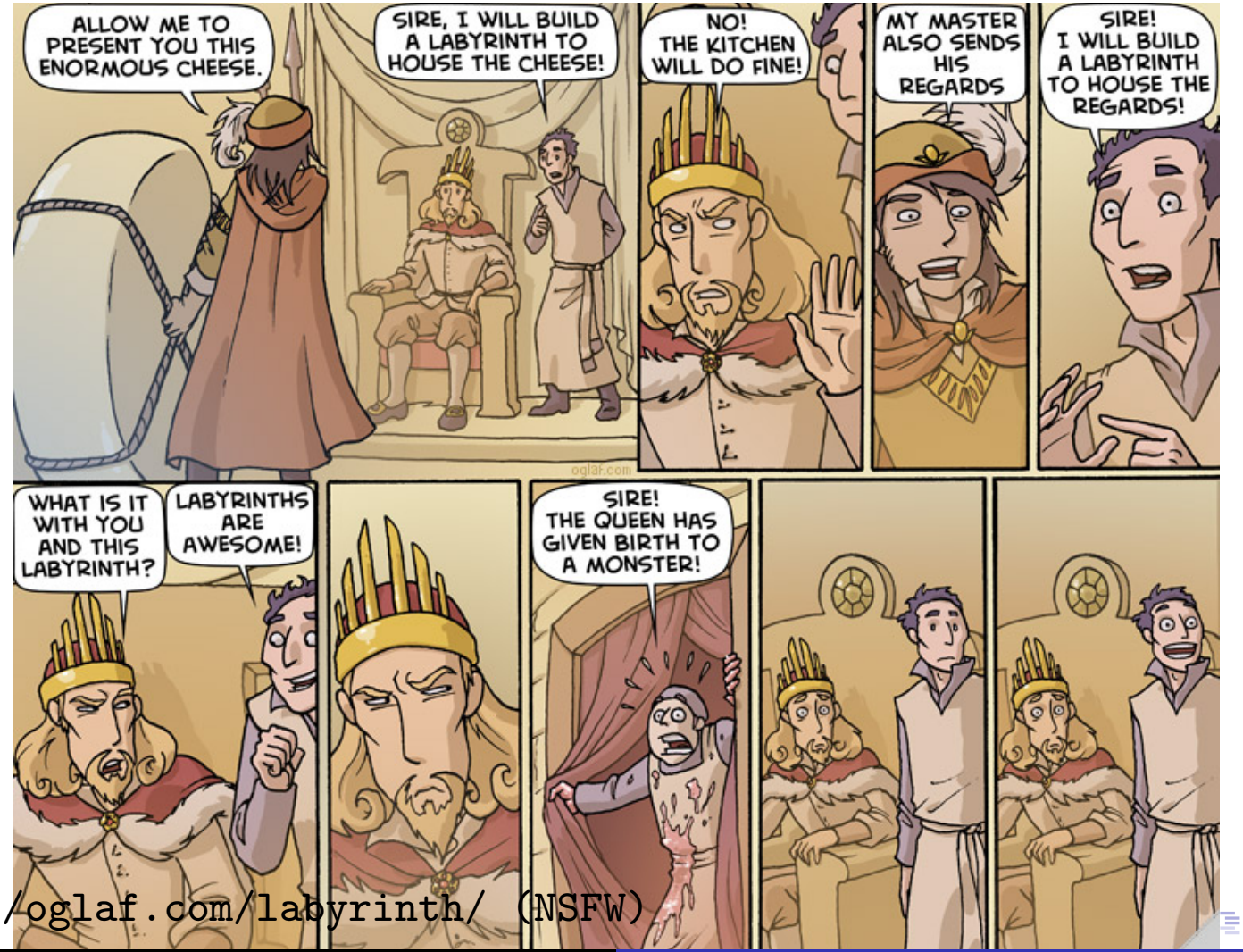
Approximating Gaussian random fields by Gaussian Markov random fields: A decade on

Daniel Simpson
Joint work with Finn Lindgren and Håvard Rue (NTNU)
Ian Turner, Chris Strickland and Tony Pettitt (QUT)

Department of Biosciences, University of Helsinki

May, 2012

"The minotaur justifies the labyrinth"—Jorge Luis Borges



http://oglaf.com/labyrinth/ (NSFW)

Outline

"Let's start from the very beginning..."

