

Misha Tseitliani

25 November 2025

▼ Using INLA-SPDE for Spatial Models in R



Overview

- Supplement to notes that are available on [GitHub](#) (and Moodle) →
- HTML notes are **much more comprehensive**
- This presentation covers (not in that order)
 - Method overview
 - Common pitfalls
 - Some example syntax
- Look at notes if time allows

An Introduction to INLA-SPDE in R

Misha Tseitliani

2025-11-25

Background

At this point, we have gone through a number of methods for processing spatial data. Many of these were **non-parametric** or **semi-parametric**: this means that it can be difficult to link specific changes in a parameter or covariate to the intended result. In addition, many have been **spatially implicit**: they use coordinates as a proxy for spatial structure rather than modelling some *true space*. INLA can do all these too but is especially powerful for **parametric** and **spatially explicit** models. If you want to conduct inference on your data, then this might be perfect for you!

Definitions

Before going too deep into the package details, first we need to ensure we're familiar with the language of spatial statistics. The point is not to teach theory (which will mostly be omitted) but to ensure we have a *shared language* when questions arise. If you *are* interested in the theory, there are numerous good [resources](#) out there.¹⁻³

INLA is theoretically inspired by hierarchical and state space models and has both **latent** and **observed** processes:

- **latent process**: the *true* environmental reality (e.g., number of species in a sampling plot). Also called a **state process**.
- **observation process**: what we see with our data (e.g., the number of species *recorded* in a sampling plot). With a normal model, this is what you are looking at.

With that sorted, spatial modelling has vocabulary corresponding to the geospatial language you know (and maybe love):

- **(Poisson) point process**: a 2D distribution for **point vector data** and the basis of many geospatial models.
- **Cox process**: a doubly stochastic Poisson process for **point vector data** where the intensity (i.e., expected number of points in a specific area) is itself stochastically random. INLA has special functions for Cox and LGCP (log-Gaussian Cox process) models.
- **Gaussian random field**: a continuous 2D field for **raster data** or **fully defined vector polygons**. In essence, fields are the **latent process** from which you draw point process distributions (via sampling).
- **Gauss-Markov random field (GMRF)**: a Gaussian random field for **raster data** or **fully defined vector polygons** where values structurally depend on some specific distance, time, etc (i.e., Markovian dependencies) but are otherwise independent.

To reiterate, INLA's spatial models are directly based on GMRFs, and point process models are treated as drawn from some latent GMRF. This means that you can always include raster data (e.g., as covariates) directly in your models even if your response is vector data.

So, INLA (and SPDE)?

Integrated Nested Laplace Approximations (INLA) are a method of estimating Bayesian models that run faster than MCMC (Markov Chain Monte Carlo) while still generating similar results. Spatial models in the `INLA` package use the **SPDE (Stochastic Partial Differential Equations)** approach to better fit spatial structures: hence the name INLA-SPDE. These are fit over a spatial mesh using splines to run faster. In practice, users need only specify a spatial mesh and regression formulation (i.e., INLA's models are linear additive regression models) to get results.

If you plan to be a casual user at best, you can consider the basic SPDE field as a 2D random effect controlling for spatial patterns and move on.

Assumptions

What is it?

- Integrated Nested Lagrange Approximation (**INLA**) – 2009
 - Faster alternative to Bayesian estimation than MCMC
- Stochastic Partial Differential Equations (**SPDE**) – 2011
 - Computational method for spatial models over a defined area
- Implemented with three packages in R
 - INLA (2009) – original interface
 - inlabru (2019) – updated cleaner syntax & defaults
 - fmesher (2023) – split off from INLA; area triangulation
- Used most in ecology, health studies (e.g., epidemiology), and environmental studies (e.g., spatial econometrics)

What is it for?

