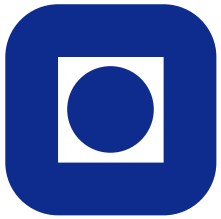


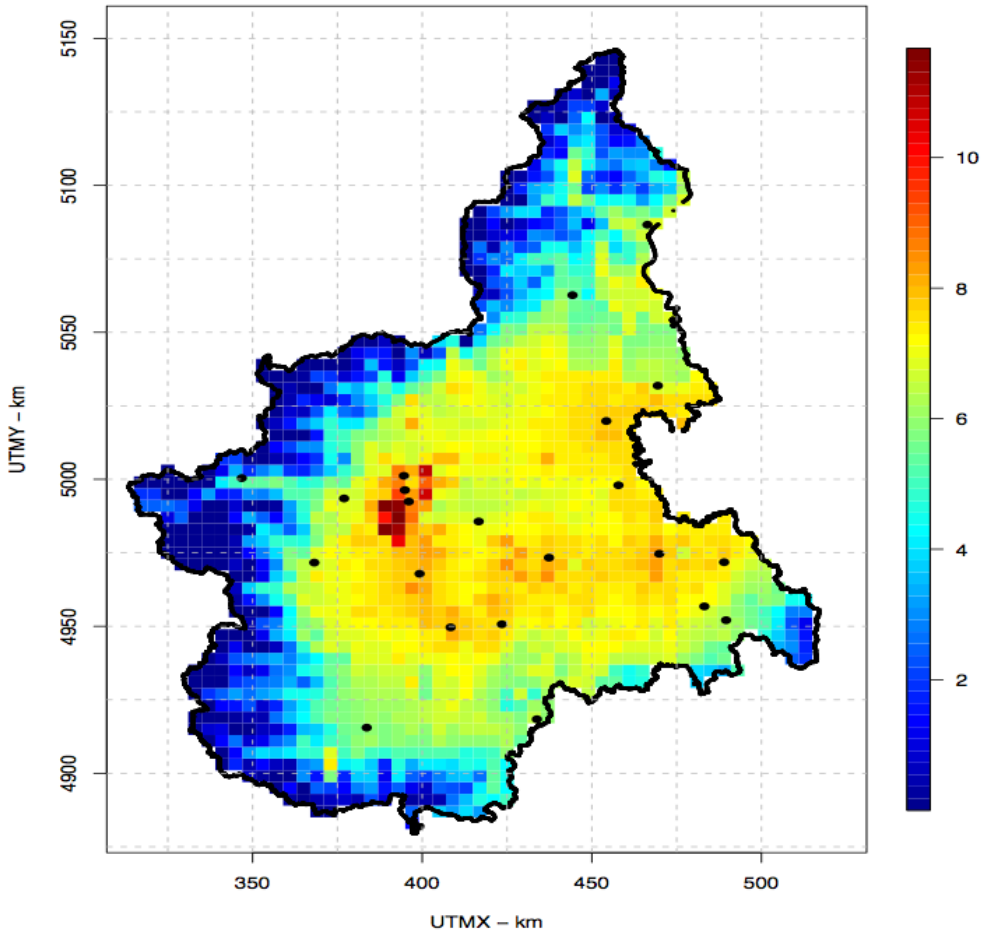
NTNU
Norwegian University of
Science and Technology

Computational and modelling extensions

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

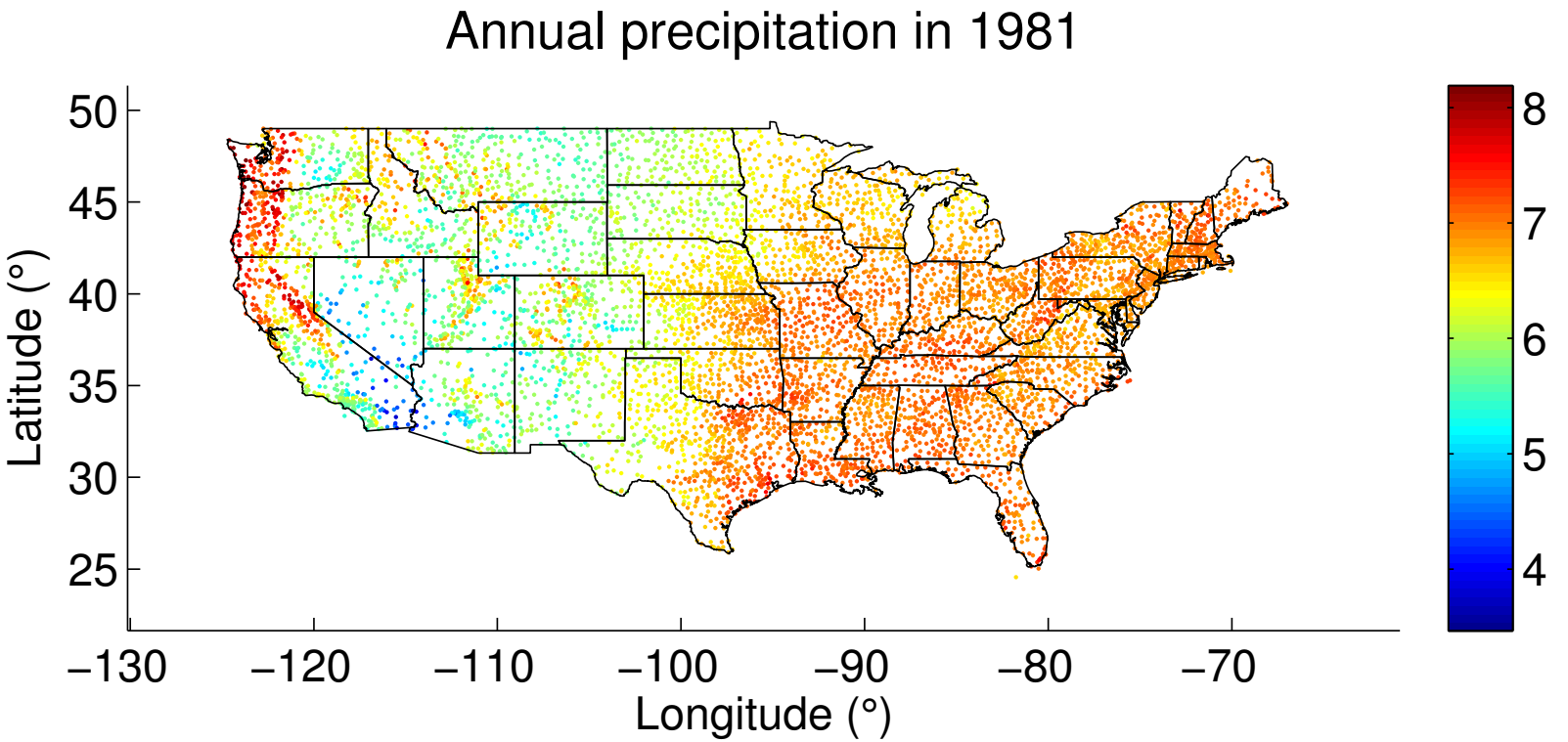
Outline

Spatial mapping

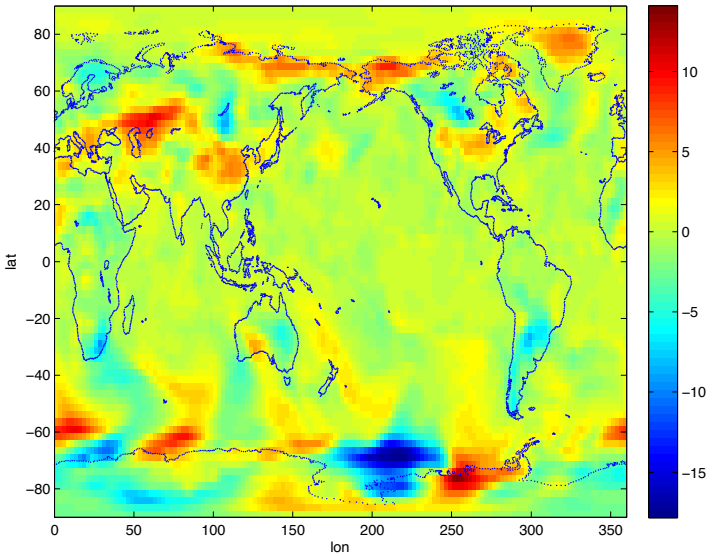


Daily PM-10 concentration in the Piemonte region, 10/05–03/06.

