# The R-INLA package

David Bolin
Chalmers University of Technology

Búzios, RJ, Brazil,
June 18, 2014

# R-INLA: some history

- In the beginning there was

  GMRFLib

  A C library for fast computations for GMRFs.
- GMRFLib begot

  INLA

  A C library for fast approximate inference, accessed through .ini files.
- After much wailing and gnashing of teeth there came

  R-INLA

  which takes `R` code and writes an appropriate .ini file for the INLA C–program to read.

# Obtaining R-INLA

Because of (among other things) the structure with .ini files and an external C library, R-INLA is not on CRAN.

For easy installation instructions, see

`http://www.r-inla.org/download`

The easiest way is probably to open R and write

```
source("http://www.math.ntnu.no/inla/givemeINLA.R")
```

# Bayesian structured additive regression models

R-INLA supports hierarchical GMRF models of the following type:

$$y_j|\eta_j, \boldsymbol{\theta}_1 \sim \pi(y_j|\eta_j, \boldsymbol{\theta}_1), \quad j \in J$$
$$\eta_i = \alpha + \mathbf{z_i}^T\boldsymbol{\beta} + \sum_\gamma f_\gamma(c_{\gamma,i}) + \mathbf{u_i}, \quad i \in I$$
$$\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \sim \pi(\boldsymbol{\theta}), \quad \text{(priors for hyperparameters)}$$

where $J \subset I$ and

| | | |
|---|---|---|
| $\alpha$ | : | Intercept |
| $\boldsymbol{\beta}$ | : | linear effects of covariates $\mathbf{z}$ |
| $\{f_\gamma(\cdot)\}$ | : | Non-linear/smooth effects of covariates $\mathbf{c}_\gamma$ |
| $\mathbf{u}$ | : | Unstructured error terms |
| $\mathbf{x}$ | = | $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \{f_\gamma(\cdot)\}, \mathbf{u}\}$ has a Gaussian prior. |
| $\boldsymbol{\eta}$ | : | enters the likelihood through a known link function $g(\cdot)$. |

# Examples where models like this are used

- Dynamic linear models
- Stochastic volatility models
- Generalised linear (mixed) models
- Generalised additive (mixed) models
- Spline smoothing
- Semiparametric regression
- Space-varying (semiparametric) regression models
- Disease mapping
- Log-Gaussian Cox-processes
- Model-based geostatistics
- Spatio-temporal models
- Survival analysis
- +++

# The structure of an R program using INLA

There are essentially three parts to an INLA program:

1. The data organisation
2. The *formula*—notation inherited from R's native `glm` function
3. The call to the INLA program.

# The inla function

```
> result <- inla(
      formula,   #This describes your latent field
      family = "gaussian", #The likelihood distribution.
      data = dat #A list or dataframe
      #This is all that's needed for a basic call

     verbose = TRUE, # I use this a lot!
      keep = FALSE, #Keeps the output

      #Then there are some "control statements"
      #that allow you to customise some things
      control.predictor=list(A = ObservationMatrix)
        )
```

# `formula`: Specifying the latent field

$$\eta_i = \alpha + \mathbf{z_i}^T \boldsymbol{\beta} + \sum_\gamma f_\gamma(c_{\gamma,i}) + \mathbf{u_i}$$

The latent field is specified using the "standard" `R` method

`formula = y ~ 1 + covariate + f(...).`

- $y$ is the name of your data in the data frame.
- The `f` function contains the random effect specifications.
- An intercept is fitted *automatically*! Use `-1` in your formula to avoid it.
- The fixed effects (covariates) are taken as i.i.d. normal with a common prior. (This can be changed)

