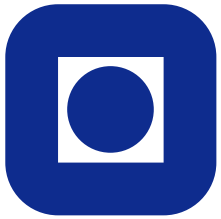


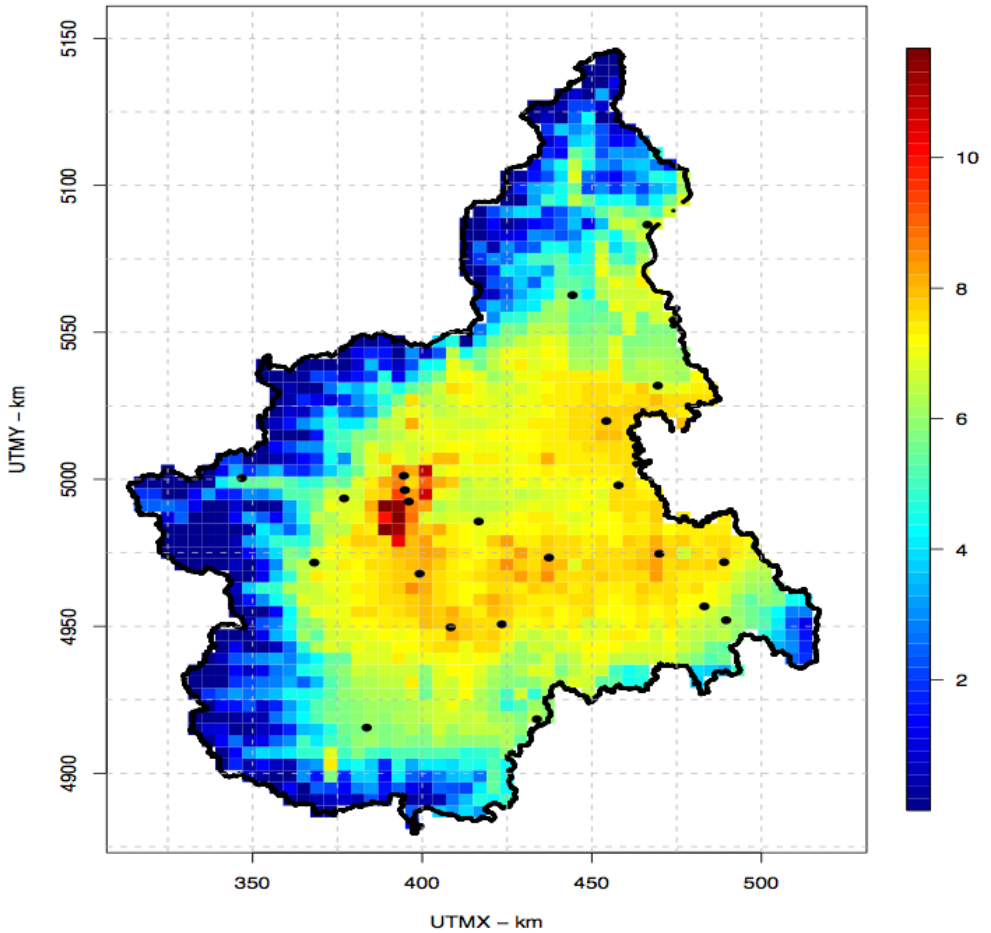
NTNU
Norwegian University of
Science and Technology

Practical spatial statistics: utilising the continuous Markov property

Daniel Simpson
Håvard Rue, Geir-Arne Fuglstad, Haakon Bakka (NTNU)
Finn Lindgren (Bath)
Janine Illian (St Andrews)
Sigrunn Sørbye (Tromsø)
Xiangping Hu (Oslo)

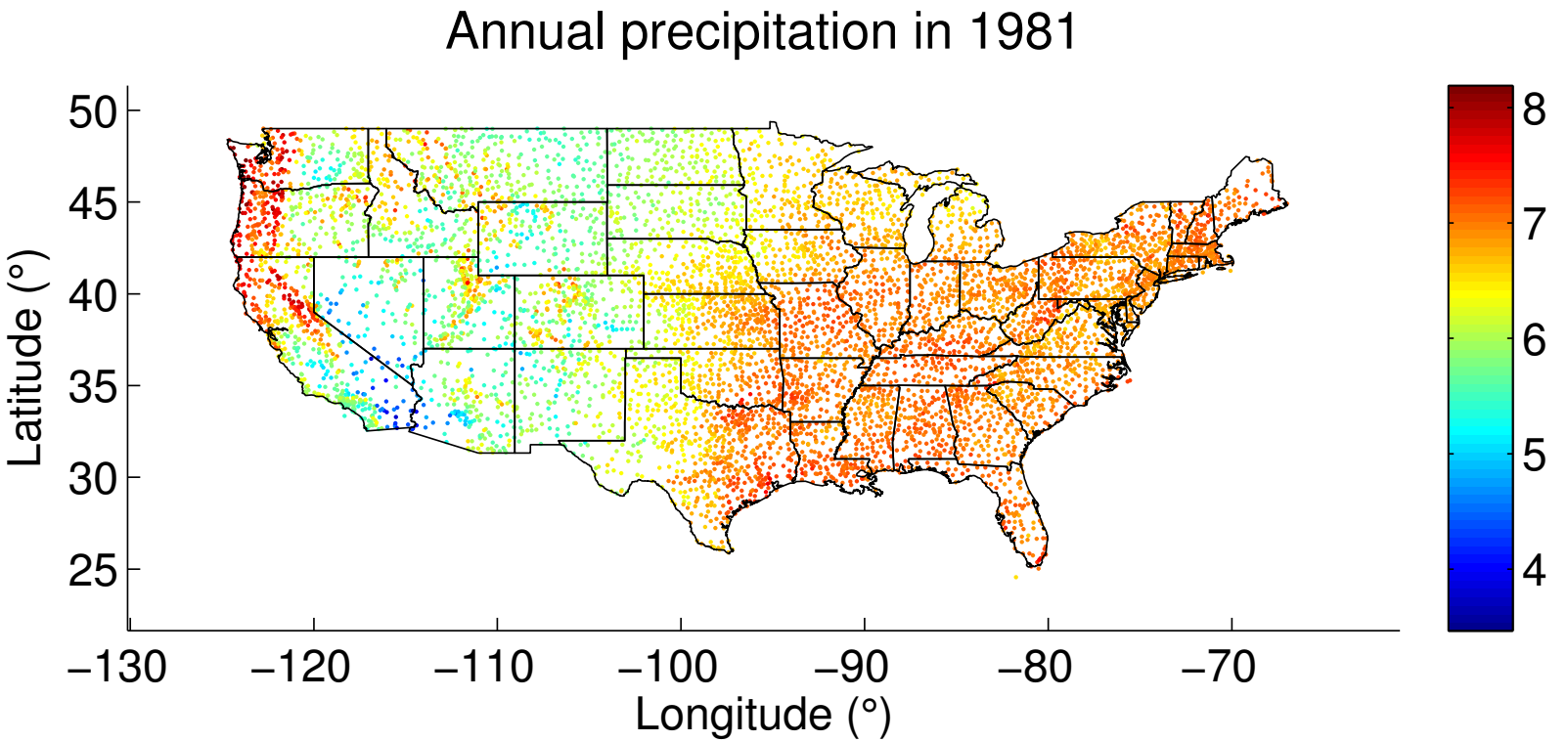
Outline

Spatial mapping

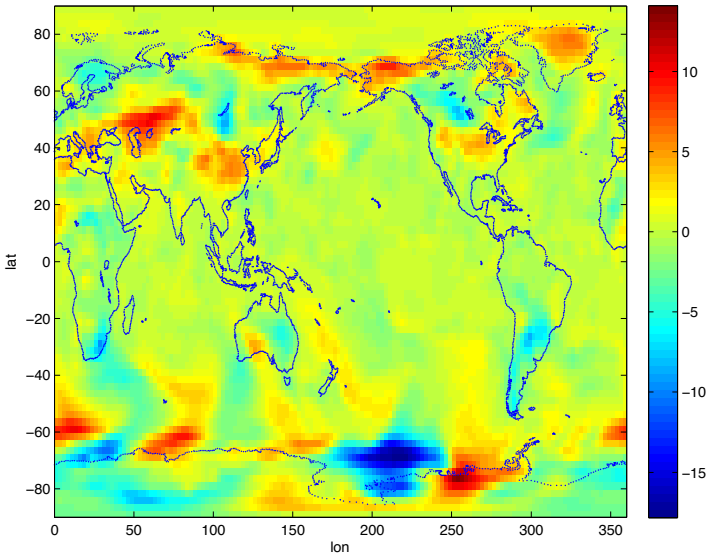


Daily PM-10 concentration in the Piemonte region, 10/05–03/06.