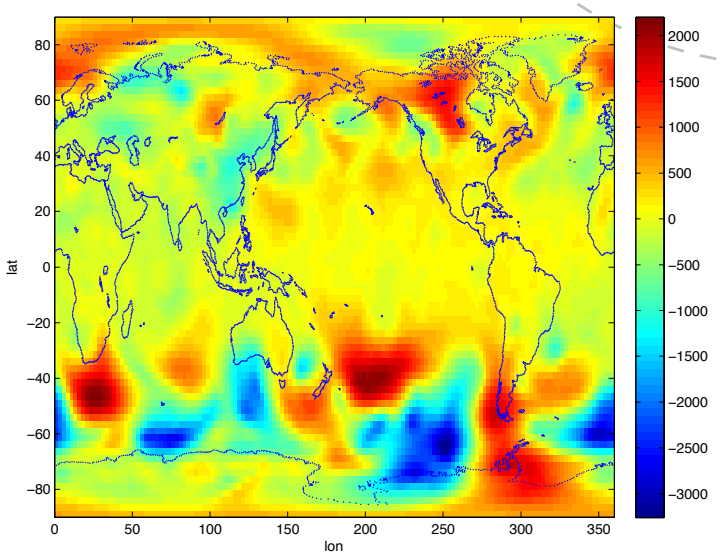
Large-scale rainfall mapping



There's power in a union

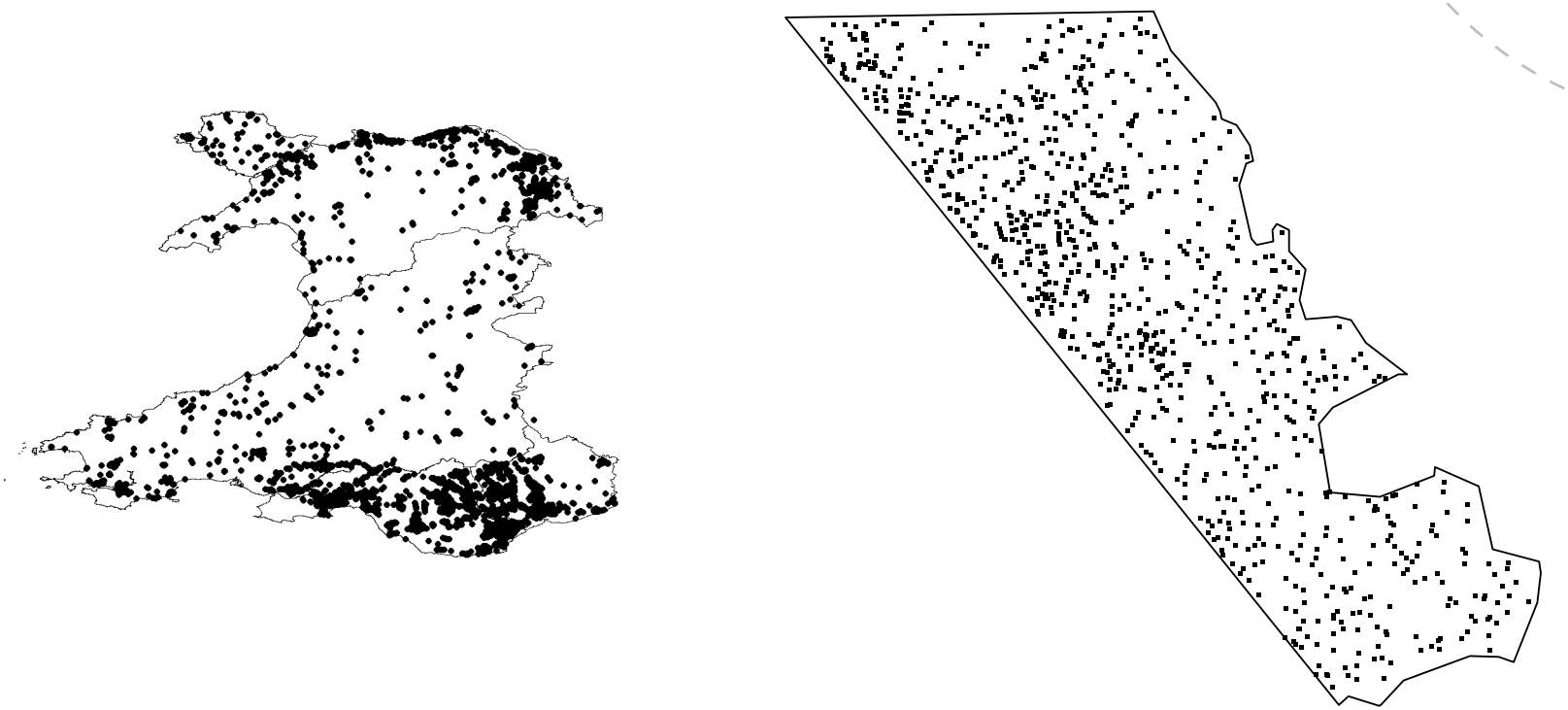


(a) Temperature



(b) Pressure

Crime and Koalas



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

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

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

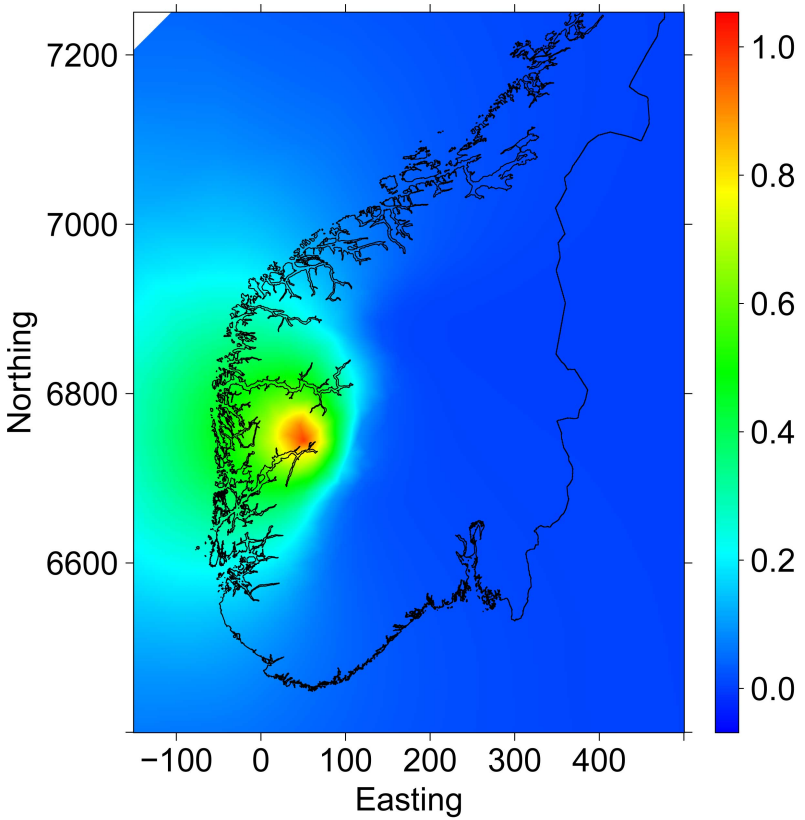
where

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

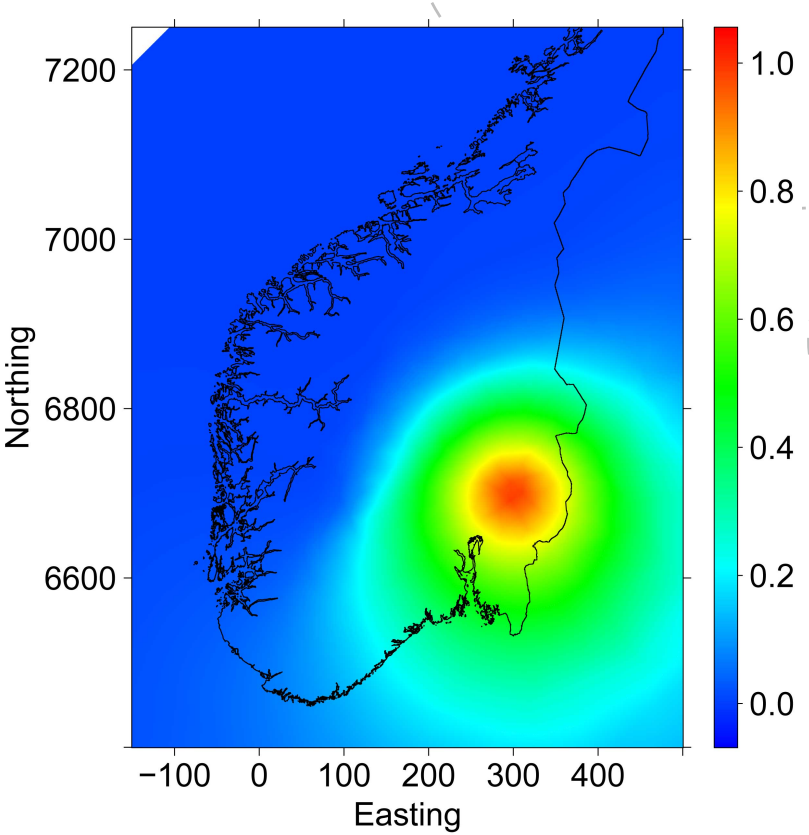
They take

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

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?

