Embedding stochastic PDEs in Bayesian spatial statistics software

Finn Lindgren
finn.lindgren@ed.ac.uk

NASPDE, EURANDOM, Eindhoven, 17 May 2023

GAMs and general kriging

- Linear GAMs with GPs on space and covariates:

$$\boldsymbol{\eta}_i = \sum_k v_k(z_{ik}) + u(\mathbf{s}_i),$$

each $v_k(\cdot)$ and $u(\cdot)$ represented with basis expansions with jointly Gaussian coefficients $\boldsymbol{x}$.

- Linear observations with additive Gaussian observation noise: $\boldsymbol{y} = \boldsymbol{\eta} + \boldsymbol{\epsilon} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\epsilon}$
- Covariance kriging

$$\boldsymbol{\Sigma}_{\boldsymbol{y}} = \boldsymbol{A}\boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{A}^\top + \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}$$
$$\mathsf{E}(\boldsymbol{x}|\boldsymbol{y}) = \boldsymbol{\mu} + \boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{A}^\top\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\mu})$$

- Precision kriging

$$\boldsymbol{Q}_{\boldsymbol{x}|\boldsymbol{y}} = \boldsymbol{Q}_{\boldsymbol{x}} + \boldsymbol{A}^\top\boldsymbol{Q}_{\boldsymbol{\epsilon}}\boldsymbol{A}$$
$$\mathsf{E}(\boldsymbol{x}|\boldsymbol{y}) = \boldsymbol{\mu} + \boldsymbol{Q}_{\boldsymbol{x}|\boldsymbol{y}}^{-1}\boldsymbol{A}^\top\boldsymbol{Q}_{\boldsymbol{\epsilon}}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\mu})$$

- Non-Gaussian observations with link function: $\mathsf{E}(y_i|\boldsymbol{\theta}, \boldsymbol{x}) = h(\eta_i)$

Observation level covariance vs latent level precision

- Covariance kriging: linear solve with a $\boldsymbol{\Sigma}$, $\Sigma_{ij} = \text{Cov}(y_i, y_j)$
- Precision kriging: linear solve with a $\boldsymbol{Q}$, $Q_{ij} = \text{Prec}(x_i, x_j|\boldsymbol{y})$
  $\boldsymbol{Q} = \boldsymbol{L}\boldsymbol{L}^\top$ for a given latent variable ordering, and sparse lower triangular L with the sparsity from $\boldsymbol{Q}$ plus Cholesky infill.
  The prior $\boldsymbol{Q}_x$ for GRF/SPDE process components are obtained via a local Finite Element construction, giving the model in a chosen finite function space closest to the full model.

Finite element structure

Matérn-Whittle processes

Linear Gaussian process/field representations via SPDEs:

$$(\kappa^2 - \Delta)^\alpha u(\mathbf{s})\,\mathrm{d}\mathbf{s} = \mathrm{d}\mathcal{W}(\mathbf{s})\kappa^{\alpha-d/2}/\tau$$

For constant parameters, $u(\mathbf{s})$ has spatial Matérn covariance on $\mathbb{R}^d$, and generalised Matérn-Whittle covariance on general manifolds. The smoothness index is $\nu = \alpha - d/2$ and the variance is proportional to $1/\tau^2$. Whittle (1954, 1963), Lindgren et al (2011)

Discrete domain Gaussian Markov random fields (GMRFs)

$\boldsymbol{x} = (x_1, \ldots, x_n) \sim \mathrm{N}(\boldsymbol{\mu}, \boldsymbol{Q}^{-1})$ is Markov with respect to a neighbourhood structure $\{\mathcal{N}_i, i = 1, \ldots, n\}$ if $Q_{ij} = 0$ whenever $j \neq \mathcal{N}_i \cup i$.

- Continuous domain basis representation with weights: $x(\boldsymbol{s}) = \sum_{k=1}^n \psi_k(\boldsymbol{s}) u_k$
- Project the SPDE solution space onto local basis functions: random Markov dependent basis weights (Lindgren et al, 2011).

Non-stationarity

Non-stationary Matérn-Whittle processes

The Sampson & Guttorp (1992) deformation method motivates a non-stationary generalisation on $\mathbb{R}^2$:

$$(\kappa(\mathrm{s})^2 - \nabla \cdot \boldsymbol{H}(\mathrm{s})\nabla)^\alpha \frac{u(\mathrm{s})}{\sigma(\mathrm{s})}\,\mathrm{d}\mathrm{s} = \mathrm{d}\mathcal{W}(\mathrm{s})\kappa(\mathrm{s})^{\alpha-d/2},$$

where $\kappa(\mathrm{s})$ and $\boldsymbol{H}(\mathrm{s})$ are derived from the metric tensor of the deformation. For deformation *not* from $\mathbb{R}^d$ onto $\mathbb{R}^d$, this non-stationary model is distinct from the deformation method, but keeps much of the intuition, as the variance will be approximativey independent of $\kappa(\mathrm{s})$.

RKHS inner products of linear SPDEs

The spatial solutions $u(\mathrm{s})$ to

$$\mathcal{L}u(\mathrm{s})\,\mathrm{d}\mathrm{s} = \mathrm{d}\mathcal{W}(\mathrm{s}) \quad \text{where } \mathrm{d}\mathcal{W}(\mathrm{s}) \text{ is white noise on } \Omega$$

have RKHS inner product

$$\mathcal{Q}_\Omega(f,g) = \langle \mathcal{L}f, \mathcal{L}g\rangle_\Omega$$

plus potential boundary terms.

Non-separable space-time: Matérn driven heat equation on the sphere

The iterated heat equation is a simple non-separable space-time SPDE family:

$$\left[\phi\frac{\partial}{\partial t} + (\kappa^2 - \Delta)^{\alpha_s/2}\right]^{\alpha_t} u(\mathbf{s},t)\,\mathrm{d}t = \mathrm{d}\mathcal{E}_{(\kappa^2-\Delta)^{\alpha_e}}(\mathbf{s},t)/\tau$$

For constant parameters, $u(\mathbf{s},t)$ has spatial Matérn covariance (for each t) on $\mathbb{R}^2$ and a generalised Matérn-Whittle sense on $\mathbb{S}^2$.