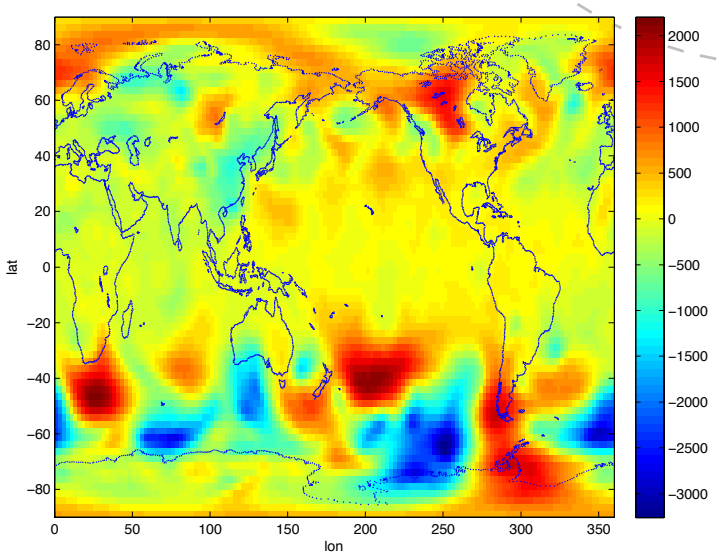
Large-scale rainfall mapping



There's power in a union

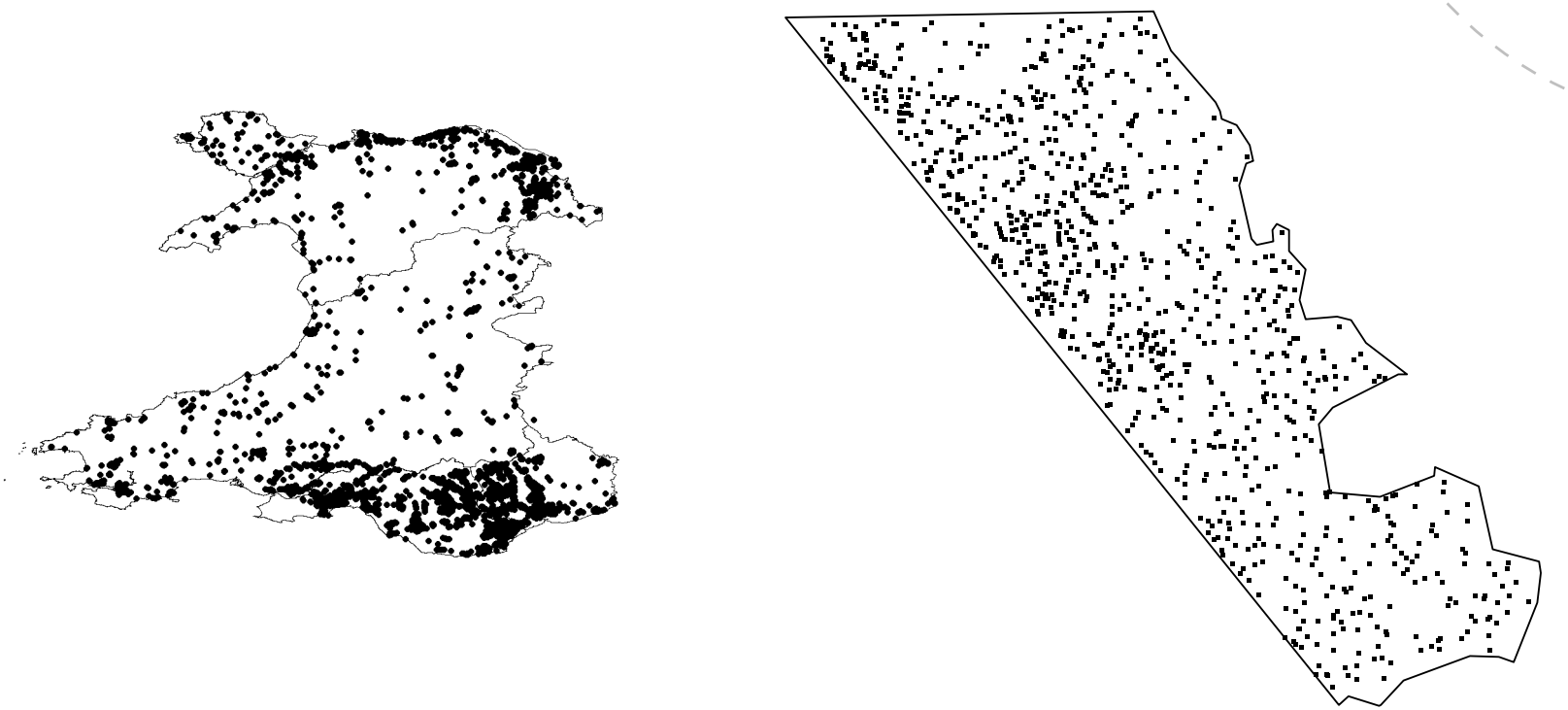


(a) Temperature



(b) Pressure

Crime and Koalas



(Left: Antisocial behaviour in Wales. Right: Koalas in Australia)

What are we looking for

What do we need?

— Good spatial models

What are we looking for

What do we need?

- Good spatial models
- Flexible spatial models

What are we looking for

What do we need?

- Good spatial models
- Flexible spatial models
- Computationally efficient spatial models

What are we looking for

What do we need?

- Good spatial models
- Flexible spatial models
- Computationally efficient spatial models
- Appropriate inference methods

The aim/dream is construct a suite of models that can be used by non-specialists as part of their statistical modelling toolbox.
(`R` interface, easy model building etc)

Outline

Gaussian random fields

Now that we know what we want, how do we do it mathematically?

— We don't ever observe a function *everywhere*.

Gaussian random fields

Now that we know what we want, how do we do it mathematically?

- We don't ever observe a function *everywhere*.
- If $\boldsymbol{x}$ is a vector of observations of $x(s)$ at different locations, we want this to be normally distributed:

$$\boldsymbol{x} = (x(s_1), \ldots, x(s_p))^T \sim N(\mathbf{0}, \boldsymbol{\Sigma})$$

Gaussian random fields

Now that we know what we want, how do we do it mathematically?

- We don't ever observe a function *everywhere*.
- If $\boldsymbol{x}$ is a vector of observations of $x(s)$ at different locations, we want this to be normally distributed:

$$\boldsymbol{x} = (x(s_1), \ldots, x(s_p))^T \sim N(\mathbf{0}, \boldsymbol{\Sigma})$$

- This is actually quite tricky: the covariance matrix $\boldsymbol{\Sigma}$ will need to depend on the set of observation sites and always has to be positive definite.

Gaussian random fields

Now that we know what we want, how do we do it mathematically?

- We don't ever observe a function *everywhere*.
- If $\boldsymbol{x}$ is a vector of observations of $x(s)$ at different locations, we want this to be normally distributed:

$$\boldsymbol{x} = (x(s_1), \ldots, x(s_p))^T \sim N(\mathbf{0}, \boldsymbol{\Sigma})$$

- This is actually quite tricky: the covariance matrix $\boldsymbol{\Sigma}$ will need to depend on the set of observation sites and always has to be positive definite.
- It turns out you can actually do this by setting $\Sigma_{ij} = c(\boldsymbol{s}_i, \boldsymbol{s}_j)$ for some *covariance function* $c(\cdot, \cdot)$.

Gaussian random fields

Now that we know what we want, how do we do it mathematically?

- We don't ever observe a function *everywhere*.
- If $\boldsymbol{x}$ is a vector of observations of $x(s)$ at different locations, we want this to be normally distributed:

$$\boldsymbol{x} = (x(s_1), \ldots, x(s_p))^T \sim N(\mathbf{0}, \boldsymbol{\Sigma})$$

- This is actually quite tricky: the covariance matrix $\boldsymbol{\Sigma}$ will need to depend on the set of observation sites and always has to be positive definite.
- It turns out you can actually do this by setting $\Sigma_{ij} = c(\boldsymbol{s}_i, \boldsymbol{s}_j)$ for some *covariance function* $c(\cdot, \cdot)$.
- Not every function will ensure that $\boldsymbol{\Sigma}$ is positive definite!

Gaussian random fields

Defn: Gaussian random fields

A random function $x(s)$ is a GRF iff there is a postitive definite function $c(s, s')$ such that, for every finite collection of points $\{s_1, \ldots, s_p\}$,

$$\boldsymbol{x} \equiv (x(s_1), \ldots, x(s_p))^T \sim N(\boldsymbol{0}, \boldsymbol{\Sigma}),$$

where $\Sigma_{ij} = c(s_i, s_j)$.

— $\boldsymbol{\Sigma}$ will almost never be sparse.
— It is typically very hard to find families of parameterised positive definite functions.
— This is hard for nonstationary, multivariate or spatiotemporal processes.

The problem with infinite dimensional models is

We only have a finite amount of information!

- "Asymptotics is a foreign country"
- We *never* overcome our prior!
- It's very expensive to compute....

Making compromises

While we write (and think) in terms of infinite dimensional models, it's becoming increasingly uncommon to compute with them.

- Does not scale to big data
- Does not scale to fine-scale prediction
- It isn't easy to specify difficult dependency structures

Outline

Approximate models

The easiest situation to cover is when the likelihood is computationally intractable.

— An overabundance of examples occur in the field of “inverse problems”:
  - This is basically ‘spatial statistics’ under a different name
  - It shares many similar problems: infinite dimensional priors vs low information, computational challenges etc
  - The difference is in the likelihood, which is typically extremely complicated and expensive to compute (minutes/hours/days/weeks).

Approximate models

The easiest situation to cover is when the likelihood is computationally intractable.

- An overabundance of examples occur in the field of “inverse problems”:
  - This is basically ‘spatial statistics’ under a different name
  - It shares many similar problems: infinite dimensional priors vs low information, computational challenges etc
  - The difference is in the likelihood, which is typically extremely complicated and expensive to compute (minutes/hours/days/weeks).
- There are also a collection of “classical” statistical problems in which this occurs.

A useful example: Log-Gaussian Cox processes

The likelihood *in the most boring case* is

$$\log(\pi(Y|x(s))) = |\Omega| - \int_\Omega \Lambda(s)\, ds + \sum_{s_i \in Y} \Lambda(s_i),$$

where Y is the set of observed locations and $\Lambda(s) = \exp(x(s))$, and $x(s)$ is a Gaussian random field.

The is very different from the Gaussian examples: it requires the field everywhere!

An approximate likelihood

NB: *The number of points in a region R is Poisson distributed with mean* $\int_R \Lambda(s)\, ds$.

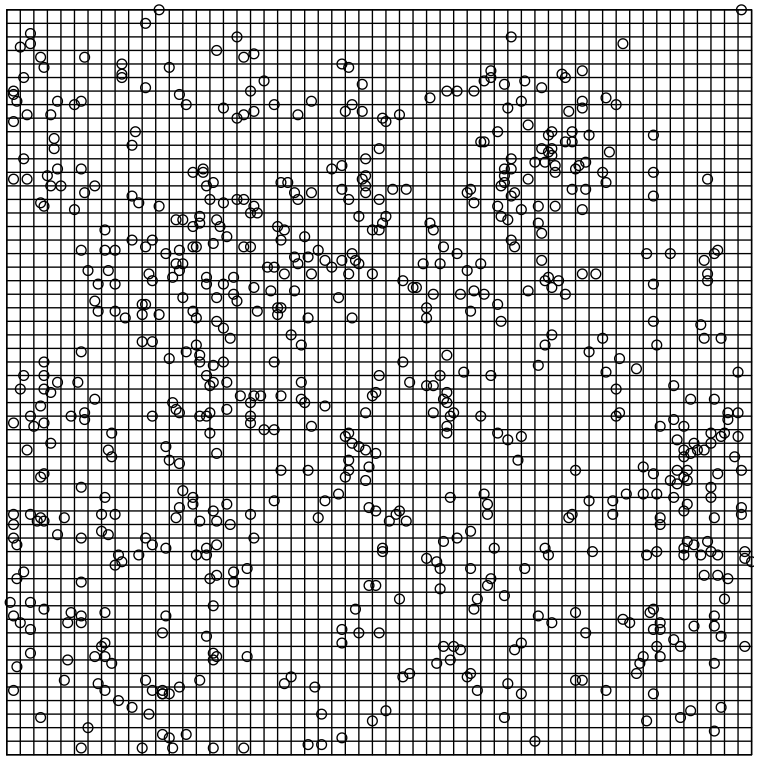
— Divide the 'observation window' into rectangles.