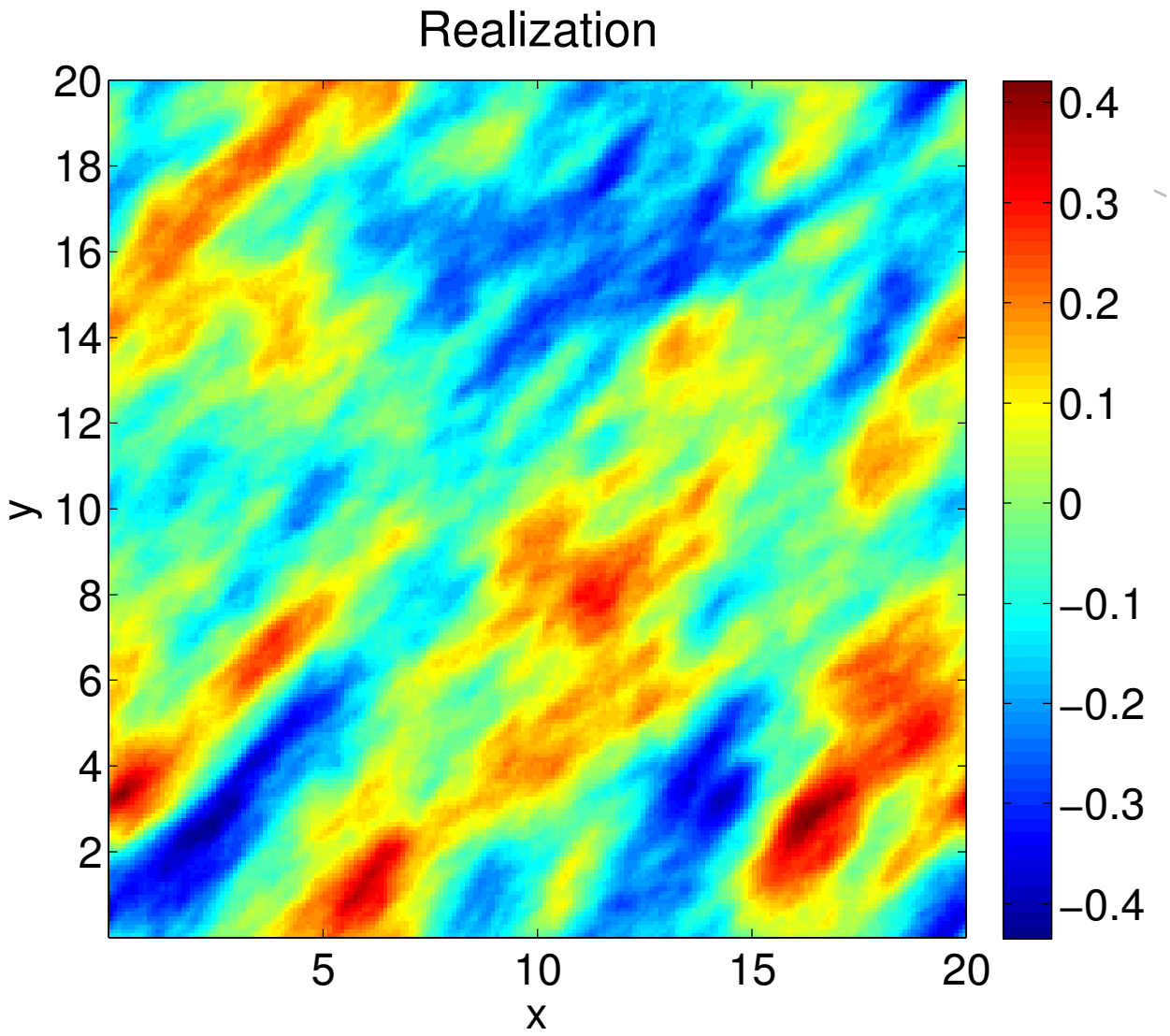
— Go all maths-y and start talking about Riemannian metrics. (blegh)

— Be a bit physics-y and talk about diffusion.

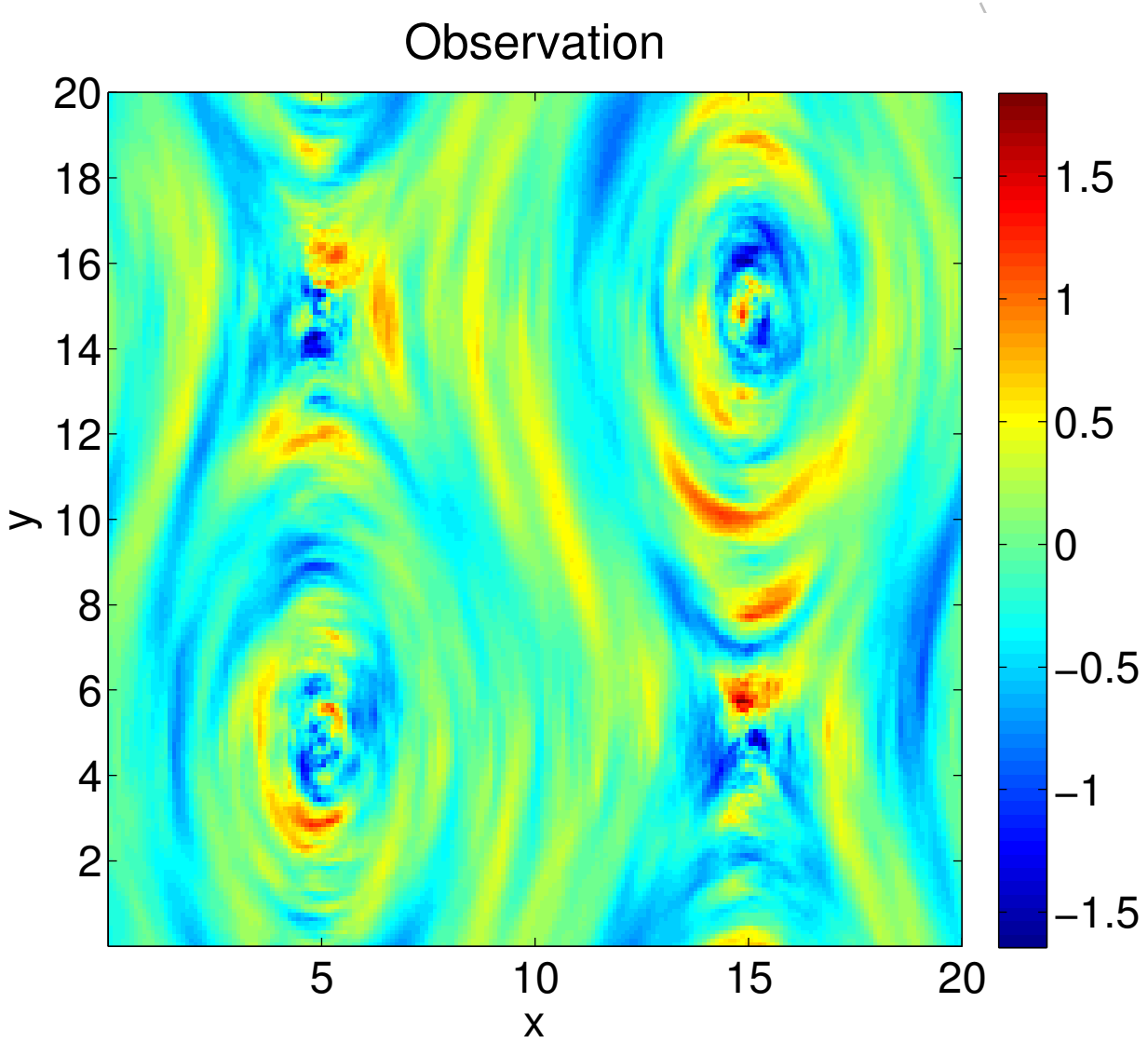
If we define the local diffusion tensor (matrix) by $\boldsymbol{H}(s)$, then we can build a model where the important directions and their relative distances are modelled by the eigenvectors and eigenvalues of $\boldsymbol{H}$.