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.
- Even harder for multivariate (Xianping Hu), non-stationary (Geir-Arne Fuglstad) or spatiotemporal processes.

Second solution

Hi-Defn: Gaussian Random Fields

A Gaussian random field can be characterised by it's law, which is a Gaussian measure specified through a mean functional (taken to be zero) and [suppressing technicalities!] its Cameron-Martin space $(H, \langle \cdot, \cdot \rangle_H)$, which is a Hilbert space with a *given* inner product.

- The inner product on H can be specified in terms of an operator $\mathcal{Q}$ (i.e. $\langle x, y \rangle_H = \langle x, \mathcal{Q}x \rangle$). This is the precision operator and is the "inverse" of the covariance operator.
- Lots of things simplify, especially for hierarchical models!
- Furthermore $\mathcal{Q}$ is is (pseudo-)differential operator, which we can approximate efficiently through standard wavelet (or Galerkin) methods.

"Being Joan Crawford at 17 was easy..."

Lovely definition, Daniel, but what if I want to actually do something?

- Parameter estimation: Requires the field at N data points, $\mathbf{\Sigma} \in \mathbb{R}^{N \times N}$.
- Spatial prediction (Kriging): Requires the field at m points, probably densely through the domain, $\mathbf{\Sigma} \in \mathbb{R}^{m \times m}$.
- Joint parameter and spatial estimation: Needs it at both, $\mathbf{\Sigma} \in \mathbb{R}^{(N+m) \times (N+m)}$.

Clearly, if N or m is large, this will be *very* expensive. This is the classical problem of spatial statistics.

Det kommer bara leda till nåt ont..

Most of the methods aimed at reducing the "big N problem" in spatial statistics is based on some sort of low-dimensional approximation:

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

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

There are *lots* of details!

The village green preservation society

With our model in hand, we now need to perform inference.

Direct methods

All Bayesian inference methods (and most Frequentist ones) require a factorisation of the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{R}\boldsymbol{R}^T$ or the precision matrix $\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1} = \boldsymbol{L}\boldsymbol{L}^{\boldsymbol{T}}$.

- Making these factorisations computationally feasible is the main aim of modern spatial statistics.
- This was the (computational) state of the art 10 years ago and, with some minor blips, it is still the state of the art.

Outline

"Oh those Russians!"—Indirect methods

For spatial problems in applied mathematics, physics, and engineering, direct methods are typically overlooked in favour of *iterative methods*.

- The are a huge variety of *Krylov subspace* methods, the most famous being the Conjugate Gradient method (also GMRES, BiCG-Stab, LSQR,...)
- These *do not* require the matrix, but rather access to matrix-vector products of the form $\boldsymbol{Qx}$.
- Typically, these methods are exact if you run them for long enough, and they converge superlinearly in subspace size.