— Let y_i be the number of points in rectangle i.

$$y_i | x_i, \theta \sim Po(e^{x_i}),$$

— The log-risk surface is replaced with

$$\mathbf{x}|\theta \sim N(\mu(\theta), \mathbf{\Sigma}(\theta)).$$



But does this lead to valid inference?

Yes—we have perturbation bounds.

- Loosely, the error in the likelihood is transferred exactly (order of magnitude) to the Hellinger distance between the true posterior and the computed posterior.
- This is conditional on parameters.
- For the LGCP example, it follows that, for smooth enough fields $x(s)$, the error is $\mathcal{O}(n^{-1})$

The approximation turns an impossible problem into a difficult, but still useful, problem.

Approximating the Gaussian random field

In order to exert some control over the computational cost of spatial problems, it has become common to replace the infinite dimensional GRF $x(s)$ with some finite dimensional version

$$x(s) \approx \sum_{i=1}^{n} w_i \phi_i(s),$$

where $\boldsymbol{w}$ is jointly Gaussian and $\phi_i(s)$ are a set of known deterministic functions.

Some thoughts on low dimensional methods

$$x(s) \approx \sum_{i=1}^{n} w_i \phi_i(s)$$

— The number of basis functions (n) is typically chosen independently of the number of data points (N)

— Obviously not every choice of weights and basis functions will be a good one!

— It's worth getting this right: for GRFs with s square-integrable derivatives, we can construct an approximate LGCP with error like $\mathcal{O}(n^{-s})$!

Outline

Kernel representations

Most GRFs can be represented as

$$x(s) = \int_{\mathbb{R}^2} k(s,t)\, dW(t),$$

where $W(t)$ is white noise, and $k(s,t)$ is a deterministic "kernel" function.

— It is often suggested that we model $k(\cdot,\cdot)$ directly.

Kernel representations

Most GRFs can be represented as

$$x(s) = \int_{\mathbb{R}^2} k(s,t)\, dW(t),$$

where $W(t)$ is white noise, and $k(s,t)$ is a deterministic "kernel" function.

- It is often suggested that we model $k(\cdot,\cdot)$ directly.
- We can approximate the integral by a sum (Higdon, '98)

$$x(s) \approx \sum_{i=1}^{n} k(x,t_i)\xi_i,$$

where ξ_i are i.i.d. normals.

Kernel representations

Most GRFs can be represented as

$$x(s) = \int_{\mathbb{R}^2} k(s,t)\, dW(t),$$

where $W(t)$ is white noise, and $k(s,t)$ is a deterministic "kernel" function.

- It is often suggested that we model $k(\cdot,\cdot)$ directly.
- We can approximate the integral by a sum (Higdon, '98)

$$x(s) \approx \sum_{i=1}^{n} k(x,t_i)\xi_i,$$

  where ξ_i are i.i.d. normals.
- This does not work. (S, Lindgren, Rue, '10, Bolin and Lindgren '10)

Approximation properties

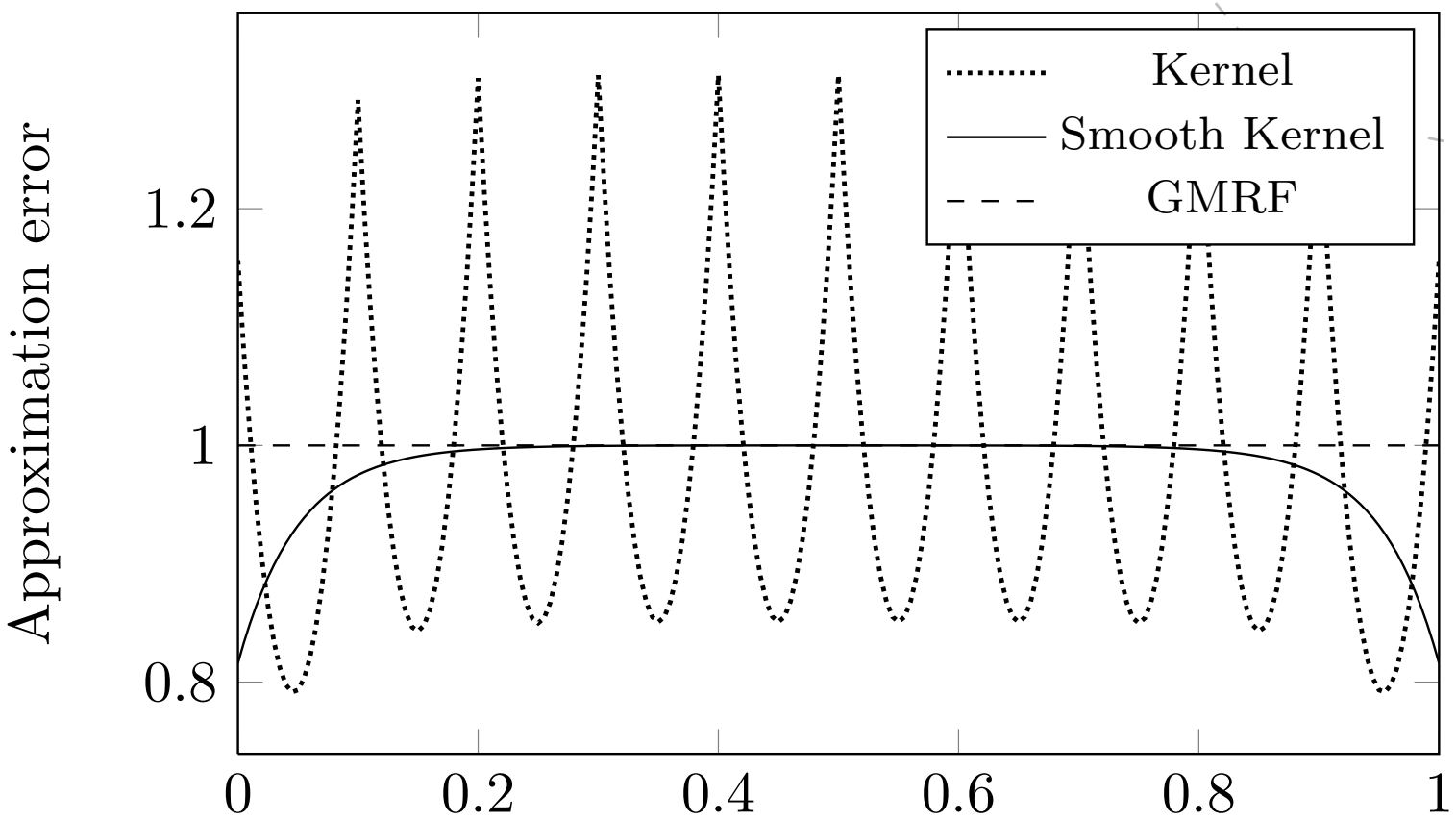
Realisations of Gaussian Random Fields are *functions*.

Appropriate question

How well can realisations of $x(\cdot)$ be approximated by functions of the form $\sum_{i=1}^{n} w_i \phi_i(s)$.

- Equivalent question: How well do functions in $\text{span}\{\phi_i\}_{i=1}^{n}$ approximate functions of a given smoothness.
- This is *not* an asymptotic question! n NEVER goes to infinity.
- Related question: How stable is the approximation procedure? (is $\|P_n f(\cdot)\| \leq C \|f\|$).
- Without considering these questions, you *cannot* know how a method will work!

Best Kernel approximation to a constant



Why did kernel methods perform badly?

— Kernel methods performed badly because there weren't enough points.

Why did kernel methods perform badly?

— Kernel methods performed badly because there weren't enough points.

— Kernel methods performed badly because the range was smaller than the grid spacing.

Why did kernel methods perform badly?

- Kernel methods performed badly because there weren't enough points.
- Kernel methods performed badly because the range was smaller than the grid spacing.
- Kernel methods performed badly because the basis functions depend on the parameter being inferred!

This is a common problem and leads to "spotty" spatial predictions and bad uncertainty estimates.

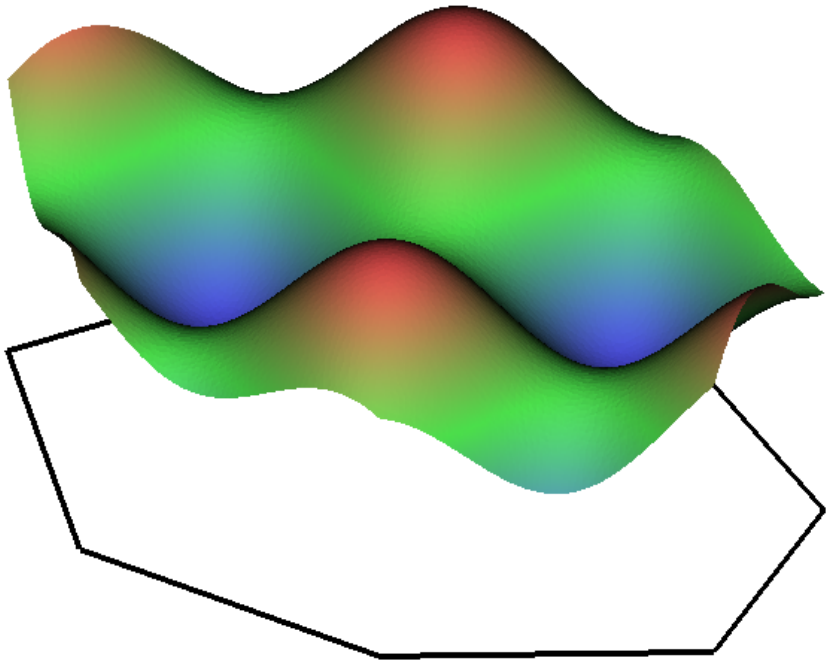
How do we do better?