Smoothness properties:

$$\nu_t = \min\left[\alpha_t - \frac{1}{2}, \frac{\nu_s}{\alpha_s}\right], \qquad \alpha_t = \nu_t \max\left(1, \frac{\beta_s}{\beta_*(\nu_s,d)}\right) + \frac{1}{2},$$

$$\nu_s = \alpha_e + \alpha_s(\alpha_t - \frac{1}{2}) - \frac{d}{2}, \quad \alpha_s = \frac{\nu_s}{\nu_t}\min\left(\frac{\beta_s}{\beta_*(\nu_s,d)}, 1\right) = \frac{1}{\nu_t}\min\left[(\nu_s + d/2)\beta_s, \nu_s\right],$$

$$\beta_*(\nu_s,d) = \frac{\nu_s}{\nu_s + d/2}, \qquad \alpha_e = \frac{1-\beta_s}{\beta_*(\nu_s,d)}\nu_s = (\nu_s + d/2)(1-\beta_s).$$

Spectra and finite element structure

- Fourier spectra are based on eigenfunctions $e_{\boldsymbol{\omega}}(\mathbf{s})$ of $-\Delta$.
  On $\mathbb{R}^d$, $-\Delta e_{\boldsymbol{\lambda}}(\mathbf{s}) = \|\boldsymbol{\lambda}\|^2 e_{\boldsymbol{\lambda}}(\mathbf{s})$, and $e_{\boldsymbol{\lambda}}(\mathbf{s})$ are harmonic functions.
- The stationary spectrum on $\mathbb{R}^d \times \mathbb{R}$ is

$$\widehat{\mathcal{R}}(\boldsymbol{\lambda}, \omega) = \frac{1}{(2\pi)^{d+1}\tau^2(\kappa^2 + \lambda_{\boldsymbol{\lambda}})^{\alpha_e}\left[\phi^2\omega^2 + (\kappa^2 + \lambda_{\boldsymbol{\lambda}})^{\alpha_s}\right]^{\alpha_t}}$$

- On $\mathbb{S}^2$, $-\Delta e_k(\mathbf{s}) = \lambda_k e_k(\mathbf{s}) = k(k+1)e_k(\mathbf{s})$, and e_k are spherical harmonics.
- The isotropic spectrum on $\mathbb{S}^2 \times \mathbb{R}$ is

$$\widehat{\mathcal{R}}(k, \omega) \propto \frac{2k+1}{\tau^2(\kappa^2 + \lambda_k)^{\alpha_e}\left[\phi^2\omega^2 + (\kappa^2 + \lambda_k)^{\alpha_s}\right]^{\alpha_t}}$$

- The finite element approximation has structure

$$u(\boldsymbol{s}, t) = \sum_{i,j} \psi_i^{[s]}(\boldsymbol{s})\psi_j^{[t]}(t)x_{ij}, \quad \boldsymbol{x} \sim \mathsf{N}(\boldsymbol{0}, \boldsymbol{Q}^{-1}), \quad \boldsymbol{Q} = \sum_{k=0}^{\alpha_t+\alpha_s+\alpha_e} \boldsymbol{M}_k^{[t]} \otimes \boldsymbol{M}_k^{[\mathbf{s}]}$$

  even, e.g., if the spatial scale parameter κ is spatially varying.

Latent Gaussian models

Hierarchical model with latent jointly Gaussian variables

$$\boldsymbol{\theta} \sim p(\boldsymbol{\theta}) \quad \text{(covariance parameters)}$$

$$(\boldsymbol{u} \mid \boldsymbol{\theta}) \sim \mathsf{N}(\boldsymbol{\mu}_u, \boldsymbol{Q}_u^{-1}) \quad \text{(latent Gaussian variables)}$$

$$(\boldsymbol{y} \mid \boldsymbol{u}, \boldsymbol{\theta}) \sim p(\boldsymbol{y} \mid \boldsymbol{u}, \boldsymbol{\theta}) \quad \text{(observation model)}$$

We are interested in the posterior densities $p(\boldsymbol{\theta} \mid \boldsymbol{y})$, $p(\boldsymbol{u} \mid \boldsymbol{y})$ and $p(u_i \mid \boldsymbol{y})$.

Approximate conditional posterior distribution

Let $\widehat{\boldsymbol{u}}(\boldsymbol{\theta})$ be the mode of the posterior density $p(\boldsymbol{u} \mid \boldsymbol{y}, \boldsymbol{\theta}) \propto p(\boldsymbol{u} \mid \boldsymbol{\theta})p(\boldsymbol{y} \mid \boldsymbol{u}, \boldsymbol{\theta})$. Construct an approximate conditional posterior distribution, via Newton optimisation for $\boldsymbol{u}$ given $\boldsymbol{\theta}$:

$$p_G(\boldsymbol{u} \mid \boldsymbol{y}, \boldsymbol{\theta}) \sim \mathsf{N}(\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{Q}}^{-1})$$

$$\boldsymbol{0} = \nabla_{\boldsymbol{u}} \left\{ \ln p(\boldsymbol{u} \mid \boldsymbol{\theta}) + \ln p(\boldsymbol{y} \mid \boldsymbol{u}, \boldsymbol{\theta}) \right\} \Big|_{\boldsymbol{u}=\widehat{\boldsymbol{\mu}}(\boldsymbol{\theta})}$$

$$\widehat{\boldsymbol{Q}} = \boldsymbol{Q}_u - \nabla_{\boldsymbol{u}}^2 \ln p(\boldsymbol{y} \mid \boldsymbol{u}, \boldsymbol{\theta}) \Big|_{\boldsymbol{u}=\widehat{\boldsymbol{\mu}}_{\boldsymbol{\theta}}}$$

Classic and compact INLA methods ($\sim$ description)