- Main use case is spatial modelling
 - Random fields – a continuous latent process in 1+ dimensions that correspond to raster or vector polygon data
 - Point process models – point vector processes that can be either marked or unmarked and are realised from some underlying latent random field
- Good temporal (and spatio-temporal) functionality too
- Meant for fast fitting to enable more iterative model development: when in doubt, fit a model.
 - Simple syntax for compact models and small datasets
 - Complex syntax (incl. parallelisation/HPC) for bigger models & data

Possibilities

- All models are **Bayesian**. They can include:
 - GLM (generalised linear models)
 - GLMMs (generalised linear mixed models)
 - GEV (generalised extreme value)
 - ARMA (autoregressive moving average)
 - LGCP (log Gaussian Cox process)
 - SVC (Spatially varying covariates)
 - Random field models
 - Barrier models
 - Joint multiple likelihoods
 - Bayesian melding model (i.e., data fusion)

Assumptions

- There are no real consistent assumptions across all model types
- In spatial models,
 - Stationarity – a weaker assumption that the underlying distribution in the latent field is consistent and properly specified across the entire sample area
 - Isotropy – a strong assumption (with some exceptions) that our spatial structures are rotation-invariant
- Any INLA model with multiple terms in a single likelihood is a generalised linear additive model
 - Recall your GLM(M) assumptions (e.g., mean-variance relationships)
- Every model has its own assumptions (e.g., temporal stationarity)



Spatial Modelling A-Z

- Clean data, make spatial, and rescale
- Build the mesh using fmesher
- Construct special model components (i.e., SPDE)
- Specify the model formula(e)
- Run the fit using INLA/inlabru
- Inspect model object; interpret results
- Conduct posterior checks

Data Structure

- INLA cannot accommodate NAs in the underlying mesh or response variables
- Preference for tidy dataframes and sf/terra spatial objects
- Single likelihoods: data is usually in **wide format** for random field models but can be **long format** for point process models
- NAs in covariates are OK, and some special effects (e.g., raster covariates) take their own, separate datasets
- Be careful of collinearity for **inference** in linear models
- Set aside a validation/test set or simulated data to do score computation for **prediction**