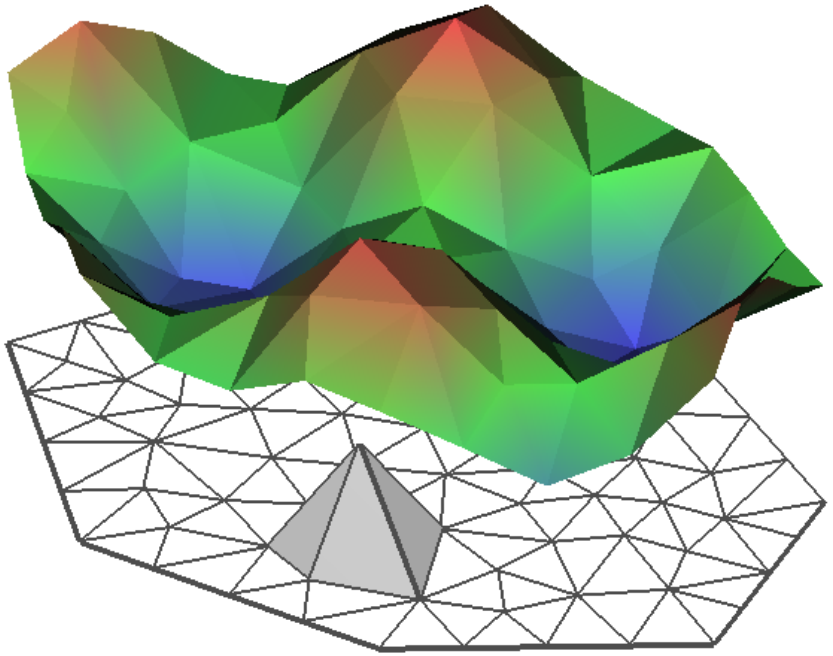
Clearly there are some problems here. We basically have two options:

— Try for another “natural approach” (see, e.g., Karhunen-Loéve expansions)

— Try to use basis functions that have good approximation properties...

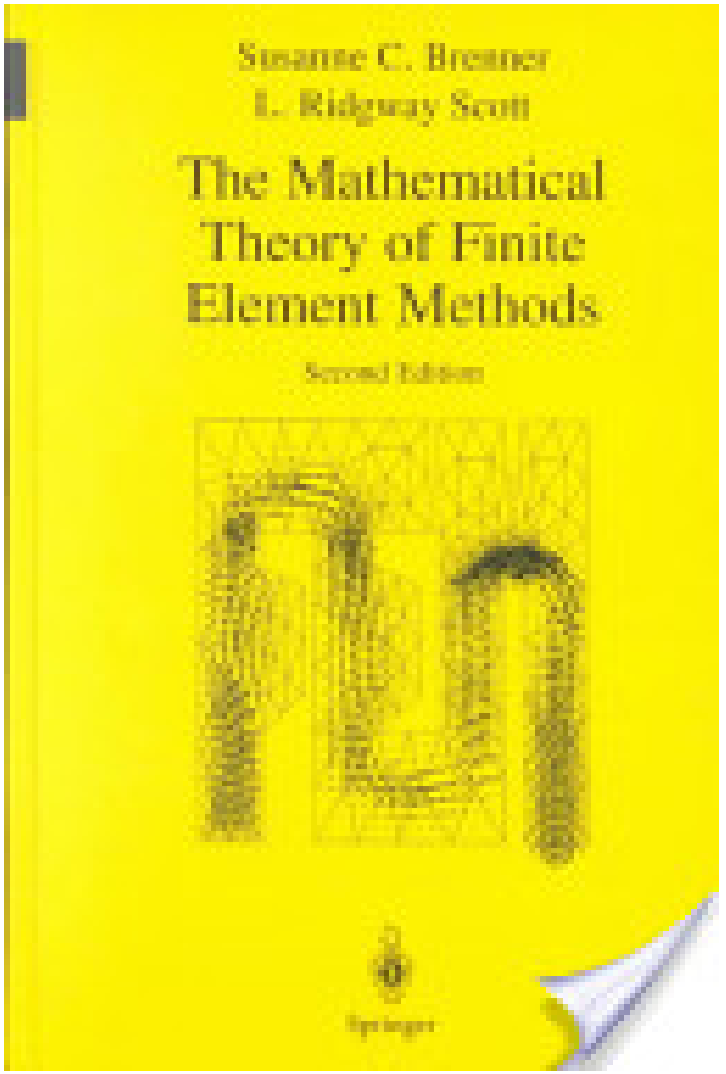
Piecewise linear approximation of surfaces





NB: The basis functions have compact support.

Known approximation properties

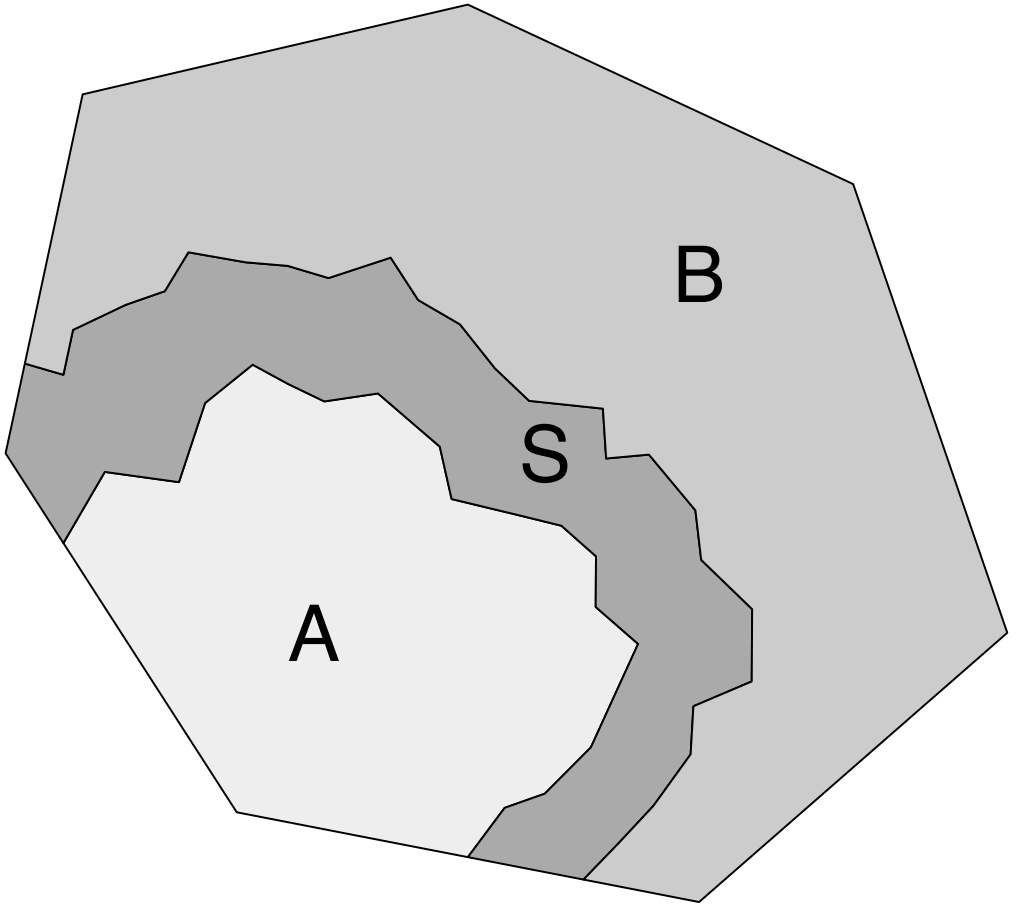


How can we use these functions?

There is no obvious way to use piecewise linear functions...

Outline

The secret is in the Markov property



How does this translate to maths?

General Result

The power spectrum of a stationary Markovian Gaussian random field has the form $R(\boldsymbol{k}) = 1/p(\boldsymbol{k})$, where $p(\boldsymbol{k})$ is a positive, symmetric polynomial.

Oh dear!

Can we salvage something from this?

Sometimes it's useful to be an engineer!

An engineering calculation

Let L be a differential operator. Then the solution to

$$Lx(s) = W(\cdot)$$

is a Gaussian random field and it has the Markov property.

- "Prove" it using Fourier transforms.
- The derivatives (local) produce the Markov property (local)
- Now we're solving (partial) differential equations: *standard!*

In the context of GMRFs

In this context, this was first noted by Whittle in the 50s (!!) who noted that Matérn fields, which have covariance function of the form

$$c(x, y) \propto (\kappa \|x - y\|)^{\nu} K_{\nu} (\kappa \|x - y\|),$$

are the stationary solutions to the SPDE

$$(\kappa^2 - \Delta)^{\frac{\nu + d/2}{2}} x(s) = W(s),$$

where

- $\Delta = \sum_{i=1}^{d} \frac{\partial^2}{\partial s_i^2}$ is the Laplacian
- $W(s)$ is spatial white noise.
- The parameter ν controls the smoothness.
- The parameter κ controls the range.

Approximating the SPDE ($\nu + d/2 = 2$)

We are looking for the piecewise linear random field

$$x_n(s) = \sum_{i=1}^{n} w_i \phi_i(s)$$

for piecewise linear $\phi_i(s)$ that *best* approximates the solution to

$$(\kappa^2 - \Delta)x(s) = W(s).$$

What comes out?

We get two matrices that are straightforward to compute:

- $\mathbf{C}_{ii} = \int_\Omega \phi_i(s)\, ds$ (the constant terms)
- $\mathbf{K}_{ij} = \int_\Omega \nabla\phi_i(s) \cdot \nabla\phi_j(s)\, ds$ (the Laplacian term)

What comes out?

We get two matrices that are straightforward to compute:

- $\mathbf{C}_{ii} = \int_\Omega \phi_i(s)\, ds$ (the constant terms)
- $\mathbf{K}_{ij} = \int_\Omega \nabla\phi_i(s) \cdot \nabla\phi_j(s)\, ds$ (the Laplacian term)

The (scary) SPDE becomes the (normal) equation

$$(\kappa^2\mathbf{C} + \mathbf{K})\mathbf{w} \sim \mathcal{N}\left(0, \mathbf{C}^{-1}\right)$$

and therefore $\mathbf{w}$ is a GMRF with precision matrix

$$\mathbf{Q} = \left(\kappa^2\mathbf{C} + \mathbf{K}\right)^T \mathbf{C}^{-1} \left(\kappa^2\mathbf{C} + \mathbf{K}\right).$$

Outline

The challenge of inference

Even after all of our approximations, we have some problems:

— The posterior random field is very high dimensional with a complicated correlation structure

- This means single-site Gibbs samplers won't work
- Markov Chain Monte Carlo (MCMC) is delicate (ask Óli Páll!)
- Numerical optimisers will also require some care!