- Laplace approximation at the conditional posterior mode $\boldsymbol{x}^*$, and uncertainty integration:

$$p(\boldsymbol{\theta}|\boldsymbol{y}) \propto \left.\frac{p(\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{\theta},\boldsymbol{x})}{p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})}\right|_{\boldsymbol{x}=\boldsymbol{x}^*} \approx \left.\frac{p(\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{\theta},\boldsymbol{x})}{p_G(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})}\right|_{\boldsymbol{x}=\boldsymbol{x}^*} = \widehat{p}(\boldsymbol{\theta}|\boldsymbol{y})$$

$$p(x_i|\boldsymbol{y}) = \int p(x_i|\boldsymbol{\theta},\boldsymbol{y})p(\boldsymbol{\theta}|\boldsymbol{y})\,\mathrm{d}\boldsymbol{\theta} \approx \sum_k \widehat{p}(x_i|\boldsymbol{\theta}^{(k)},\boldsymbol{y})\widehat{p}(\boldsymbol{\theta}^{(k)}|\boldsymbol{y})w_k = \widehat{p}(x_i|\boldsymbol{y})$$

- Let $\widehat{\boldsymbol{\mu}} = \mathsf{E}(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})$ and $\boldsymbol{Q}_\epsilon = -\nabla_x\nabla_x^\top \log p(\boldsymbol{y}|\boldsymbol{\theta},\boldsymbol{x}^*)$
- Classic method: Laplace approximation of each $\widehat{p}(x_i|\boldsymbol{\theta},\boldsymbol{y})$, and
$$\left\{\begin{bmatrix}\boldsymbol{A}\boldsymbol{x}\\ \boldsymbol{x}\end{bmatrix}\middle|\boldsymbol{\theta},\boldsymbol{y}\right\} \sim \mathsf{N}\left(\begin{bmatrix}\boldsymbol{A}\widehat{\boldsymbol{\mu}}\\ \widehat{\boldsymbol{\mu}}\end{bmatrix}, \begin{bmatrix}\boldsymbol{Q}_\epsilon+\delta\boldsymbol{I} & -\delta\boldsymbol{A}\\ -\delta\boldsymbol{A}^\top & \boldsymbol{Q}_x+\delta\boldsymbol{A}^\top\boldsymbol{A}\end{bmatrix}^{-1}\right)$$, with $\delta \gg 0$
- Compact method: Variational approximation of $\widehat{p}(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})$, and
$\{\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y}\} \sim \mathsf{N}\left(\widehat{\boldsymbol{\mu}}, [\boldsymbol{Q}_x + \boldsymbol{A}^\top\boldsymbol{Q}_\epsilon\boldsymbol{A}]^{-1}\right)$

inlabru software interface concepts

- Model components are declared similarly to R-INLA:

```
# INLA:
~ covar + f(name, model = ...)
# inlabru
~ covar + name(input, model = ...)
~ covar # is translated into...
~ covar(covar, model = "linear")
~ name(1) # Used for intercept-like components
```

- In R-INLA, $\boldsymbol{\eta} = \boldsymbol{A}\boldsymbol{u} = \boldsymbol{A}_0 \sum_{k=1}^{K} \boldsymbol{A}_k \boldsymbol{u}_k$, where the rows of $\boldsymbol{A}_k$ only extract individual elements from each $\boldsymbol{u}_k$, and the overall $\boldsymbol{A}_0$ is user defined (via `inla.stack()`).
- In inlabru, $\boldsymbol{\eta} = h(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{\boldsymbol{K}}, \boldsymbol{A}_1\boldsymbol{u}_1, \ldots, \boldsymbol{A}_K\boldsymbol{u}_{\boldsymbol{K}})$, where $h(\cdot)$ is a general R expression of named latent components $\boldsymbol{u}_k$ and intermediate "effects" $\boldsymbol{A}_k\boldsymbol{u}_k$
- $\boldsymbol{A}_k$ by default acts either as in R-INLA, or is determined by a *mapper* method. Predefined default mappers include e.g. spatial evaluation of SPDE/GRMF models that map between coordinates and meshes, and mappers that combine other mappers (used to combine main/group/replicate for all components)

Input mappers

- Each named component has main/group/replicate *inputs*, that are given to the mappers to evaluate $\boldsymbol{A}_k$. For a given latent *state*, the resulting *effect* values are made available to the predictor expression.

```
bru_mapper() # generic
bru_mapper_index(n) # Basic index mapping
bru_mapper_linear() # Basic linear mapping
bru_mapper_matrix(labels) # Basic linear multivariable mapping
bru_mapper_factor(values, factor_mapping) # Factor variable mapping
bru_mapper_multi(mappers) # kronecker product components
bru_mapper_collect(mappers, hidden) # For concatenated components, like bym
bru_mapper_const() # Constants
bru_mapper.inla.mesh(mesh) # 2D and spherical mesh mappings
bru_mapper.inla.mesh.1d(mesh) # Interval and cyclic interval mappings
```

- Common methods that return essential characteristics

```
ibm_n(mapper) # The size of the latent component
ibm_values(mapper) # The covariate/index "values" given to INLA
ibm_jacobian(mapper, input) # The "A-matrix" for given input values
```

- Model component definition example:

```
comp <- ~ -1 + field(cbind(easting, northing), model = spde) + param(1)
```

- Predictor formula examples, including naming of the response variable:

```
form1 <- my_counts ~ param + field
form2 <- response ~ exp(param) + exp(field)
```

- Main method call structure:

```
bru(components = comp,
    like(formula = form1, family = "poisson", data = data1),
    like(formula = form2, family = "normal", data = data2))
```

- Simplified notations for common special cases;

  `formula = response ~ .`

  gives the full additive model of all the available components, or

  `components = response ~ Intercept(1) + field(...`

Plain INLA code for space-time model

```
matern <- inla.spde2.pcmatern(mesh, ...)

field_A <- inla.spde.make.A(mesh,
                            coordinates(data),
                            group = data$year - min(data$year) + 1,
                            n.group = 10)
stk <- inla.stack(data = list(response = data$response),
                  A = list(field_A, 1),
                  effects = list(field_index, list(covar = data$covar)))

formula <- response ~ 1 + covar +
  f(field, model = matern, group = field_group, control.group = ...)

fit <- inla(formula = formula,
            data = inla.stack.data(stk, matern = matern),
            family = "normal",
            control.predictor = list(A = inla.stack.A(stk)))
```

inlabru code for space-time model

```
matern <- inla.spde2.pcmatern(mesh, ...)

year_mapper <- bru_mapper(inla.mesh.1d(sort(unique(data$year))), indexed = TRUE)

comp <- response ~ Intercept(1) + covar +
  field(coordinates, model = matern, group = year, group_mapper = year_mapper,
        control.group = ...)

fit <- bru(components = comp,
           data = data,
           family = "normal")
```