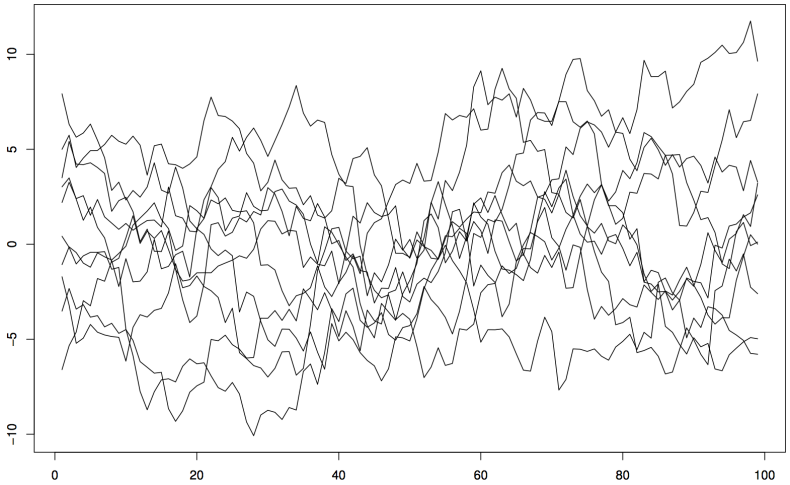
# Examples of random effects: Random walks

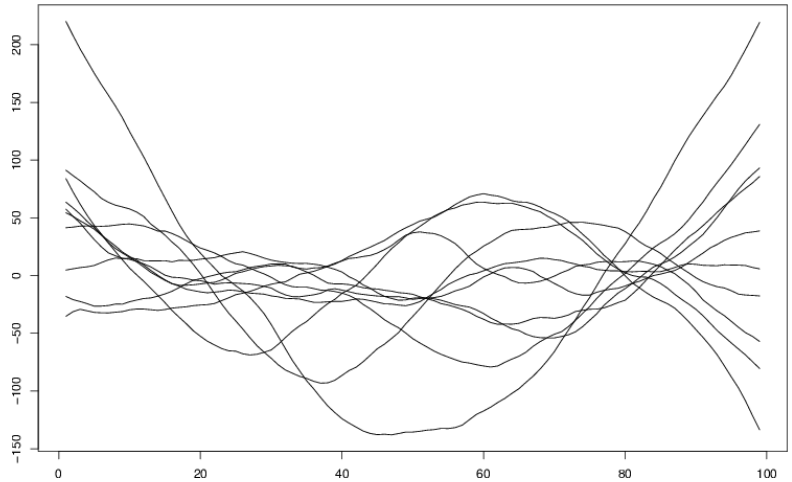
A random walk (RW) process for "smooth effects" can be used with

```
formula = Y ~  ... + f(covariate, model="rw")
```

A second-order random walk (RW2) for even "smoother" effects can be used with

```
formula = Y ~  ... + f(covariate, model="rw2")
```



# SPDE models in INLA

The SPDE models have been incorporated into the INLA package.