The challenge of inference

Even after all of our approximations, we have some problems:

- The posterior random field is very high dimensional with a complicated correlation structure
  - This means single-site Gibbs samplers won't work
  - Markov Chain Monte Carlo (MCMC) is delicate (ask Óli Páll!)
  - Numerical optimisers will also require some care!
- The hyperparameters (such as the variance and range of the GRF prior) are highly correlated with the latent field
  - The simple Gibbs sampler (splitting parameters and field) will not work!
  - Reparameterisations are possible
  - The "best" choice is to try to update them jointly

Making MCMC work

Off-the-shelf MCMC schemes will not solve spatial problems efficiently.

— "Concentration of Measure" effects mean that we are trying to hit a vanishingly small target in a very high (infinte) dimensional space

Making MCMC work

Off-the-shelf MCMC schemes will not solve spatial problems efficiently.

- "Concentration of Measure" effects mean that we are trying to hit a vanishingly small target in a very high (infinte) dimensional space
- It is possible to construct random walk / MALA/ HMC Metropolis-Hastings algorithms that know where the prior is concentrated

Making MCMC work

Off-the-shelf MCMC schemes will not solve spatial problems efficiently.

- "Concentration of Measure" effects mean that we are trying to hit a vanishingly small target in a very high (infinte) dimensional space
- It is possible to construct random walk / MALA/ HMC Metropolis-Hastings algorithms that know where the prior is concentrated
- It is hard to include likelihood information!

Making MCMC work

Off-the-shelf MCMC schemes will not solve spatial problems efficiently.

- "Concentration of Measure" effects mean that we are trying to hit a vanishingly small target in a very high (infinte) dimensional space
- It is possible to construct random walk / MALA/ HMC Metropolis-Hastings algorithms that know where the prior is concentrated
- It is hard to include likelihood information!
- One solution is to *split* the posterior into a part that is controlled by the data (low-dimensional) and the part that's mostly controlled by the prior (very high dimensional).

Making MCMC work

Off-the-shelf MCMC schemes will not solve spatial problems efficiently.

- "Concentration of Measure" effects mean that we are trying to hit a vanishingly small target in a very high (infinte) dimensional space
- It is possible to construct random walk / MALA/ HMC Metropolis-Hastings algorithms that know where the prior is concentrated
- It is hard to include likelihood information!
- One solution is to *split* the posterior into a part that is controlled by the data (low-dimensional) and the part that's mostly controlled by the prior (very high dimensional).
- Preliminary results (ask Óli Páll!) are very promising!

The problem with MCMC?

It is { very / extremely / unspeakably / unbelievably / exceptionally / extraordinarily / terrifically / remarkably / impractically } slow.

A case for approximate inference

MCMC is a general method for solving generic problems

- We are *not* solving a generic problem
- We are solving a problem where most of the posterior structure is driven by the prior
- In fact, the conditional $\boldsymbol{x} \mid \boldsymbol{y}, \theta$ is almost Gaussian!
- This observation is the basis of the Integrated Nested Laplace Approximation (INLA)

Approximating the conditional

— If we do not use them, the full conditional for $\boldsymbol{x}$ looks like

$$\begin{aligned}
\pi(\boldsymbol{x} \mid \ldots) &\propto \exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \sum_i \log(\pi(y_i|x_i))\right) \\
&\approx \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T(\boldsymbol{Q}+\text{diag}(\boldsymbol{c}))(\boldsymbol{x}-\boldsymbol{\mu})\right) \\
&= \pi_G(\boldsymbol{x}|\ldots)
\end{aligned}$$

— The Gaussian approximation is constructed by matching the

- mode, and the
- curvature at the mode.

Approximating the hyperparameter posterior

We can construct an independence sampler, using $\pi_G(\cdot)$.
The Laplace-approximation for $\boldsymbol{\theta}|\boldsymbol{x}$:

$$\begin{aligned}\pi(\boldsymbol{\theta} \mid \boldsymbol{y}) &\propto \frac{\pi(\boldsymbol{\theta})\,\pi(\boldsymbol{x}|\boldsymbol{\theta})\,\pi(\boldsymbol{y}|\boldsymbol{x})}{\pi(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})} \\ &\approx \left.\frac{\pi(\boldsymbol{\theta})\,\pi(\boldsymbol{x}|\boldsymbol{\theta})\,\pi(\boldsymbol{y}|\boldsymbol{x})}{\pi_G(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})}\right|_{\boldsymbol{x}=\text{mode}(\boldsymbol{\theta})}\end{aligned}$$

Hence, we do first

- Evaluate the Laplace-approximation at some "selected" points
- Build an interpolation log-spline
- Use this parametric model as $\widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$

Putting it all together

The final step in the (simplified) INLA approximation is to note that

$$\begin{aligned}\pi(\boldsymbol{x} \mid \boldsymbol{y}) &= \int \pi(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{\theta}) \pi(\boldsymbol{\theta} | \boldsymbol{y}) \, d\boldsymbol{\theta} \\ &\approx \sum_k w_k \pi(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{\theta}_k) \widetilde{\pi}(\boldsymbol{\theta}_k | \boldsymbol{y})\end{aligned}$$

This approximation can be improved by applying further Laplace approximations to the marginals.

Limitations and notes

- This is exact (up to integration error) for Gaussian-Gaussian models.
- This is harder to program well than MCMC, but it's worth it!
- This approximation performs well in practice as long as the "effective number of replicates" is large compared to the "effective number of parameters"
- Integrating out θ is easier when it has low dimension The `R-INLA` software package contains an implementation of these (and other) ideas

Outline

Modelling rainfall in Norway (Rikke Ingebrigtsen, Finn Lindgren, Ingelin Steinsland)

If the rain in Spain falls mainly on the plain, where does it fall in Norway?

- Accurate prediction of rainfall is important for reservoir management and electricity generation.
- Norway is *not flat*.
- The variation in topography is believed to be important for the large variation in precipitation.
- There is *no way* that this field is stationary!

Covariates in the covariance (Ingebrigtsen et al)

The usual model

$$(\kappa(s) - \Delta)(\tau(s)x(s)) = W(s)$$

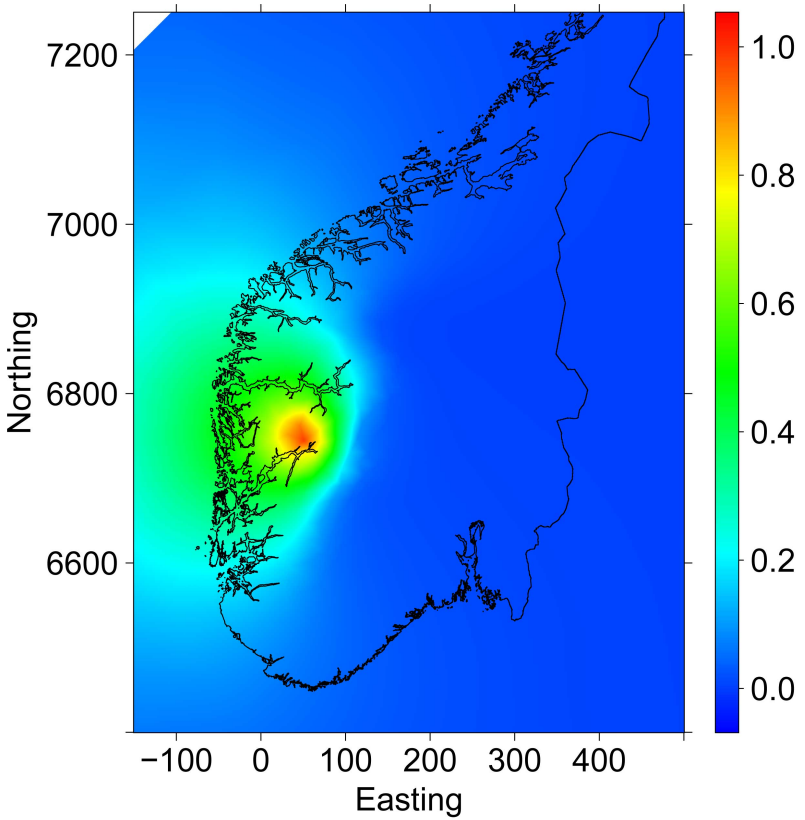
where

$$\log \tau(s) = \sum_{i=1}^{p} B_i^{\tau}(s)\theta_i, \qquad \log \kappa(s) = \sum_{i=1}^{p} B_i^{\kappa}(s)\theta_{i+p}.$$

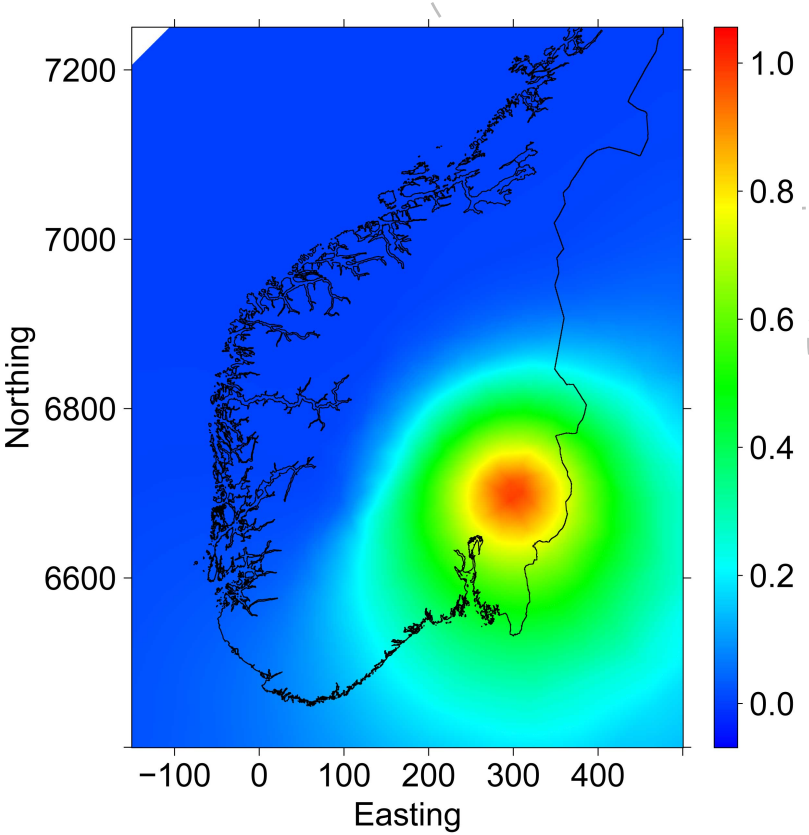
They take

$$B_1^{\tau,\kappa}(s) = 1, \quad B_1^{\tau,\kappa}(s) = \texttt{gradient}, \quad B_1^{\tau,\kappa}(s) = \texttt{elevation}.$$

What does the covariance look like?