Implied:

- `coordinates` $\longrightarrow$ `sp::coordinates(.data.)`
- `formula = response ~ .`
- `data` and `family` passed on to a `like()` call

Latent Gaussian models of R-INLA type and inlabru extension

LGM of R-INLA type

$$\boldsymbol{\theta} \sim p(\boldsymbol{\theta}), \quad \text{(hyper-)parameters}$$

$$\boldsymbol{u}|\boldsymbol{\theta} \sim \mathsf{N}\left(\boldsymbol{\mu}_u, \boldsymbol{Q}(\boldsymbol{\theta})^{-1}\right), \quad \text{complex structured latent Gaussian field}$$

$$\boldsymbol{\eta}(\boldsymbol{u}) = \boldsymbol{A}\boldsymbol{u}, \quad \text{linear predictor, linear combination of the latent variables}$$

$$y_i|\boldsymbol{u},\boldsymbol{\theta} \sim p(y_i|\eta_i(\boldsymbol{u}),\boldsymbol{\theta}), \quad \text{response variables } y_i\text{, conditionally independent}$$

Extended LGM of inlabru type

$$\boldsymbol{\theta} \sim p(\boldsymbol{\theta}), \quad \text{(hyper-)parameters}$$

$$\boldsymbol{u}|\boldsymbol{\theta} \sim \mathsf{N}\left(\boldsymbol{\mu}_u, \boldsymbol{Q}(\boldsymbol{\theta})^{-1}\right), \quad \text{complex structured latent Gaussian field}$$

$$\boldsymbol{\eta}(\boldsymbol{u}) = h(\boldsymbol{u}), \quad \text{non-linear predictor, general function of the latent variables}$$

$$y_i|\boldsymbol{u},\boldsymbol{\theta} \sim p(y_i|\eta_i(\boldsymbol{u}),\boldsymbol{\theta}), \quad \text{response variables } y_i\text{, conditionally independent}$$

Approximate INLA for non-linear predictors

Linearised predictor

Let $\widetilde{\boldsymbol{\eta}}(\boldsymbol{u})$ be the non-linear predictor, and let $\overline{\boldsymbol{\eta}}(\boldsymbol{u})$ be the 1st order Taylor approximation at some $\boldsymbol{u}_0$,

$$\overline{\boldsymbol{\eta}}(\boldsymbol{u}) = \widetilde{\boldsymbol{\eta}}(\boldsymbol{u}_0) + \boldsymbol{B}(\boldsymbol{u} - \boldsymbol{u}_0) = [\widetilde{\boldsymbol{\eta}}(\boldsymbol{u}_0) - \boldsymbol{B}\boldsymbol{u}_0] + \boldsymbol{B}\boldsymbol{u},$$

where $\boldsymbol{B}$ is the derivative matrix for the non-linear predictor, evaluated at $\boldsymbol{u}_0$.

The non-linear observation model $\widetilde{p}(\boldsymbol{y}|\boldsymbol{u},\boldsymbol{\theta})$ is approximated by

$$\overline{p}(\boldsymbol{y}|\boldsymbol{u},\boldsymbol{\theta}) = p(\boldsymbol{y}|\overline{\boldsymbol{\eta}}(\boldsymbol{u}),\boldsymbol{\theta}) \approx p(\boldsymbol{y}|\widetilde{\boldsymbol{\eta}}(\boldsymbol{u}),\boldsymbol{\theta}) = \widetilde{p}(\boldsymbol{y}|\boldsymbol{u},\boldsymbol{\theta})$$

The non-linear model posterior is factorised as

$$\widetilde{p}(\boldsymbol{\theta},\boldsymbol{u}|\boldsymbol{y}) = \widetilde{p}(\boldsymbol{\theta}|\boldsymbol{y})\widetilde{p}(\boldsymbol{u}|\boldsymbol{y},\boldsymbol{\theta}),$$

and the linear model approximation is factorised as

$$\overline{p}(\boldsymbol{\theta},\boldsymbol{u}|\boldsymbol{y}) = \overline{p}(\boldsymbol{\theta}|\boldsymbol{y})\overline{p}(\boldsymbol{u}|\boldsymbol{y},\boldsymbol{\theta}).$$

Iterated INLA in inlabru

The observation model is linked to $\boldsymbol{u}$ only through the non-linear predictor $\widetilde{\boldsymbol{\eta}}(\boldsymbol{u})$.

Iterative INLA algorithm:

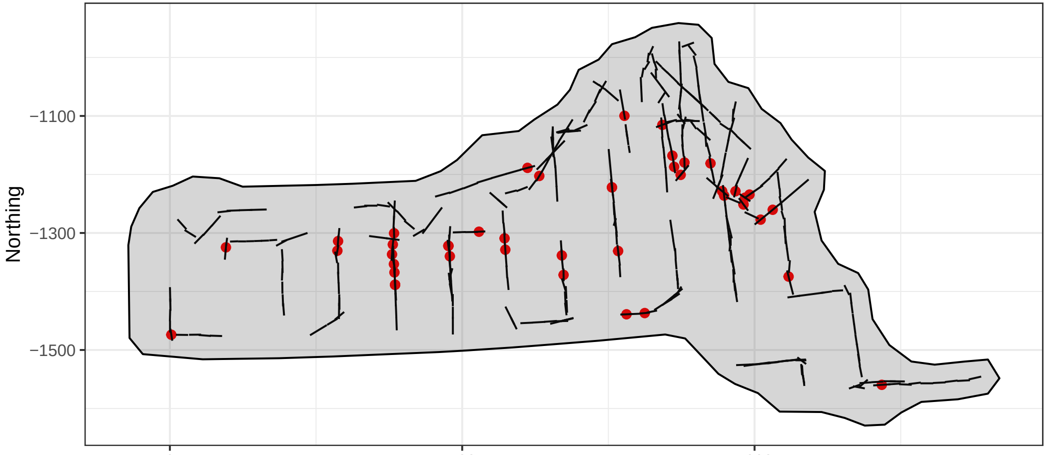
1. Let $\boldsymbol{u}_0$ be an initial linearisation point for the latent variables.
2. Compute the predictor linearisation at $\boldsymbol{u}_0$
3. Compute the linearised INLA posterior $\overline{p}(\boldsymbol{\theta}|\boldsymbol{y})$ and let $\widehat{\boldsymbol{\theta}} = \operatorname{argmax}_{\boldsymbol{\theta}} \overline{p}(\boldsymbol{\theta}|\boldsymbol{y})$
4. Let $\boldsymbol{u}_1 = \operatorname{argmax}_{\boldsymbol{u}} \overline{p}(\boldsymbol{u}|\boldsymbol{y}, \widehat{\boldsymbol{\theta}})$ be the initial candidate for new linearisation point.
5. Let $\boldsymbol{u}_\alpha = (1-\alpha)\boldsymbol{u}_0 + \alpha\boldsymbol{u}_1$, and find the value α that minimises $\|\widetilde{\eta}(\boldsymbol{u}_\alpha) - \overline{\eta}(\boldsymbol{u}_1)\|$.
6. Set the linearisation point $\boldsymbol{u}_0$ to $\boldsymbol{u}_\alpha$ and repeat from step 2, unless the iteration has converged to a given tolerance.
7. Compute $\overline{p}(\boldsymbol{u}|\boldsymbol{y})$

In step 4, only the *conditional* posterior mode for $\boldsymbol{u}$ is needed, so the costly nested integration step of the R-INLA algorithm only needs to be run in a final iteration of the algorithm, in step 7.

Step 5 can use an approximate line search method.

Example: Thinned Poisson point processes

We want to model the presence of groups of dolphins using a Log-Gaussian Cox Process (LGCP)
However, when surveying dolphins from a ship travelling along lines (*transects*), the probability of detecting a group of animals depends their distance distance from the ship.



Example: Thinned Poisson point processes

We want to model the presence of groups of dolphins using a Log-Gaussian Cox Process (LGCP)
However, when surveying dolphins from a ship travelling along lines (*transects*), the probability of detecting a group of animals depends their distance distance from the ship, e.g. via

$$\mathsf{P}(\text{detection}) = 1 - \exp\left(-\frac{\sigma}{\text{distance}}\right) \quad \text{(hazard rate model)}$$

This results in a *thinned* Poisson process model on $(\text{space}, \text{distance})$ along the transects:

$$\log(\lambda(\boldsymbol{s}, \text{distance})) = \text{Intercept} + \text{field}(\boldsymbol{s}) + \log\left[\mathsf{P}(\text{detection at } \boldsymbol{s} \mid \text{distance}, \sigma)\right] + \log(2)$$

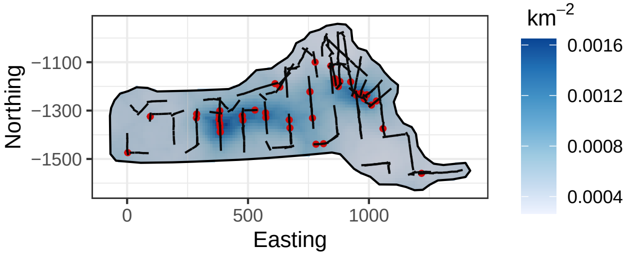
inlabru knows how to construct the Poisson process likelihood along lines and on polygons, and kronecker spaces (line $\times$ distance)
We can define $\log(\sigma)$ as a latent Gaussian variable and iteratively linearise. The non-linearity is mild, and the iterative INLA method converges.

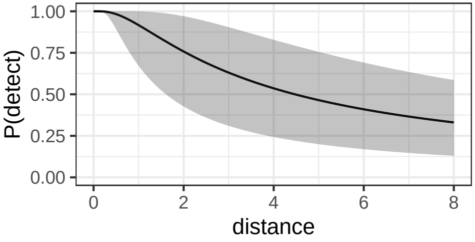
```r
log_det_prob <- function(distance, log_sig) {
  log1p(-exp(-exp(log_sig) / distance))
}

comp <- ~ field(coordinates, model = matern) + log_sig(1) + Intercept(1)
form <- coordinates + distance ~
  Intercept + field + log_det_prob(distance, log_sig) + log(2)

fit <- bru(
  components = comp,
  like(
    family = "cp", formula = form,
    data = mexdolphin$points, # sp::SpatialPointsDataFrame
    samplers = mexdolphin$samplers, # sp::SpatialLinesDataFrame
    domain = list(
      coordinates = mexdolphin$mesh,
      distance = INLA::inla.mesh.1d(seq(0, 8, length.out = 30))
    )
  )
)
```

Posterior prediction method

```
pred_points <- pixels(mexdolphin$mesh, nx = 200, ny = 100, mask = mexdolphin$ppoly)
pred <- predict(fit, pred_points, ~ exp(field + Intercept))

det_prob <- function(distance, log_sig) { 1 - exp(-exp(log_sig) / distance) }
pred_dist <- data.frame(distance = seq(0, 8, length = 100))
det_prob <- predict(fit, pred_dist, ~ det_prob(distance, log_sig), include = "log_sig")
```

```
ggplot() + gg(pred) + gg(mexdolphin$samplers) + gg(mexdolphin$ppoly) + ...
```





Data level prediction

47 groups were seen. How many would be seen along the transects under perfect detection?

```
predpts_transect <- ipoints(mexdolphin$samplers, mexdolphin$mesh)
Lambda_transect <- predict(fit, predpts_transect,
                           ~ 16 * sum(weight * exp(field + Intercept)))
```

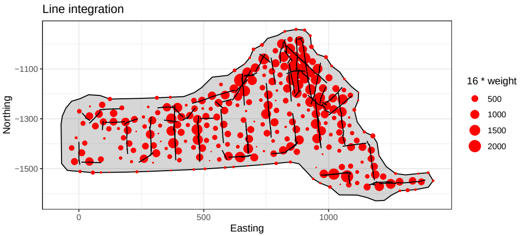
mean	sd	q0.025	q0.5	q0.975	median	mean.mc_std_err	sd.mc_std_err
85.93922	24.93825	42.0879	82.61408	149.988	82.61408	2.493825	2.239346

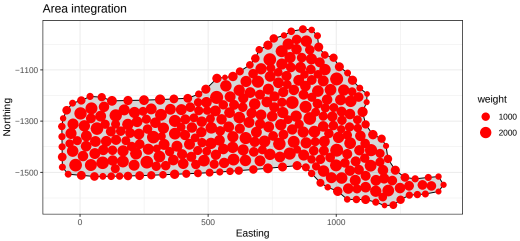
How many would be seen under perfect detection across the whole study area?