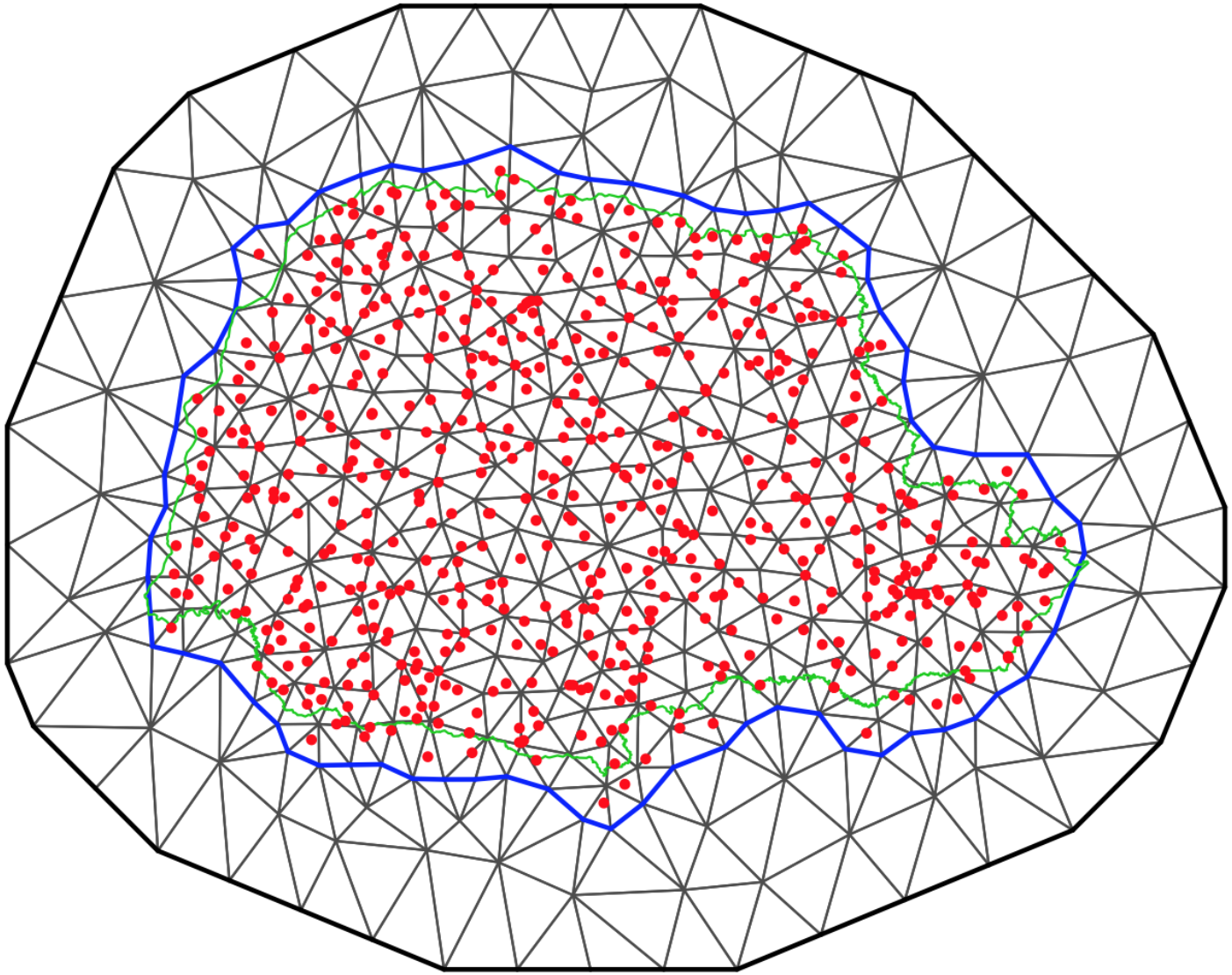
- This means that they work well with INLA!
- But they also work outside of INLA

To specify an SPDE model we need to

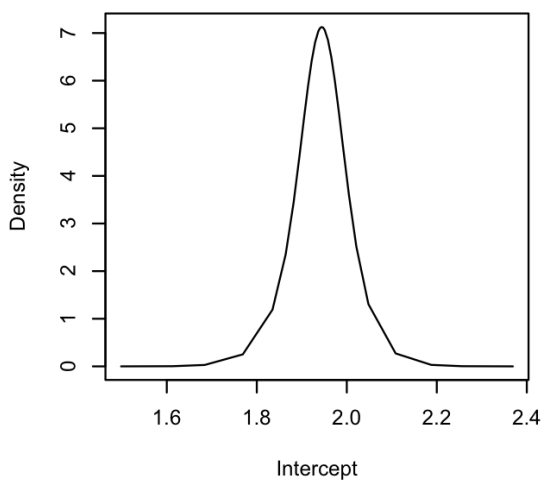
1. Create the mesh for the model.
2. Construct the observation matrix that links the measurement locations to the mesh locations.
3. Create an `spde` object.

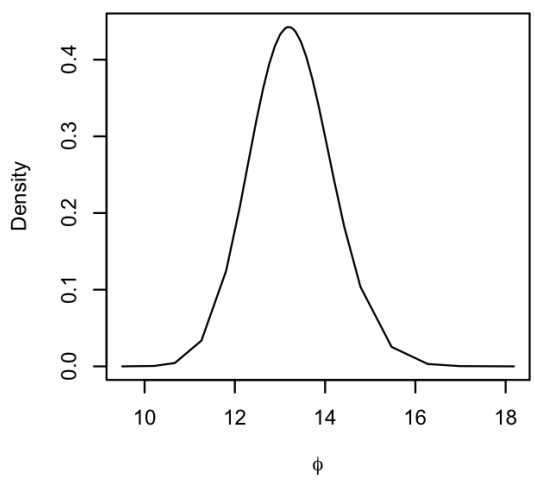
After this, the SPDE model is used like any other random effect model in the `formula`, using `f(locations, model="spde")`