(c) Covariance to the west



(d) Covariance to the east

"Unstructured" non-stationarity

Generally speaking, we're not going to have some sort of covariate that can explain the non-stationarity.

- Lots of methods for doing this.
- Most common is the deformation method of Samson and Guttorp: Define $x(s) = \tilde{x}(\psi(s))$ where $\tilde{x}$ is a stationary field on the deformed surface $\psi(\mathbb{R}^d)$.
- Excellent idea! But there are "barriers" to real-world application.

Idea: Rather than modelling the mapping $\psi(\cdot)$ directly, just "model" the concept of intrinsic distance.

A little bit fancy

Q: So how do you model distance?

— Go all maths-y and start talking about Riemannian metrics. (blegh)

A little bit fancy

Q: So how do you model distance?

- Go all maths-y and start talking about Riemannian metrics. (blegh)
- Be a bit physics-y and talk about diffusion.

A little bit fancy

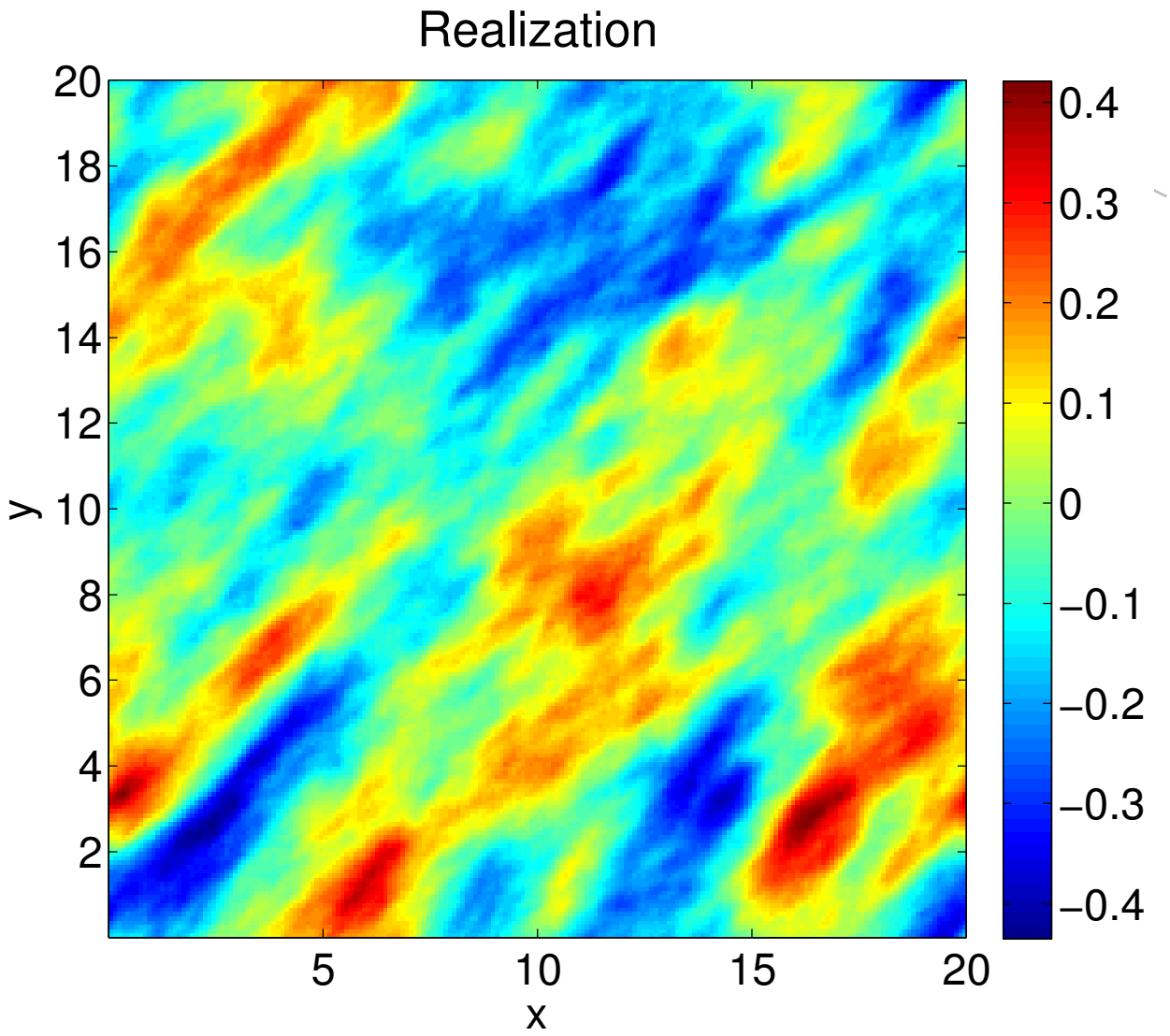
Q: So how do you model distance?

- Go all maths-y and start talking about Riemannian metrics. (blegh)
- Be a bit physics-y and talk about diffusion.

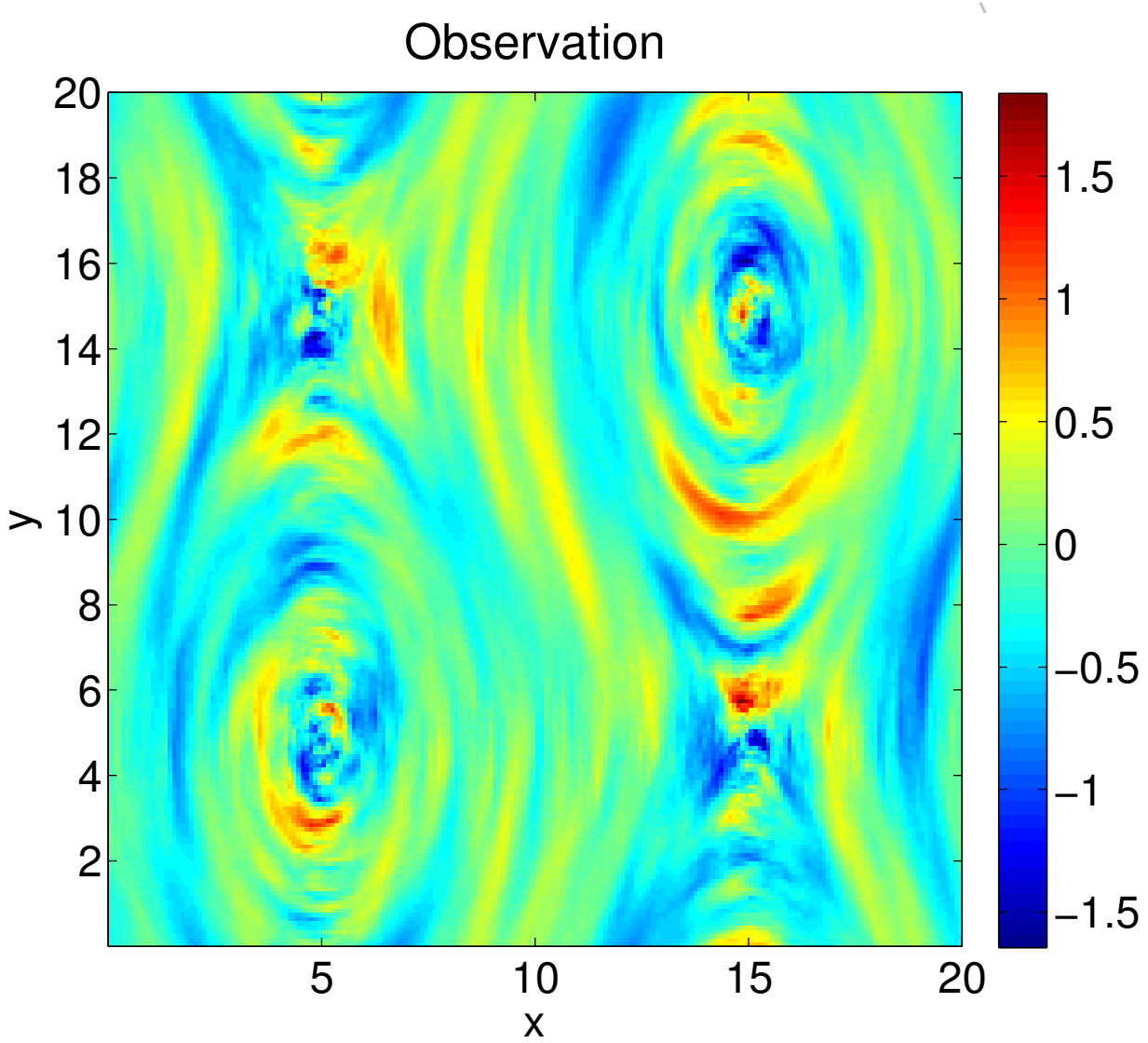
If we define the local diffusion tensor (matrix) by $\boldsymbol{H}(s)$, then we can build a model where the important directions and their relative distances are modelled by the eigenvectors and eigenvalues of $\boldsymbol{H}$.

$$\kappa^2(s)x(s) - \nabla \cdot (\boldsymbol{H}(s)\nabla x(s)) = \tau(s)W(s).$$

Constant *H*


Realization
20
18
16
14
12
10
8
6
4
2
y
5
10
15
20
x
0.4
0.3
0.2
0.1
0
−0.1
−0.2
−0.3
−0.4