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

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 $H(s)$



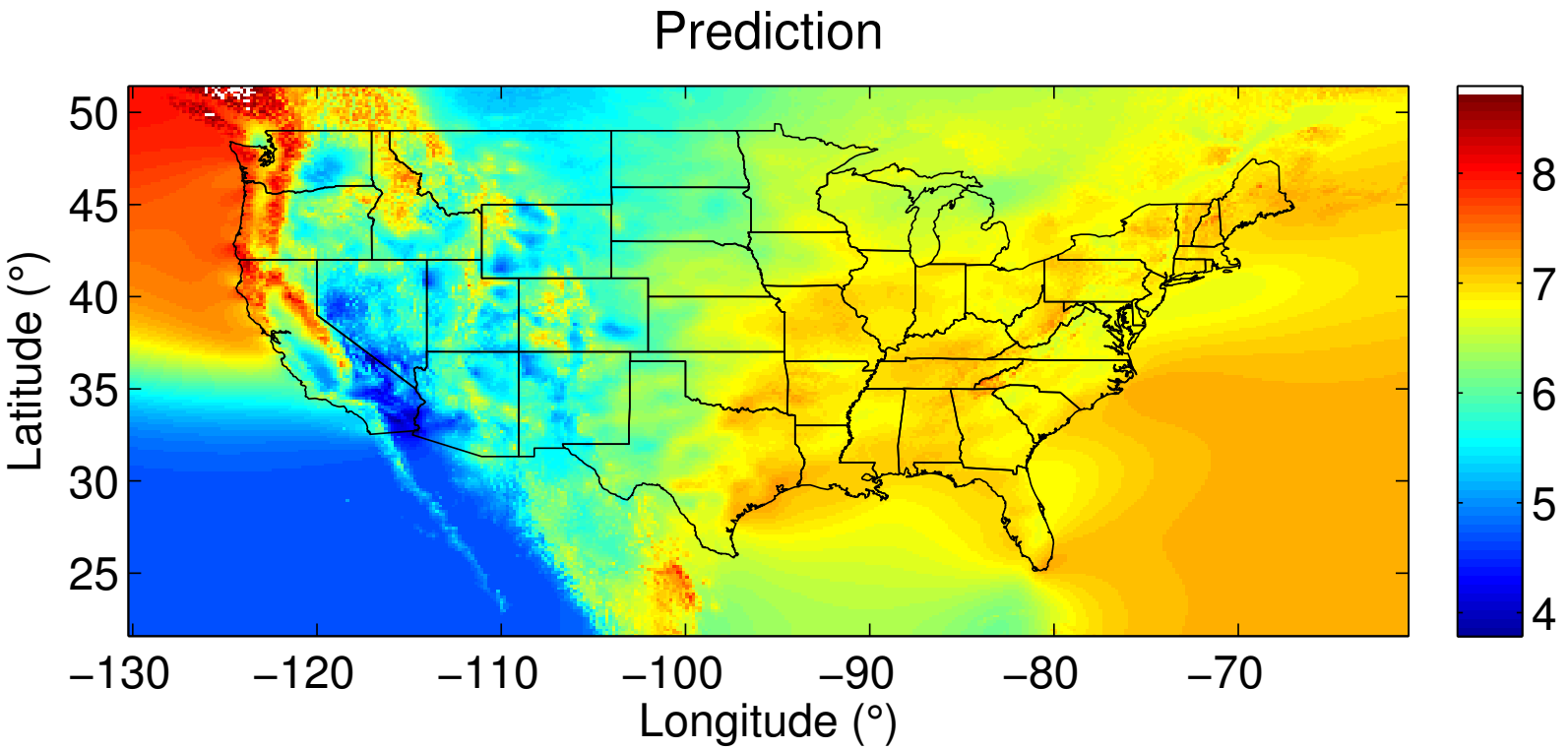
So how do we model $\boldsymbol{H}(s)$?

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

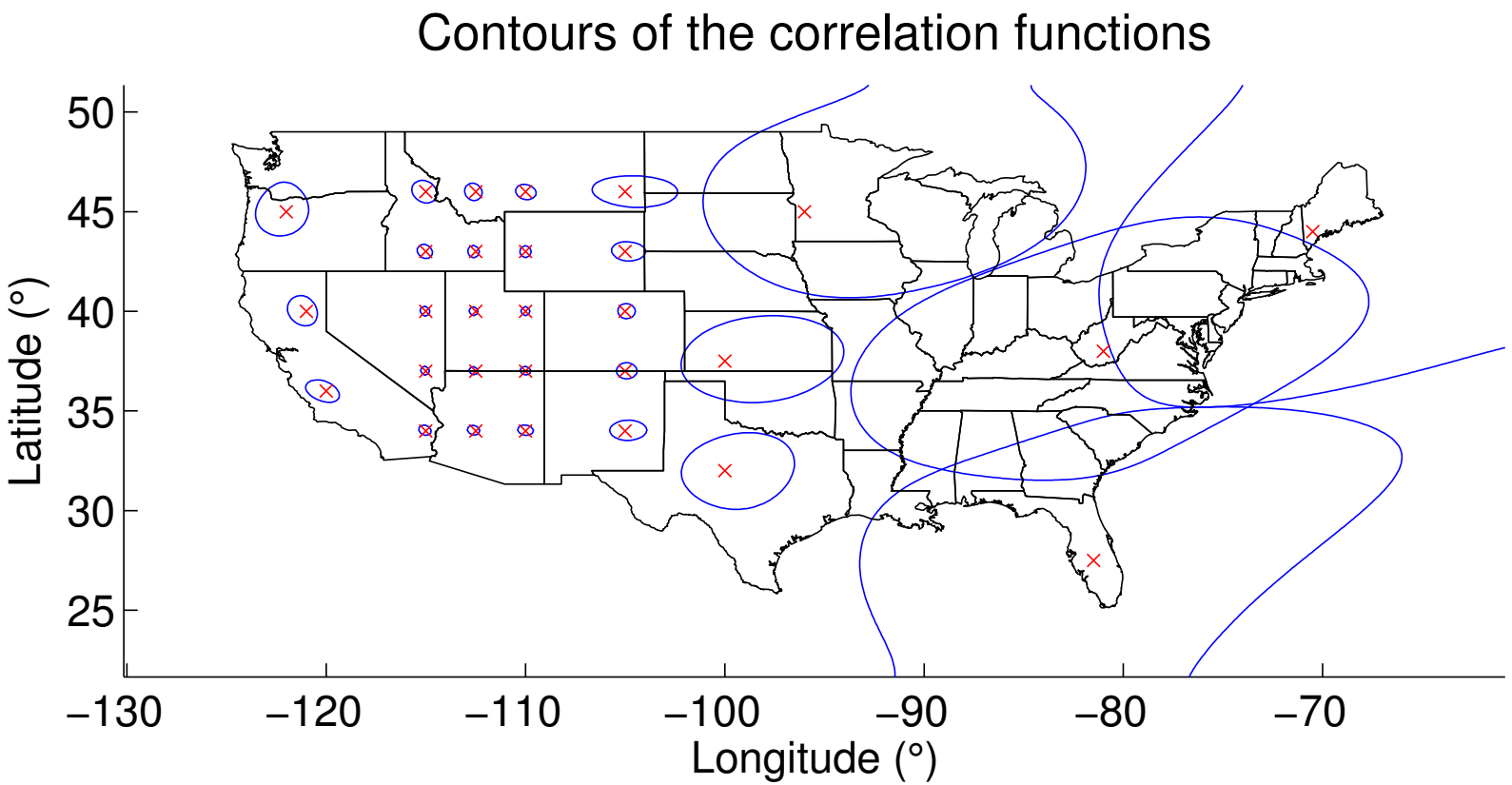
$$\boldsymbol{H}(s) = \gamma(s)\boldsymbol{I} + \boldsymbol{v}(s)\boldsymbol{v}(\boldsymbol{s})^T.$$

- $\gamma(s)$ 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.