2D Mesh & SPDE

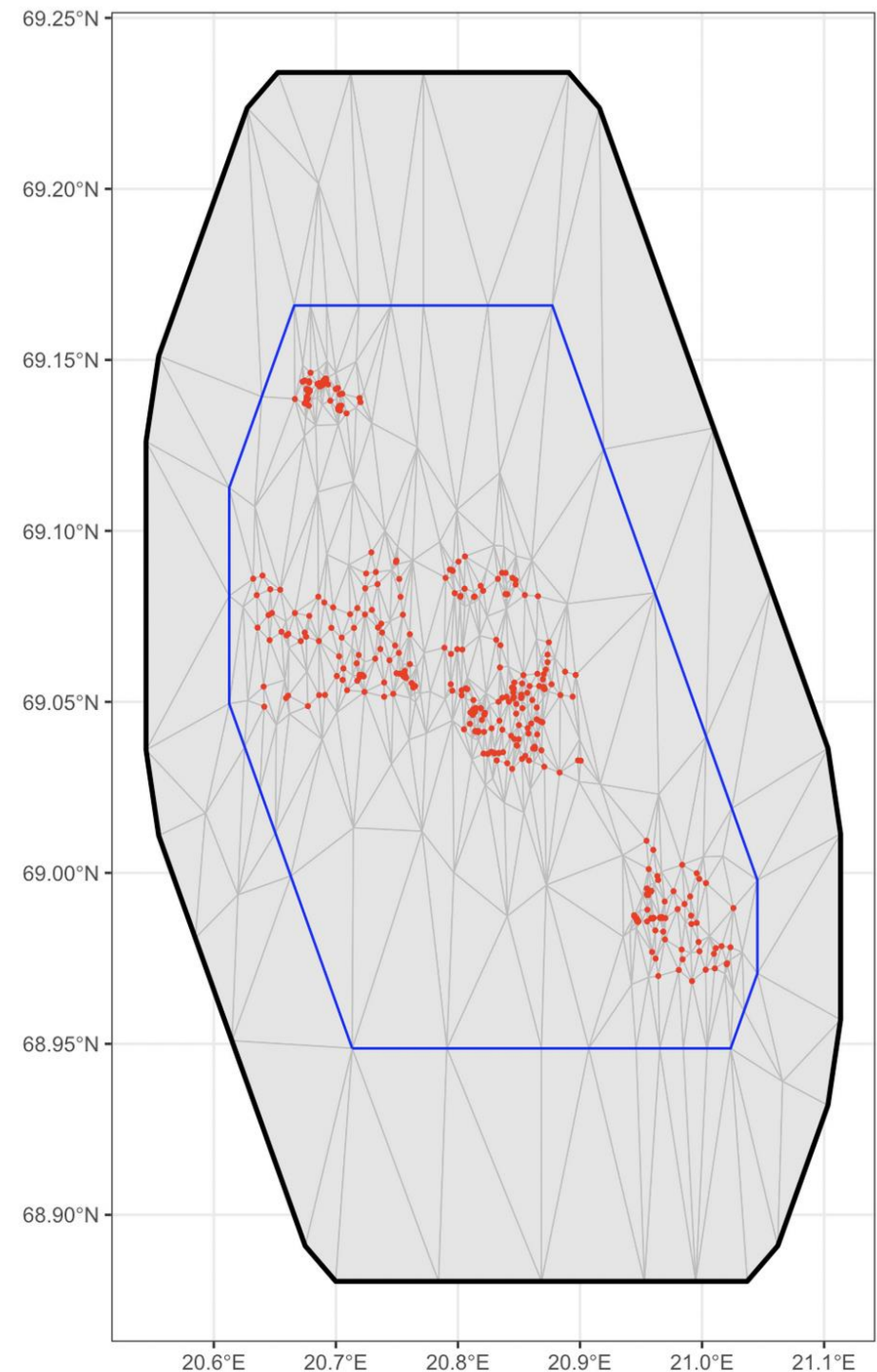
```
meshBasic <- fm_mesh_2d_inla(  
  # points sf object  
  loc = kjGeo$geometry,  
  # set to account for boundary effects  
  max.edge = c(0.9, # smaller inner units give a more detailed mesh within the sampling area  
               1), # smaller outer units give more detailed spatial estimates of boundary effects  
  crs = fm_crs(kjGeo)  
)  
  
# visually confirm  
meshPlot <- ggplot() +  
  # fmesher plotting function  
  geom_fm(data = meshBasic) +  
  # overlay actual points  
  geom_sf(data = kjGeo, size = 0.5, colour = "red") +  
  theme_bw()  
meshPlot  
  
# define Matern correlation on the mesh for SPDE  
matern2D <- inla.spde2.matern(mesh = meshBasic)
```

2D Mesh & SPDE

```
meshBasic <- fm_mesh_2d_inla(  
  # points sf object  
  loc = kjGeo$geometry,  
  # set to account for boundary effects  
  max.edge = c(0.9, # smaller inner units give a more detail  
               1), # smaller outer units give more detailed  
  crs = fm_crs(kjGeo)  
)
```

```
# visually confirm  
meshPlot <- ggplot() +  
  # fmesher plotting function  
  geom_fm(data = meshBasic) +  
  # overlay actual points  
  geom_sf(data = kjGeo, size = 0.5, colour = "red") +  
  theme_bw()  
meshPlot
```

```
# define Matern correlation on the mesh for SPDE  
matern2D <- inla.spde2.matern(mesh = meshBasic)
```



Model Formula & Run

```
# write the formula
formulaGeo <- species_rich ~ Intercept(1) + cwm_max_height + cwm_LDMC +
  FDD_T3 + GDD3_T3 + median_moist +
# new stuff
  field(geometry, model = matern2D)
# to demonstrate the syntax for adding a second effect
# + field2(site, model = "iid")

# run the model
fitGeo <- bru(formulaGeo, kjGeoNorm, family = "tweedie")
```

Model Inspection & Checks

- Heavily dependent on model type and analysis aims
- Use `summary()`, residuals, information criteria (i.e., WAIC, DIC)
Make sure to interpret accounting for the model link function and Bayesian methodology
- Check hyperparameters and priors
- Generating predictions is harder but best practice

[illegible]