Inconstant $H(s)$



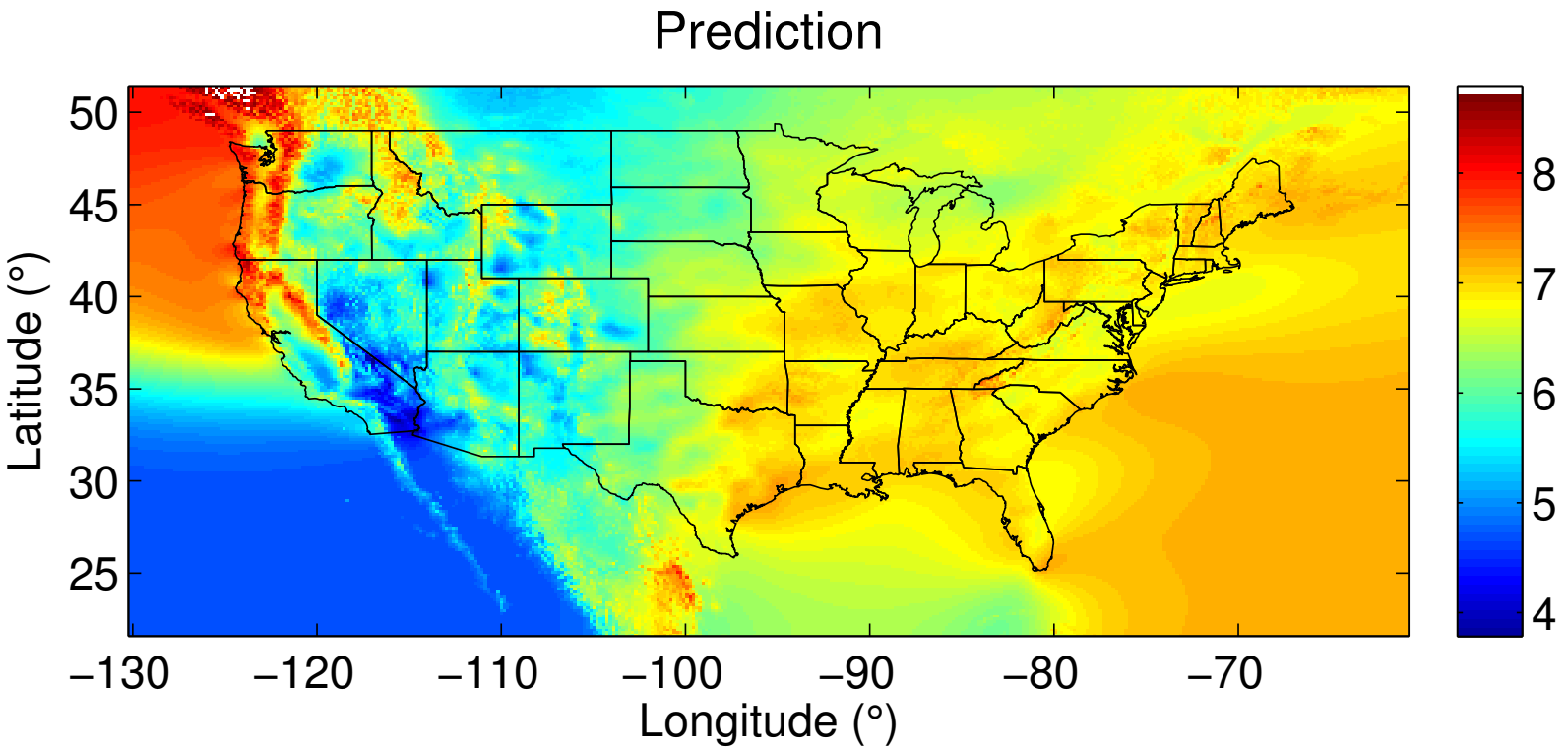
So how do we model $\boldsymbol{H}(s)$?

We need to model a 2×2 symmetric positive definite matrix.

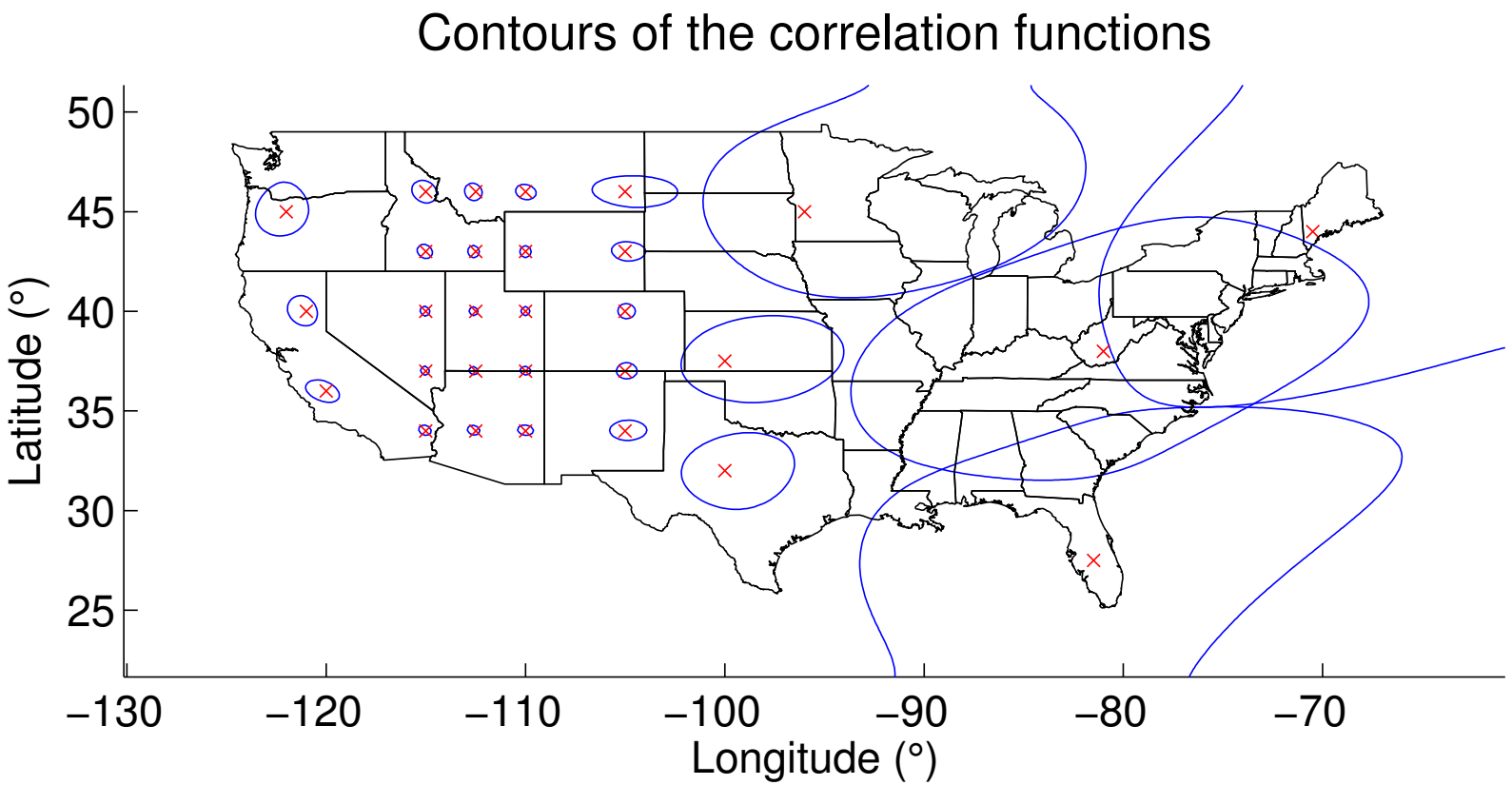
$$\boldsymbol{H}(s) = \gamma(s)\boldsymbol{I} + \boldsymbol{v}(s)\boldsymbol{v}(\boldsymbol{s})^T.$$

- $\gamma(s)$ is the amount "baseline" diffusion,
- $\boldsymbol{v}(s)$ is the principle eigenvector of $\boldsymbol{H}$.
- The amount of excess diffusion in the $\boldsymbol{v}$ direction (compared to the the orthogonal direction) is $1 + \gamma^{-1} \|\boldsymbol{v}\|^2$.
- We model $\gamma(s)$, $v_1(s)$ and $v_2(s)$ as (stationary) Gaussian random fields. We may include covariates etc.

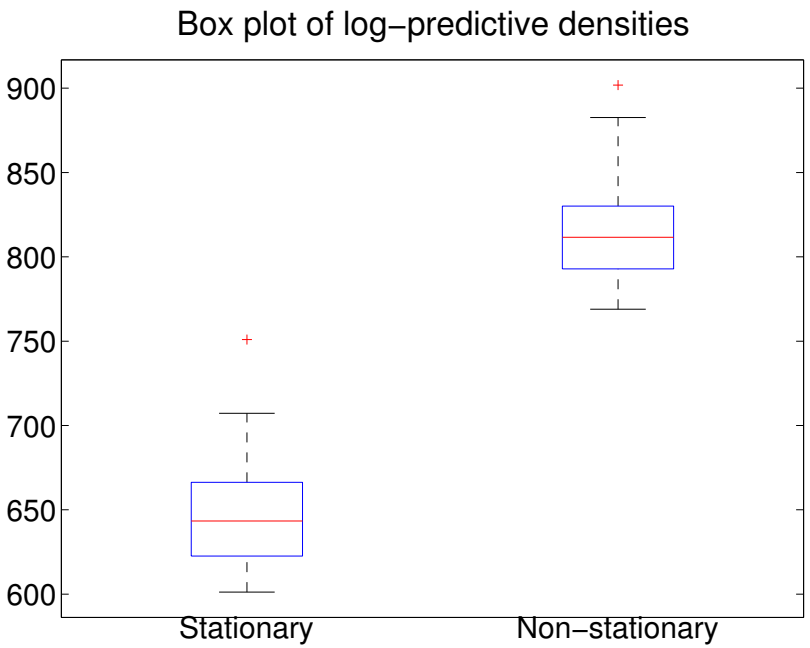
November rain

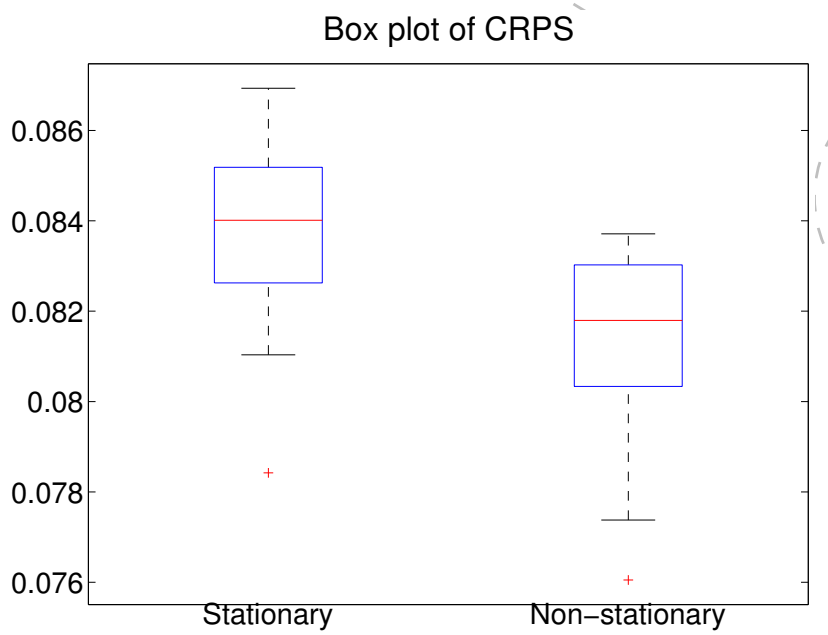


Purple rain


Contours of the correlation functions
Latitude (°)
50
45
40
35
30
25
−130
−120
−110
−100
−90
−80
−70
Longitude (°)