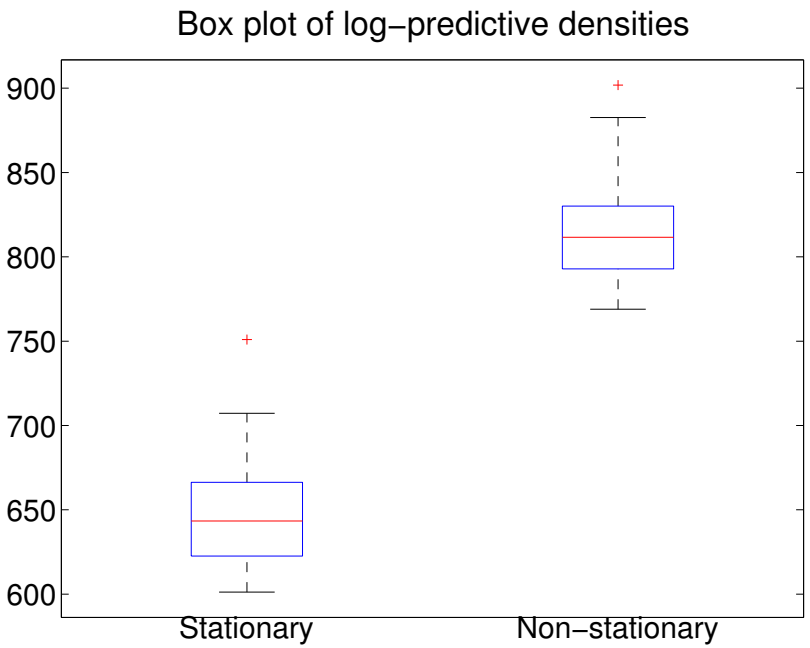
November rain

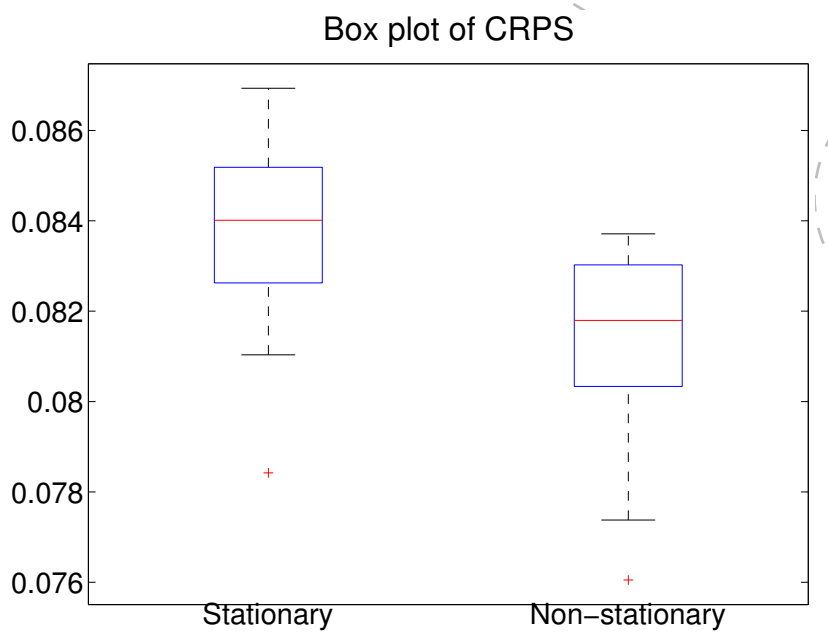


Purple rain


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





Moving on up

— Sometimes, quantities of interest come in correlated blocks. (Temperature and Pressure)

Moving on up

— Sometimes, quantities of interest come in correlated blocks. (Temperature and Pressure)

— Sometimes, quantities of interest don't come on $\mathbb{R}^2$. (Surprise!)

Moving on up

— Sometimes, quantities of interest come in correlated blocks. (Temperature and Pressure)

— Sometimes, quantities of interest don't come on $\mathbb{R}^2$. (Surprise!)

— The first problem can be attacked with "Linear models of co-regionalisation" or novel covariance function methods.

Moving on up

- Sometimes, quantities of interest come in correlated blocks. (Temperature and Pressure)
- Sometimes, quantities of interest don't come on $\mathbb{R}^2$. (Surprise!)
- The first problem can be attacked with "Linear models of co-regionalisation" or novel covariance function methods.
- The second problem involves fun with spherical harmonics...

Moving on up

- Sometimes, quantities of interest come in correlated blocks. (Temperature and Pressure)
- Sometimes, quantities of interest don't come on $\mathbb{R}^2$. (Surprise!)
- The first problem can be attacked with "Linear models of co-regionalisation" or novel covariance function methods.
- The second problem involves fun with spherical harmonics...
- These "problems" is essentially trivial with SPDE approaches

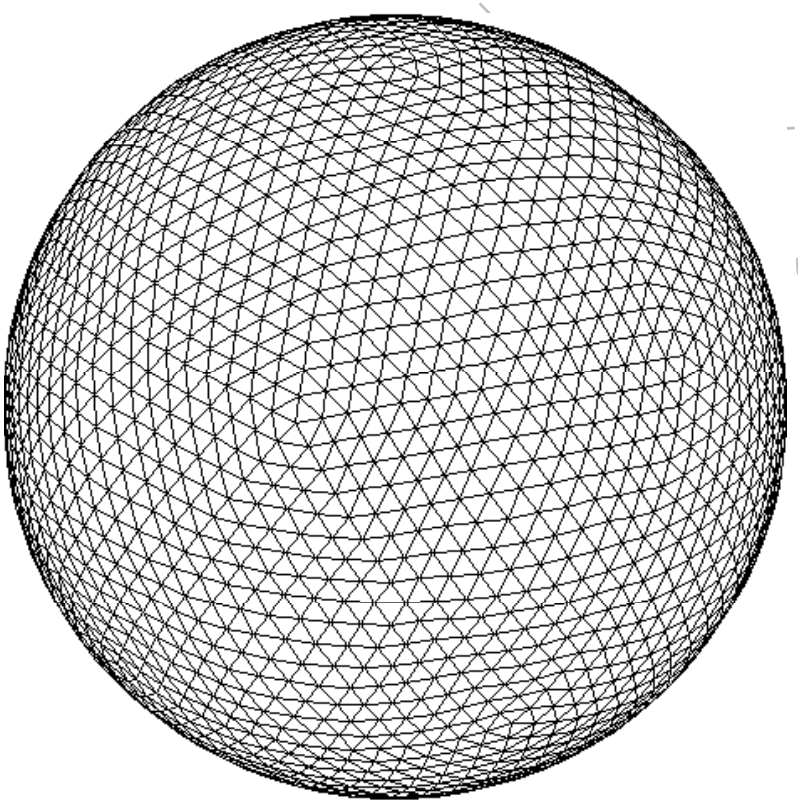
Systems of Stochastic Partial Differential Equations

$$\begin{pmatrix} L_{11} & L_{12} & \dots & L_{1k} \\ L_{21} & L_{22} & \dots & L_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ L_{k1} & L_{k2} & \dots & L_{kk} \end{pmatrix} \begin{pmatrix} x_1(\mathbf{s}) \\ x_2(\mathbf{s}) \\ \vdots \\ x_k(\mathbf{s}) \end{pmatrix} = \begin{pmatrix} W_1(\mathbf{s}) \\ W_2(\mathbf{s}) \\ \vdots \\ W_k(\mathbf{s}) \end{pmatrix}$$

L_{ij} are differential operators and W_i are (possibly not identical) noises.

- Just apply FE method to each element of the LHS matrix
- Normal problem with multivariate GRFs - overparameterisation! Take LHS to be triangular.

Manifolds/curved-spaces



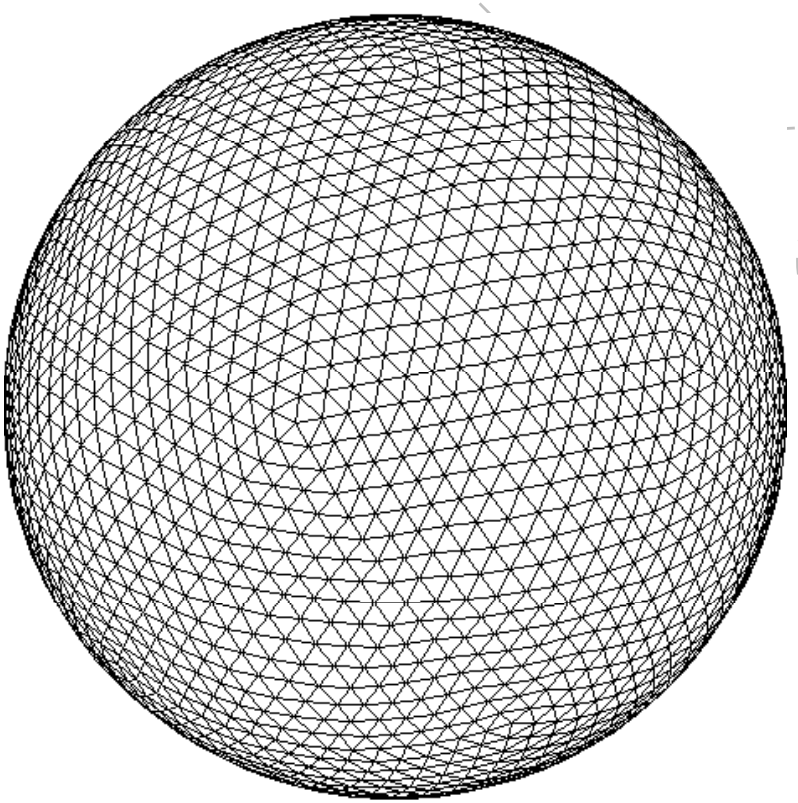
Manifolds/curved-spaces

Define Matérn fields using

$$(\kappa^2 - \Delta)^{\alpha/2} \boldsymbol{x}(\boldsymbol{s}) = \boldsymbol{W}(\boldsymbol{s})$$

on the manifold $\mathcal{S}$, driven by Gaussian "white noise" on $\mathcal{S}$

$$\text{Cov}(\boldsymbol{W}(A_i), \boldsymbol{W}(A_j)) = \int_{A_i \cap A_j} d\mathcal{S}(\boldsymbol{s})$$



