMB, family = gaussian(),
+    range = 0.5, phi = 0.3, sigma_O = 0.2, B = c(0.2, -0.4))
```

## A.3. Example sdmTMB model fit

```
R> fit_sdmTMB <- sdmTMB(observed ~ a1, data = sim_dat, mesh = mesh_sdmTMB,
+    family = gaussian(), priors =
+      sdmTMBpriors(matern_s = pc_matern(range_gt = 0.05, sigma_lt = 2)))
```

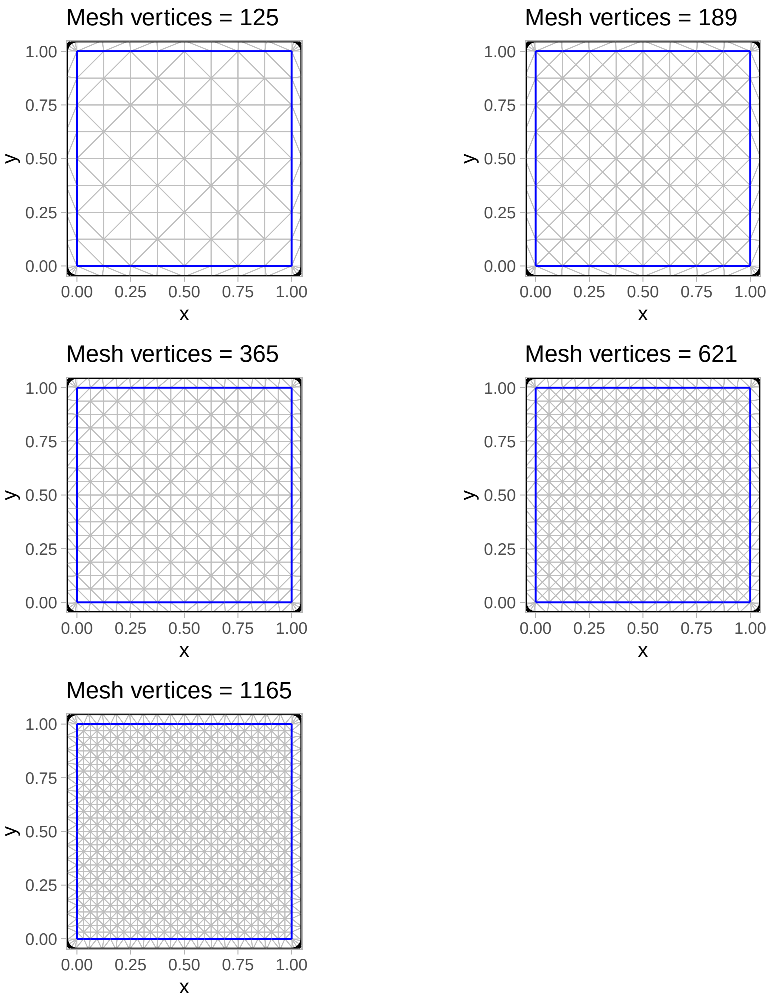


Figure 14: The meshes used in simulations from least to most vertices.

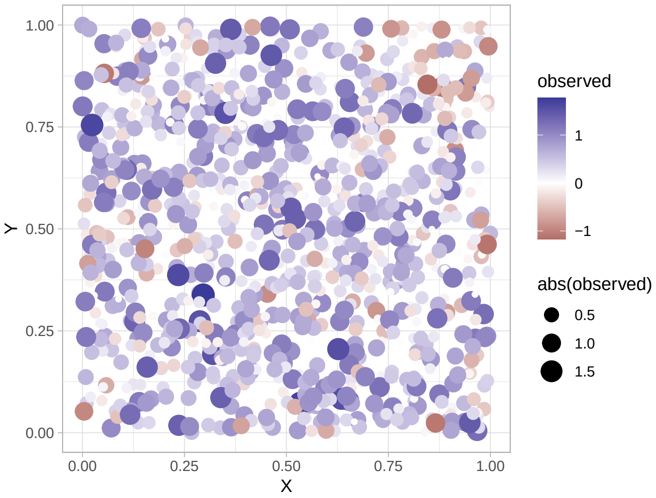


Figure 15: Example simulated dataset with a spatial random field.

## A.4. Example spaMM model fit

```
R> spde <- INLA::inla.spde2.pcmatern(mesh = mesh, prior.range = c(0.05, 0.05),
+    prior.sigma = c(2, 0.05))
R> fit_spaMM <- fitme(observed ~ a1 + IMRF(1 | X + Y, model = spde),
+    family = gaussian(), data = sim_dat)
```

## A.5. Example inlabru model fit

```
R> dat_sp <- sp::SpatialPointsDataFrame(cbind(sim_dat$X, sim_dat$Y),
+    proj4string = sp::CRS(
+      "+proj=aea +lat_0=45 +lon_0=-126 +lat_1=50 +lat_2=58.5 +x_0=1000000
+    +y_0=0 +datum=NAD83 +units=km +no_defs"
+    ), data = sim_dat)
R> components <- observed ~ -1 + Intercept(1) + a1 +
+    spatrf(main = coordinates, model = spde)
R> like <- like(observed ~ Intercept + a1 + spatrf,
+    family = "gaussian", data = dat_sp)
R> fit_bru <- bru(like, components = components,
+    options = bru_options(
+      control.inla = list(int.strategy = "eb", strategy = "gaussian"),
+      bru_max_iter = 1, num.threads = "1:1"
+    ))
```

## A.6. Example mgcv model fit

First define `smooth.construct.spde.smooth.spec()` and `Predict.matrix.spde.smooth()` from the supplement of Miller *et al.* (2020), then:

```
R> fit_bam <- bam(observed ~ a1 +
+      s(X, Y, bs = "spde", k = mesh$n, xt = list(mesh = mesh)),
+    data = sim_dat, family = gaussian(), method = "fREML",
+    control = gam.control(scalePenalty = FALSE), discrete = TRUE)
```

**Affiliation:**

Sean C. Anderson, Philina A. English
Pacific Biological Station
Fisheries and Oceans Canada
Nanaimo, BC, Canada
E-mail: sean.anderson@dfo-mpo.gc.ca, philina.english@dfo-mpo.gc.ca

Eric J. Ward
Northwest Fisheries Science Center
National Oceanic and Atmospheric Administration
National Marine Fisheries Service
Seattle, WA, United States of America
E-mail: eric.ward@noaa.gov

Lewis A. K. Barnett, James T. Thorson
Alaska Fisheries Science Center
National Marine Fisheries Service
National Oceanic and Atmospheric Administration
Seattle, WA, United States of America
E-mail: lewis.barnett@noaa.gov, james.thorson@noaa.gov

*Journal of Statistical Software* https://www.jstatsoft.org/
published by the Foundation for Open Access Statistics https://www.foastat.org/
December 2025, Volume 115, Issue 2 *Submitted:* 2023-02-14
doi:10.18637/jss.v115.i02 *Accepted:* 2024-08-28