Blame it on the rain





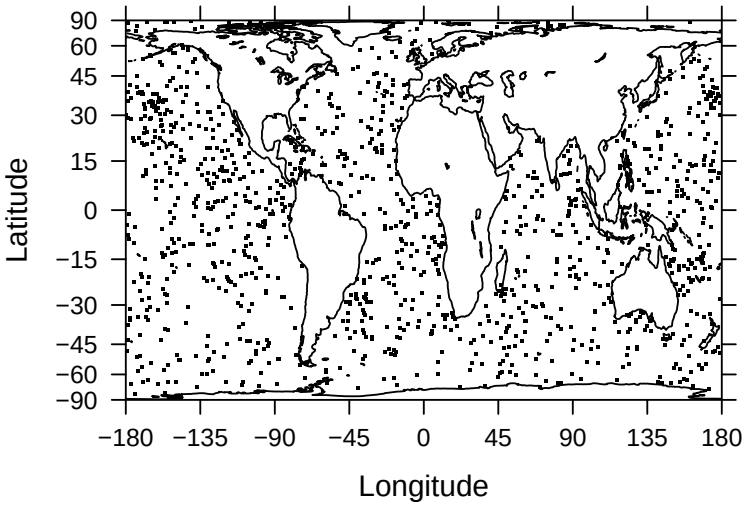
Outline

A non-convex region on a sphere!

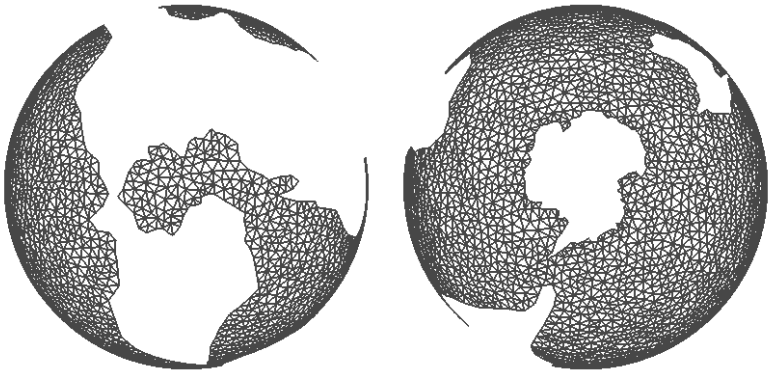
Rectangles are boring...

- We are often interested in bounded observation windows on the sphere
- Guiding application: Freak waves!
- How do we build a random field on the oceans? Just use an SPDE!

Some simulated data

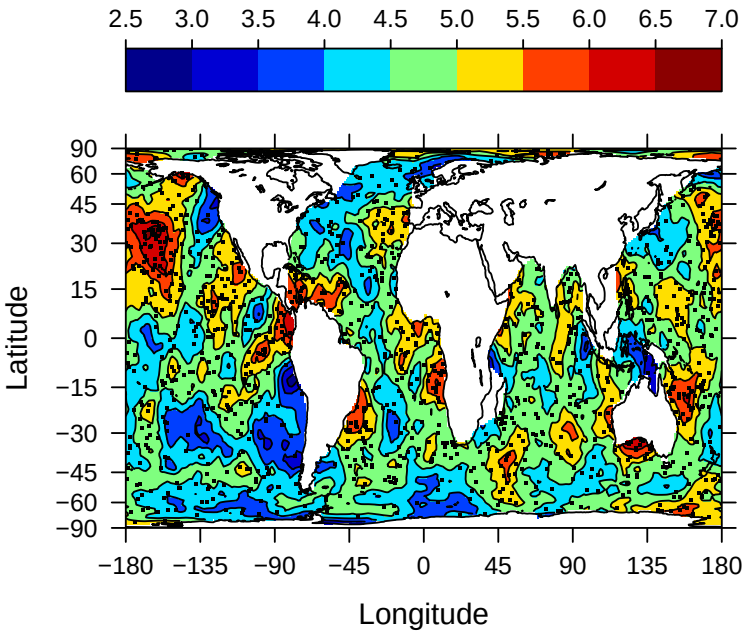


(e) Simulated data

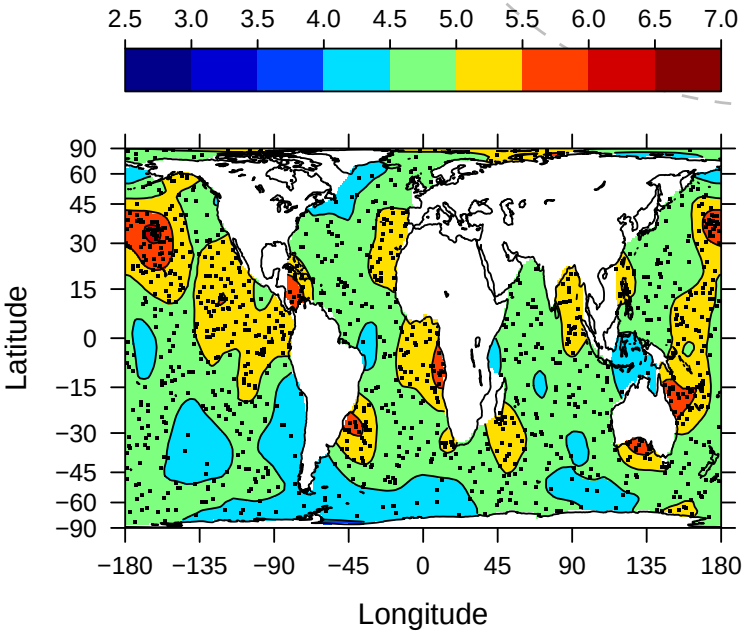


(f) A mesh over the oceans

Posterior mean

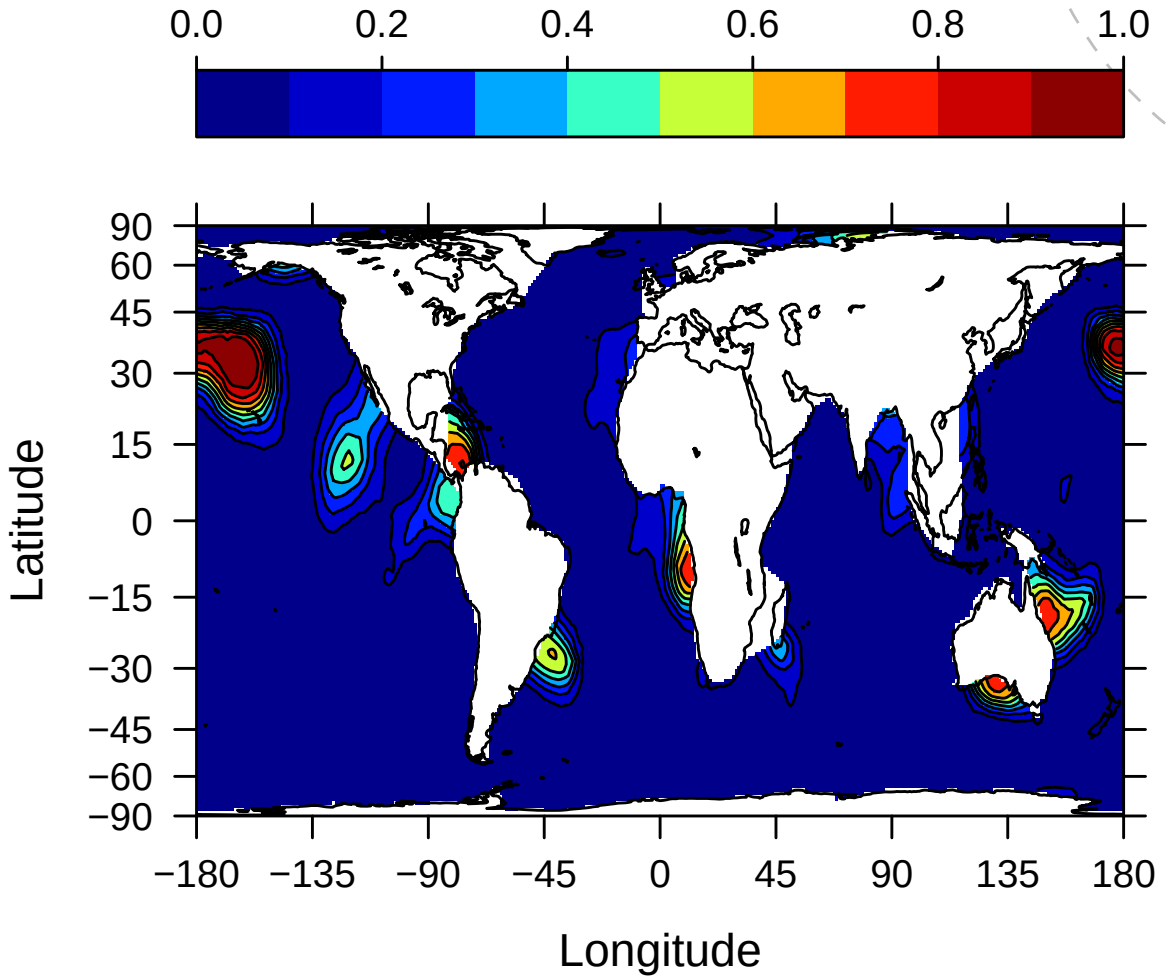


(g) True surface



(h) Reconstructed surface

What people actually want



Posterior risk map $P(\log(\lambda(s)) > 5.5)$. Easily computed with INLA.

Outline

Stuff to do

We are working on a bunch of extensions:

- Extreme value modelling where latent fields enter in the location and scale parameters (with Óli Páll and Birgir)
- Non-stationary log-Gaussian Cox processes
- Multivariate log-Gaussian point processes
- Multivariate extremes
- Simple inverse problems
- Moving appropriate versions of these models into INLA

The aim is always to construct flexible, interpretable models that are computationally feasible for large problems.