The advantage of SPDE approaches

Everything stays the same

(just change *ds* to the surface measure)

Joint modelling of temperature and pressure

— There is small scale data on the US. Our methods had better predictive performance than covariance function based methods of Gneiting et al.

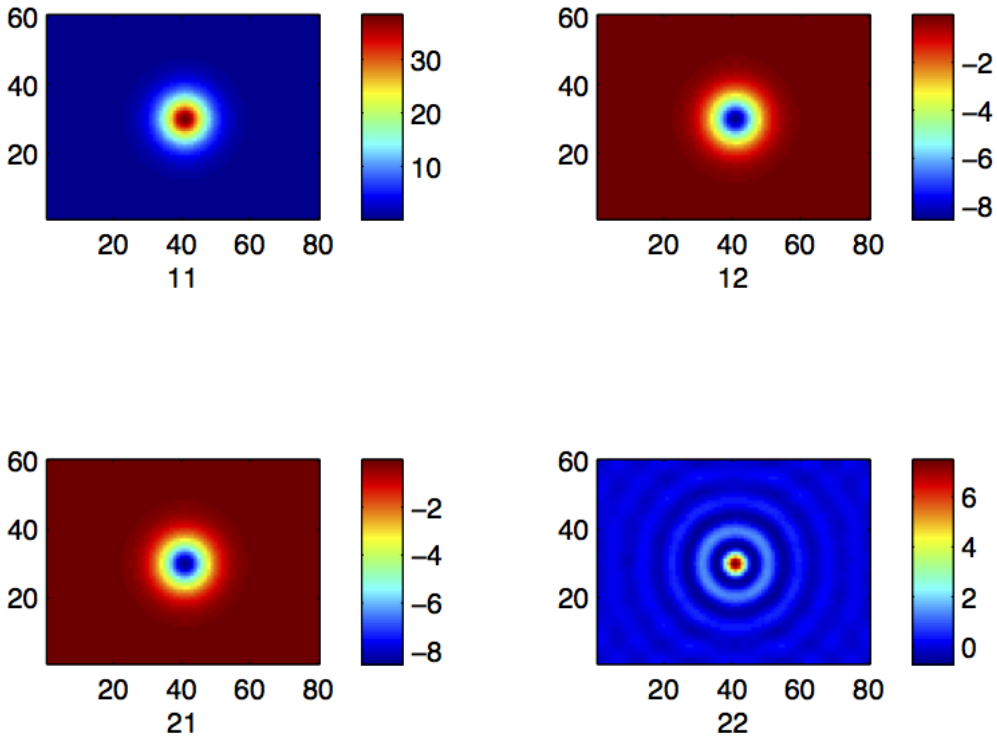
Joint modelling of temperature and pressure

- There is small scale data on the US. Our methods had better predictive performance than covariance function based methods of Gneiting et al.
- There is global re-analysis data available to play with.

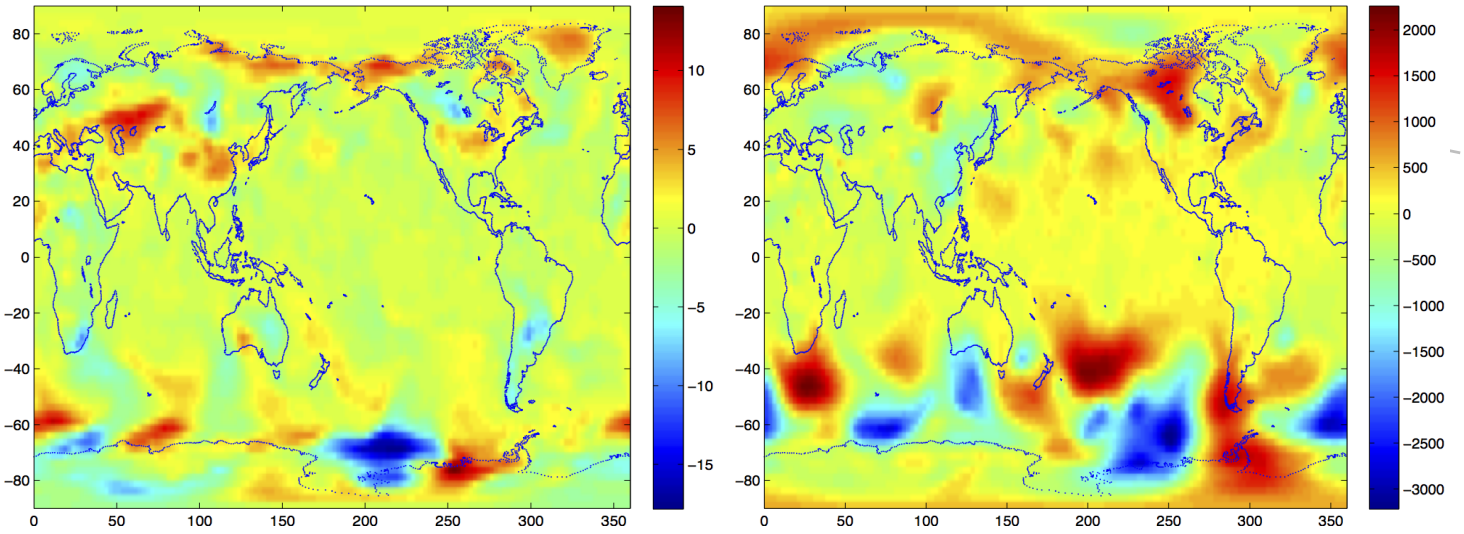
Joint modelling of temperature and pressure

- There is small scale data on the US. Our methods had better predictive performance than covariance function based methods of Gneiting et al.
- There is global re-analysis data available to play with.
- The challenge here is that pressure typically oscillating over a large spatial scale

One oscillating component



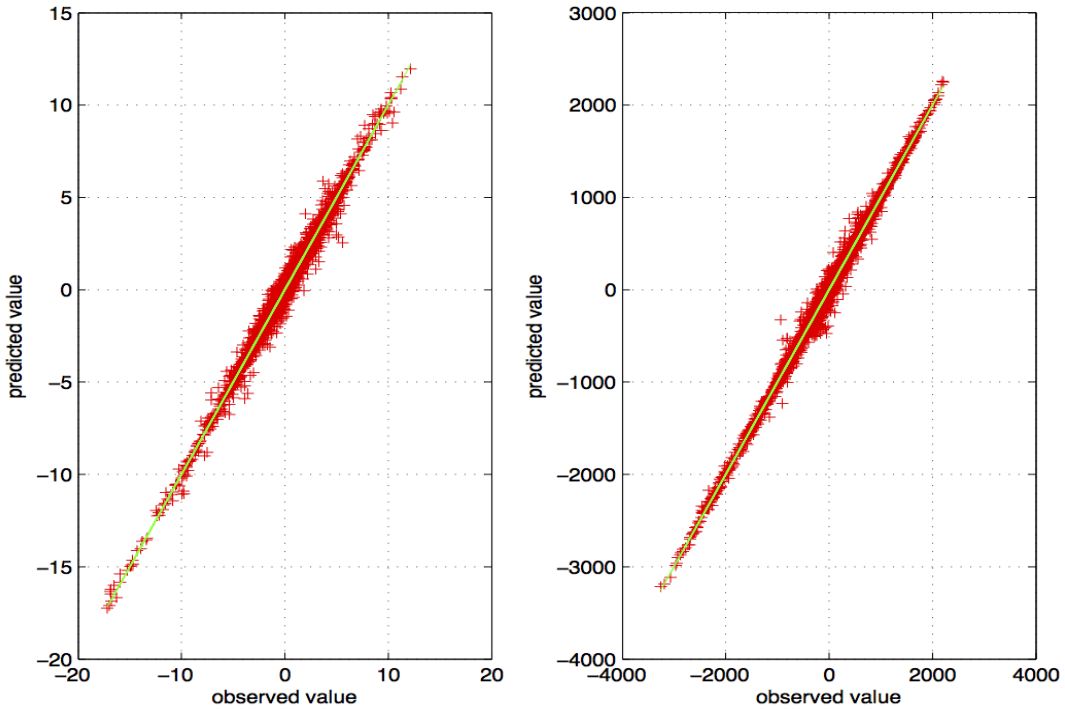
Estimated fields



(e) (f)

Estimated bivariate random fields for ERA 40 database with temperature (a) and pressure (b)

How well did we do?