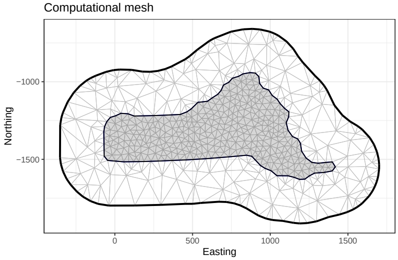
```
predpts <- ipoints(mexdolphin$ppoly, mexdolphin$mesh)
Lambda <- predict(fit, predpts, ~ sum(weight * exp(field + Intercept)))
```

mean	sd	q0.025	q0.5	q0.975	median	mean.mc_std_err	sd.mc_std_err
319.752	81.54696	190.0406	313.1301	520.4723	313.1301	8.154696	8.796282

Integration points





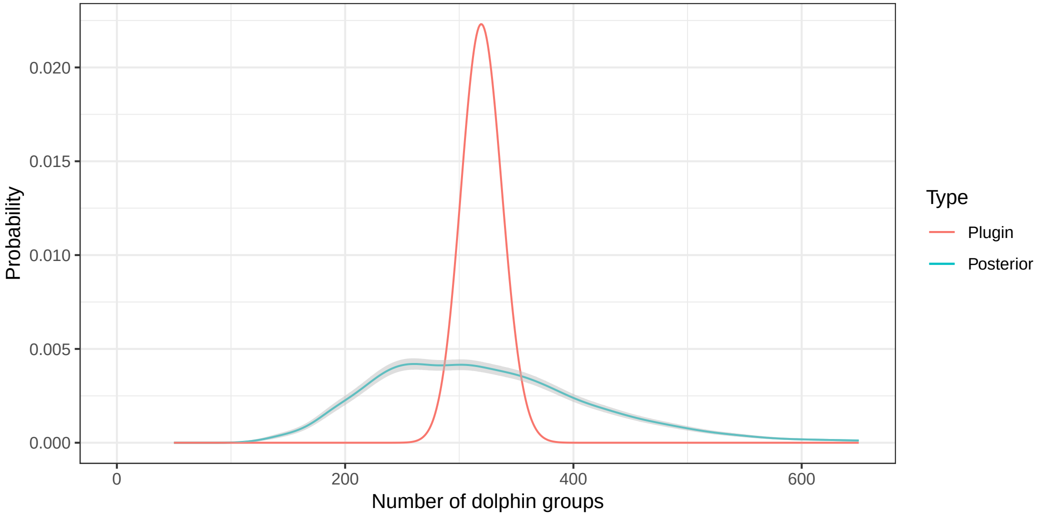


Complex prediction expressions

What's the predictive distribution of group counts?

```
Ns <- seq(50, 650, by = 1)
Nest <- predict(
  fit,
  predpts,
  ~ data.frame(
    N = Ns,
    density = dpois(Ns, lambda = sum(weight * exp(field + Intercept)))
  ),
  n.samples = 2500
)

Nest$plugin_estimate <- dpois(Nest$N, lambda = Lambda$mean)
```

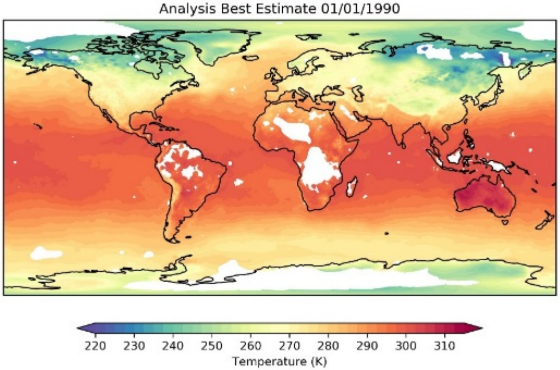
Full posterior prediction uncertainty vs plugin prediction



EUSTACE ANALYSIS

Combines in-situ and satellite data sources to derive daily air temperatures across the globe with quantified uncertainties.

- Daily mean air temperature (2 m) estimates from the mid-late 19th century at ¼ degree resolution.
- Observational dataset for use in climate monitoring, services and research.
  - Quantify bias and uncertainty arising from observational sampling (in space and time);
  - Quantify uncertainty from instrumental effects/network changes.
- Higher resolution daily gridded analyses for regional climate
  - Combine in situ and remote sensing data to support high resolution analysis.
  - Absolute temperature rather than anomaly product.

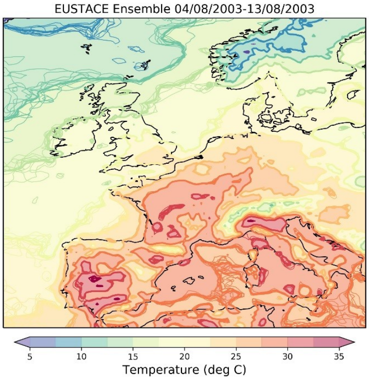


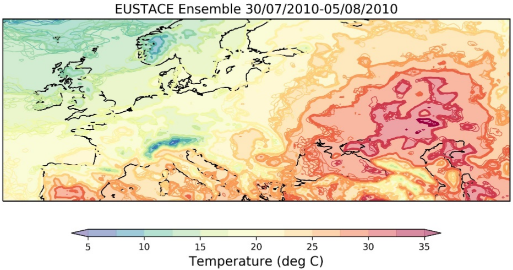
Met Office
Hadley Centre

EUSTACE

ENSEMBLE ANALYSIS

- Samples drawn from joint posterior distribution of temperature and bias variables.
- Temperature model samples projected onto analysis grid.
- Spatial/temporal correlation in analysis errors is encoded into the ensemble.
- Summary statistics can be derived from the ensemble. Expected value, total uncertainty and observation constraint information also available.