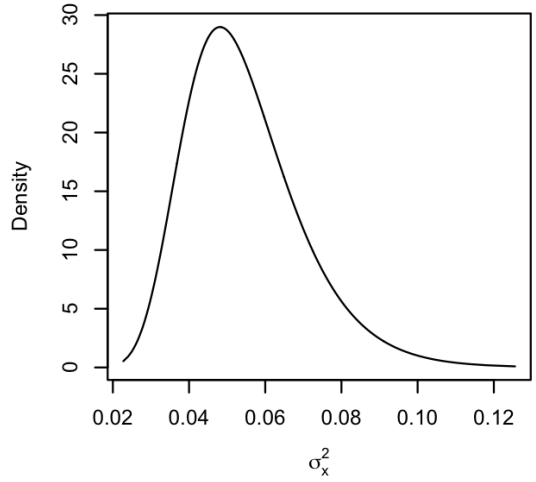
# The mesh

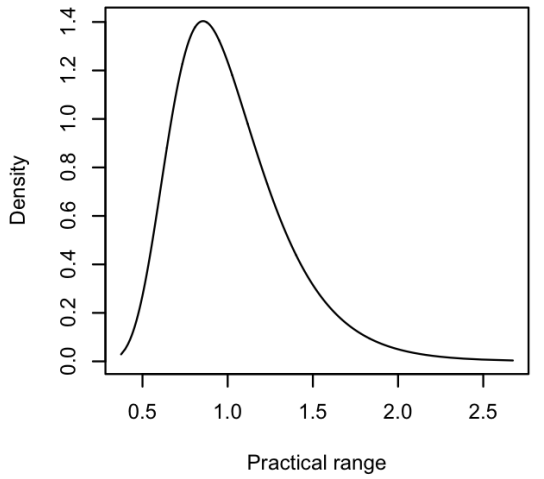


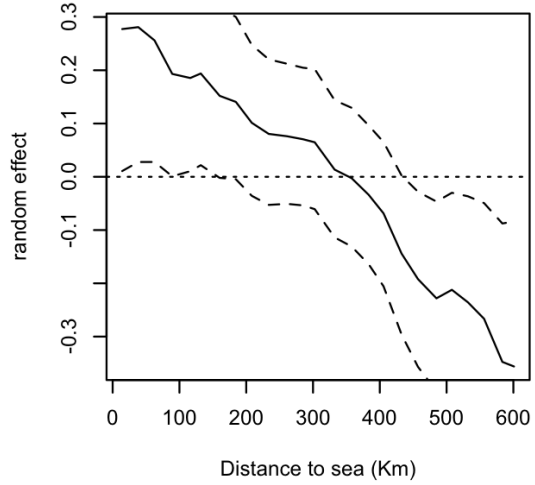
# Posteriors for hyperparameters

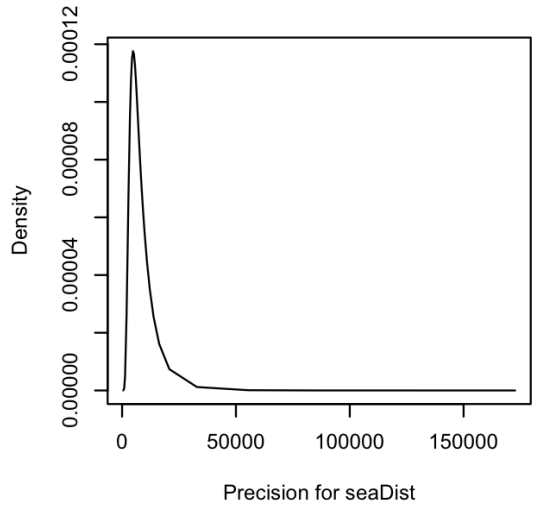


# Kriging results: Gamma (left) and Gaussian (right)