Prediction for the bivariate random fields at another 5000 data points for temperature (left) and pressure (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

The village green preservation society

Direct methods

All methods from sampling from a Gaussian require a factorisation of the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{R}\boldsymbol{R}^T$ or the precision matrix $\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1} = \boldsymbol{L}\boldsymbol{L}^{\boldsymbol{T}}$. This is always† done with a Cholesky factorisation.

- Fine for small problems.
- Computes the determinant for free!
- Obviously doesn't scale well.....

What is a “massive” problem

Folk definition

A problem becomes *massive* when the methods I want to use no longer work.

- Solving “massive” problems require investment in modelling.
- Solving “massive” problems require investment in computation.
- Solving “massive” problems requires compromise.

Inverse problems are “massive” problems.

Whatever happened to Baby Jane?

We've forgotten θ!

- We need to keep track of the "change-in-volume" $|\boldsymbol{Q}(\theta^*)|/|\boldsymbol{Q}(\theta)|$.
- (For technical reasons, we need to estimate each determinant separately)
- ITERATIVE METHODS DO NOT COMPUTE DETERMINANTS
- DETERMINANTS ARE *very* HARD TO COMPUTE

The village green preservation society

Direct methods

All methods from sampling from a Gaussian require a factorisation of the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{R}\boldsymbol{R}^T$ or the precision matrix $\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1} = \boldsymbol{L}\boldsymbol{L}^{\boldsymbol{T}}$. This is always† done with a Cholesky factorisation.

- Fine for small problems.
- Computes the determinant for free!
- Obviously doesn't scale well.....

But we can do better

We can construct approximate samples

- Deterministic, modern *iterative* methods from numerical linear algebra
- They are fast!
- They can scale!
- Best cases, we can get $\mathcal{O}(n \log n)$ or even $\mathcal{O}(n)$ samples!

I like big buts

But...

The problem with iterative methods

Iterative methods (LSQR for least squares sampling, and the matrix function methods) have one major drawback:

THEY DON'T COMPUTE THE LOG-DETERMINANT!

DETERMINANTS ARE VERY DIFFICULT TO COMPUTE!

Idea 1: Approximate factorisations

Concept: Even if we don't want to use the approximate factorisation to compute the sample, it will give a decent approximation to the determinant.

Problem: We have no control over the error. Furthermore, there is no way of checking how good your answer is.

Idea 2: Matrix functions (Bai et al '96)

If the Cholesky decomposition is unavailable, a better way is to use the identity

$$\log(\det(A)) = \mathrm{tr}(\log(A)) = \sum_{i=1}^{n} e_i^T \log(A) e_i.$$

Is there a cheap way to approximate $\mathrm{tr}(\log(A))$*?*

A Stochastic Estimator of the Trace

Theorem (Hutchinson '90)

Let $B \in \mathbb{R}^{n \times n}$ be a symmetric matrix with non-zero trace. Let Z be the discrete random variable which takes the values $-1, 1$ each with probability $1/2$ and let z be a vector of n independent samples from Z. Then $z^T Bz$ is an unbiased estimator of $\text{tr}(B)$ and Z is the unique random variable amongst zero mean random variables for which $z^T Bz$ is a minimum variance, unbiased estimator of $\text{tr}(B)$.

Therefore

$$\log(\det(A)) = \mathbb{E}\left(z^T \log(A) z\right).$$

This can be estimated using a Monte Carlo method.

Can we control the variance?

For large Gaussian problems, $\tilde{\mathcal{L}} = \boldsymbol{z}^T \log(\boldsymbol{Q})\boldsymbol{z}$ is an unbiased estimator.

- The best choice of $\boldsymbol{z}$ has i.i.d. ± 1 random variables.
- But the variance can be very large.
- Can we use the structure of the problem to reduce it?

Can we design a better set of random variables

Let's take a close look at $\boldsymbol{z}^T \log(\boldsymbol{Q})\boldsymbol{z}$.

— For any vector $\boldsymbol{z} \in -1, 1^n$

$$\boldsymbol{z}^T \mathbf{log}(Q)\boldsymbol{z} = \sum_{i=1}^{n} [\log(\boldsymbol{Q})]_{ii} + 2 \sum_{i \neq j} z_i z_j [\log(\boldsymbol{Q})]_{ij}$$

Can we design a better set of random variables

Let's take a close look at $\boldsymbol{z}^T \log(\boldsymbol{Q})\boldsymbol{z}$.

— For any vector $\boldsymbol{z} \in -1, 1^n$

$$\boldsymbol{z}^T \mathbf{log}(Q)\boldsymbol{z} = \sum_{i=1}^{n} [\log(\boldsymbol{Q})]_{ii} + 2 \sum_{i \neq j} z_i z_j [\log(\boldsymbol{Q})]_{ij}$$

— Clearly, the off diagonal elements of $\boldsymbol{Q}$ "pollute" the solution.

Can we design a better set of random variables

Let's take a close look at $\boldsymbol{z}^T \log(\boldsymbol{Q})\boldsymbol{z}$.

— For any vector $\boldsymbol{z} \in -1, 1^n$

$$\boldsymbol{z}^T \mathbf{log}(Q)\boldsymbol{z} = \sum_{i=1}^{n} [\log(\boldsymbol{Q})]_{ii} + 2 \sum_{i \neq j} z_i z_j [\log(\boldsymbol{Q})]_{ij}$$

— Clearly, the off diagonal elements of $\boldsymbol{Q}$ "pollute" the solution.

— Fun fact: $[\log(\boldsymbol{Q})]_{ij} \leq e^{-\kappa d(i,j)}$, where $d(i,j)$ is the graph distance.

Can we design a better set of random variables

Let's take a close look at $\boldsymbol{z}^T \log(\boldsymbol{Q})\boldsymbol{z}$.

- For any vector $\boldsymbol{z} \in -1, 1^n$

$$\boldsymbol{z}^T \log(Q)\boldsymbol{z} = \sum_{i=1}^{n} [\log(\boldsymbol{Q})]_{ii} + 2 \sum_{i \neq j} z_i z_j [\log(\boldsymbol{Q})]_{ij}$$

- Clearly, the off diagonal elements of $\boldsymbol{Q}$ "pollute" the solution.
- Fun fact: $[\log(\boldsymbol{Q})]_{ij} \leq e^{-\kappa d(i,j)}$, where $d(i,j)$ is the graph distance.
- Can we use this? Decompose $\boldsymbol{z} = \sum_{c \in C} \boldsymbol{z}_c$ where C is a partition of $\{1, 2, \ldots, n\}$.

Can we design a better set of random variables

Let's take a close look at $\boldsymbol{z}^T \log(\boldsymbol{Q})\boldsymbol{z}$.