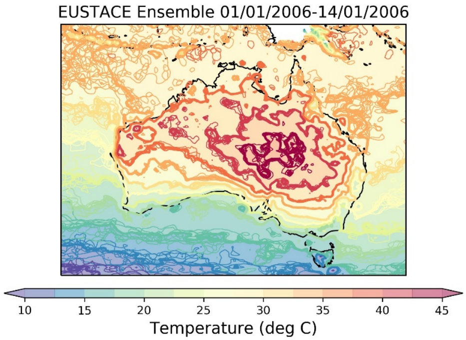


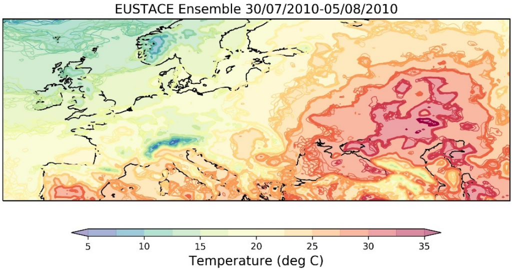


Met Office
Hadley Centre

ENSEMBLE ANALYSIS

- Samples drawn from joint posterior distribution of temperature and bias variables.
- Temperature model samples projected onto analysis grid.
- Spatial/temporal correlation in analysis errors is encoded into the ensemble.
- Summary statistics can be derived from the ensemble. Expected value, total uncertainty and observation constraint information also available.





Met Office
Hadley Centre

MULTI-SCALE ANALYSIS MODEL

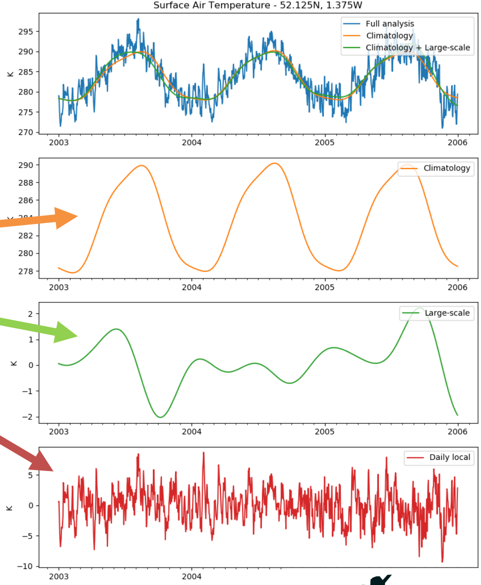
Statistical model for temperature variations and different scales (space and time):

- **Climatological variation**: local seasonal cycle with effects of latitude, altitude and coastal influence.
- **Large-scale variation**: Slowly varying climatological mean temperature field.
- **Daily Local**: daily variability associated with weather.

Simultaneously estimates observational biases of known bias structures:

- e.g. satellite biases, station homogenisation.

Central England Temperature Decomposition



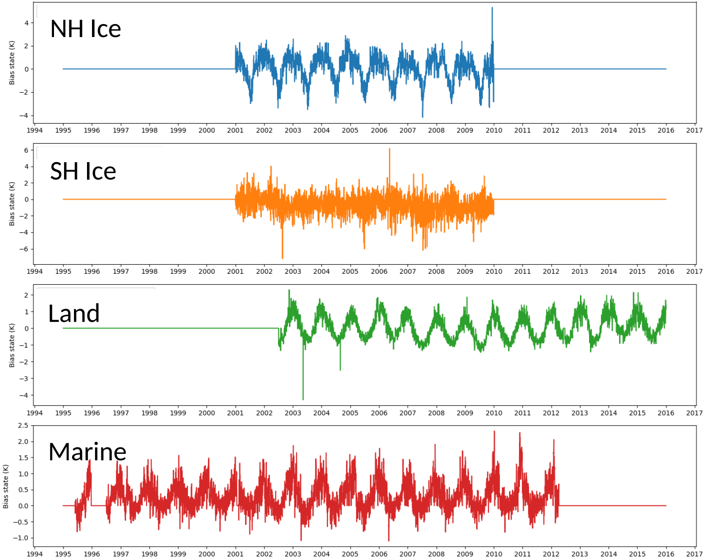
Met Office Hadley Centre

EUSTACE

SATELLITE BIAS MODELS

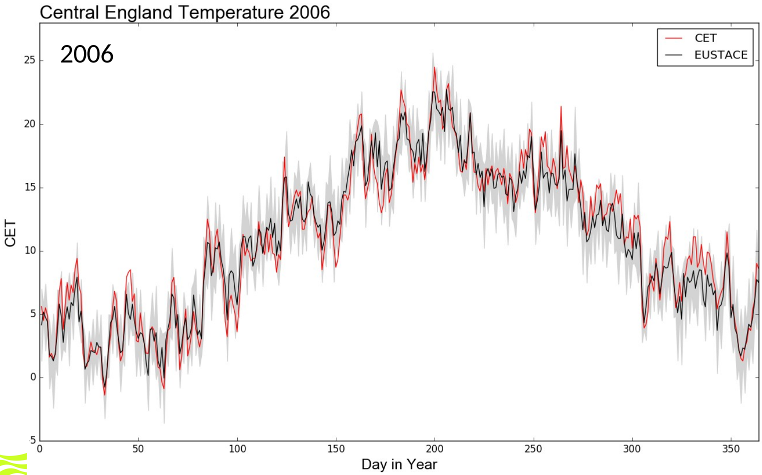
- Simplified model of known error structures in satellite air temperature retrievals:
  - Global/hemispheric systematic bias covariates.
  - Daily estimates of spatially varying bias as a spatial random field.

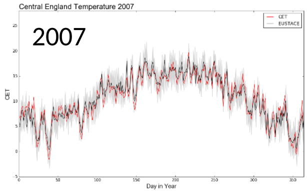
- Estimated jointly with daily temperature variability.