— For any vector $\boldsymbol{z} \in -1, 1^n$

$$\boldsymbol{z}^T \log(Q)\boldsymbol{z} = \sum_{i=1}^{n} [\log(\boldsymbol{Q})]_{ii} + 2 \sum_{i \neq j} z_i z_j [\log(\boldsymbol{Q})]_{ij}$$

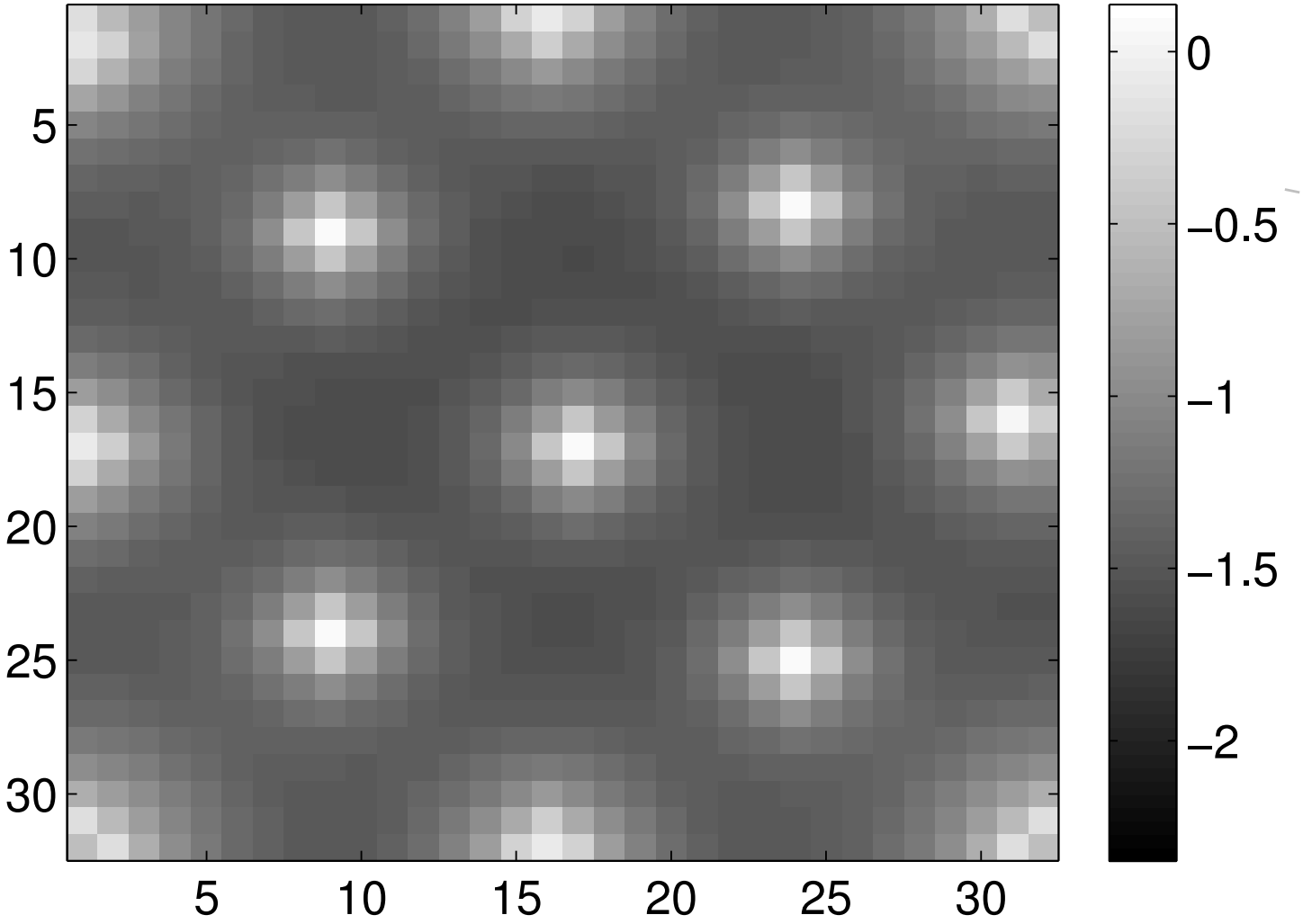
— Clearly, the off diagonal elements of $\boldsymbol{Q}$ "pollute" the solution.

— Fun fact: $[\log(\boldsymbol{Q})]_{ij} \leq e^{-\kappa d(i,j)}$, where $d(i,j)$ is the graph distance.

— Can we use this? Decompose $\boldsymbol{z} = \sum_{c \in C} \boldsymbol{z}_c$ where C is a partition of $\{1, 2, \ldots, n\}$.

— We want to make sure near-by points aren't in the same c. *Solution: Graph colouring!*

A probing vector



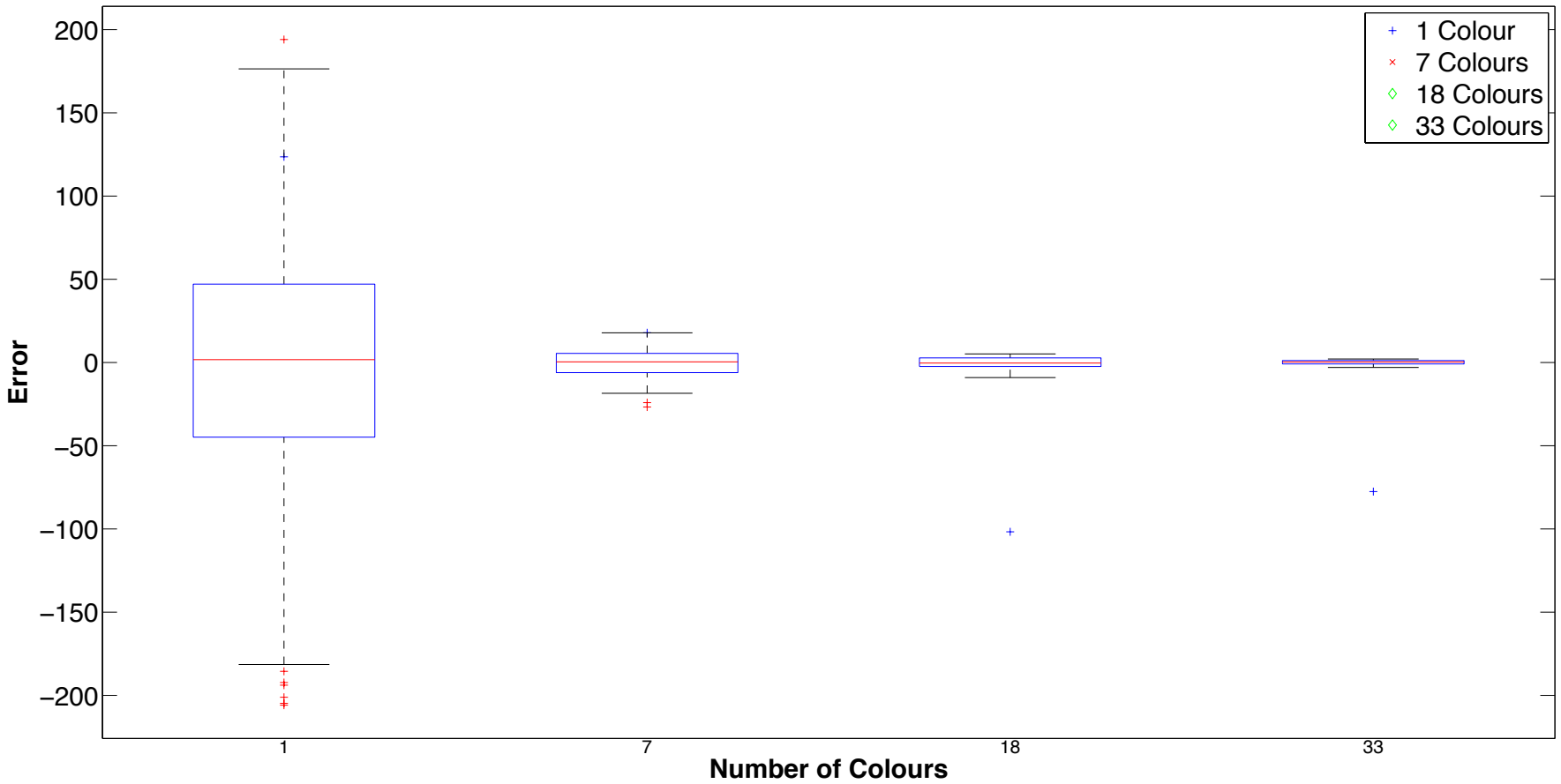
Putting it together

Here is the procedure that works best:

1. Pick a value p and produce a graph colouring of $\boldsymbol{Q}^p$.
2. For each colour c , construct a vector $\boldsymbol{z}_c$ that is randomly ± 1 (w.p. $1/2$) at the vertices of that colour and zero everywhere else
3. Use these vectors in Hutchinson's estimator of $\log(\det(\boldsymbol{Q}))$

Sometimes it's worth doing a change of basis (wavelet transform).

Variance reduction



Outline

Some things we don't yet know how to do

- How to really scale these things?
- How bad is our MCMC allowed to be?
- What happens when multiple random fields interact in non-linear way?
- How do we really do things like multivariate space/time species distribution maps?

More things we don't know how to do

- PRIORS!: These are *very* important for some models. How should we choose them
- Model checking
- The effect of mis-specification in finite dimensional models
- How to deal with the extra flexibility non-stationarity brings
- How should we parameterise space/time non-stationarity to make it interpretable for "real people"?