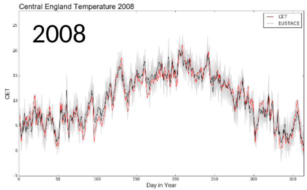


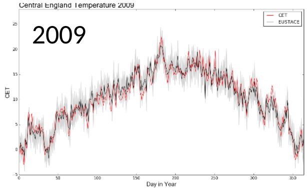
Met Office
Hadley Centre

COMPARING EUSTACE WITH CENTRAL ENGLAND TEMPERATURE









Met Office
Hadley Centre

EUSTACE

MULTI-SCALE ANALYSIS MODEL

Statistical model for temperature variations and different scales (space and time):

- Climatological variation: local seasonal cycle with effects of latitude, altitude and coastal influence.
- Large-scale variation: Slowly varying climatological mean temperature field. Station homogenisation.
- Daily Local: daily variability associated with weather. Satellite retrieval biases.

Simultaneously estimates observational biases of known bias structures:

- e.g. satellite biases, station homogenisation.

Processed on STFC's LOTUS cluster www.jasmin.ac.uk:

- Largest solves processed on 20 core/256GB RAM node.
- Highly parallel observation pre-processing.

Element	Resolution	N Variables
Seasonal	Bimonthly x 1° SPDE	245,772
Slow-scale*	5 year x 5° SPDE	107,604
Latitude	0.5° latitude SPDE	721
Altitude	(0.25° grid)	1
Coastal	(0.25° grid)	1
Grand mean	Analysis mean	1

Element	Resolution	N Variables
Large-scale	3 monthly x 5° SPDE	1,752,408
Station bias	NA	82,072

Element	Resolution	N Variables per day
Daily local	~0.5 degree SPDE	162,842
Satellite bias (marine)	Global	1
Satellite bias (land)	Global + 2.5 degree SPDE	1 + 40,962
Satellite bias (ice)	Hemispheric + 2.5 degree SPDE*	2 + 40,962

Met Office
Hadley Centre

EUSTACE

Extensions and projects in progress

- (w Liam Llamazares Elias) Penalised complexity priors for non-stationary models
- Simplified support for aggregated data models, where the predictor expression may involve integration across space (with Man Ho Suen, Andy Seaton)
- Related work (with Christopher Merchant and Xue Wang):
  Multi-band satellite data with nadir and oblique views, with non-rectangular "pixels".

$$\mathsf{E}(\text{measured}(\text{pixel}, \text{band})) = \left(\frac{1}{|D_{\text{pixel}}|} \int_{D_{\text{pixel}}} \text{conversion}[\text{SST}(\boldsymbol{s}), \text{TCWV}(\boldsymbol{s}), \text{band}]^b \, \mathrm{d}\boldsymbol{s} \right)^{1/b}$$

- Both SST and TCWV are unknown spatial fields and b is an unknown parameter
- The "conversion" function is a deterministic function evaluated on a grid of SST and TCWV for each frequency band
- Can be implemented with numerical integration for each pixel, and spline interpolation of the conversion function

Extensions and projects in progress

- (Victor Medina) Joint covariate&outcome models for longitudinal credit risk
- (Francesco Serafini) Hawkes processes for earthquake forecasting; self-exciting Poisson processes with $\lambda(\boldsymbol{s},t) = \mu(\boldsymbol{s},t,\boldsymbol{u}) + \sum_{i;t_i<t} h(\boldsymbol{s}-\boldsymbol{s}_i,,t-t_i,\boldsymbol{u})$ which is not log-linear.
- Copulas and transformation models; can handle non-Gaussian parameter priors as latent variables, e.g. $\lambda \sim \text{Exp}(\gamma)$ is equivalent to $\lambda = -\log[1-\Phi(u)]/\gamma$, where $u \sim \text{N}(0,1)$
- Extending the supported set of R-INLA models (survival models, etc)
- (w Andy Seaton) Added `sf` and `terra` support to prepare for the retirement of the `rgdal` package in 2023
- Converting the SPDE meshing code to a separate `fmesher` package
- Direct support for non-separable space-time models (`INLAspacetime`, with Elias Krainski, David Bolin, Haakon Bakka, and Haavard Rue)
- Improved support for factors and fixed effects interaction models

Further work

- How accurate are the linearised posteriors? Need diagnostic metrics for all models. Options that are more or less computable in practice include
  - $\mathsf{E}_{\boldsymbol{u}\sim\overline{p}(\boldsymbol{u}|\boldsymbol{y})}(\|\overline{\boldsymbol{\eta}} - \widetilde{\boldsymbol{\eta}}\|^2)$
  - $\sum_i \mathsf{E}_{\boldsymbol{u}\sim\overline{p}(\boldsymbol{u}|\boldsymbol{y})}(|\overline{\eta}_i - \widetilde{\eta}_i|^2)/\mathsf{Var}_{\boldsymbol{u}\sim\overline{p}(\boldsymbol{u}|\boldsymbol{y})}(\overline{\eta}_i)$
  - $\mathsf{E}_{\boldsymbol{u}\sim\overline{p}(\boldsymbol{u}|\boldsymbol{y})}\left(\log\left(\frac{\overline{p}(\boldsymbol{u}|\boldsymbol{y},\boldsymbol{\theta})}{\widetilde{p}(\boldsymbol{u}|\boldsymbol{y},\boldsymbol{\theta})}\right)\right)$
- Improved convergence diagnostics and detection of unintended incorrect user input
- Interoperability with posterior analysis and plotting packages

References

- F. Lindgren, H. Rue and J. Lindström (2011),
  An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach (with discussion), JRSSB, 73(4):423–498. Code available in `R-INLA`, see
  `http://r-inla.org/`

- Fabian E. Bachl, Finn Lindgren, David L. Borchers, and Janine B. Illian (2019)
  inlabru: an R package for Bayesian spatial modelling from ecological survey data,
  Methods in Ecology and Evolution, 10(6):760–766.
  `https://doi.org/10.1111/2041-210X.13168`

- CRAN package: `inlabru`
  `https://inlabru.org/`
  `https://inlabru-org.github`
  `https://github.com/inlabru`

- inlabru: The Scottish INLA interface