Ignition

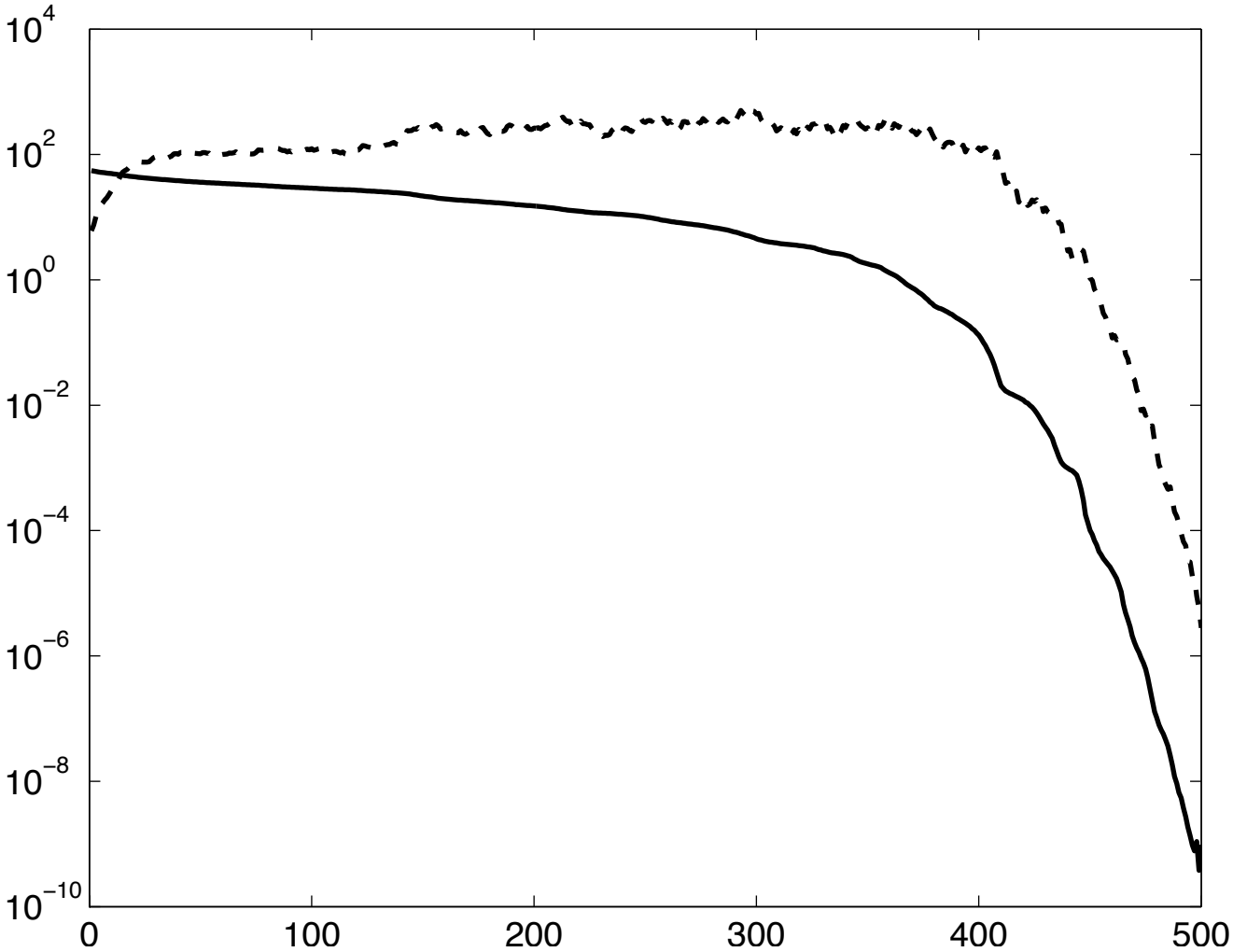
Sampling from a Gaussian

If we have a factorisation of the precision matrix $Q = LL^T$, then it's easy to see that $x = L^{-T}z$, z i.i.d. standard normal, is a sample from $N(0, Q^{-1})$.

- We need a version of $L^{-T}z$ that we can compute using only matrix-vector products from Q.
- It turns out, that we can use Krylov subspace methods to computer $Q^{-1/2}z$.
- Convergence is analogous to the conjugate gradient methods for computing $Q^{-1}z$

Outline

No easy way down



Stand by your manatee

What went wrong?

- The rate of convergence depends on the condition number of $\boldsymbol{Q}$, which is $\mathcal{O}(h^4)$.
- This is a standard problem with Krylov subspace methods.
- When solving linear systems, the solution is to *precondition* the system, i.e. find $\boldsymbol{F}\boldsymbol{F}^T \approx \boldsymbol{Q}$ and solve $\boldsymbol{F}^{-1}\boldsymbol{Q}\boldsymbol{F}^{-T}y = \boldsymbol{F}^{-1}\boldsymbol{z}$.
- A preconditioner is *optimal* if the condition number remains $\mathcal{O}(1)$ as $h \to 0$.

Can we precondition the sampling routine?

Ignition (Remix)

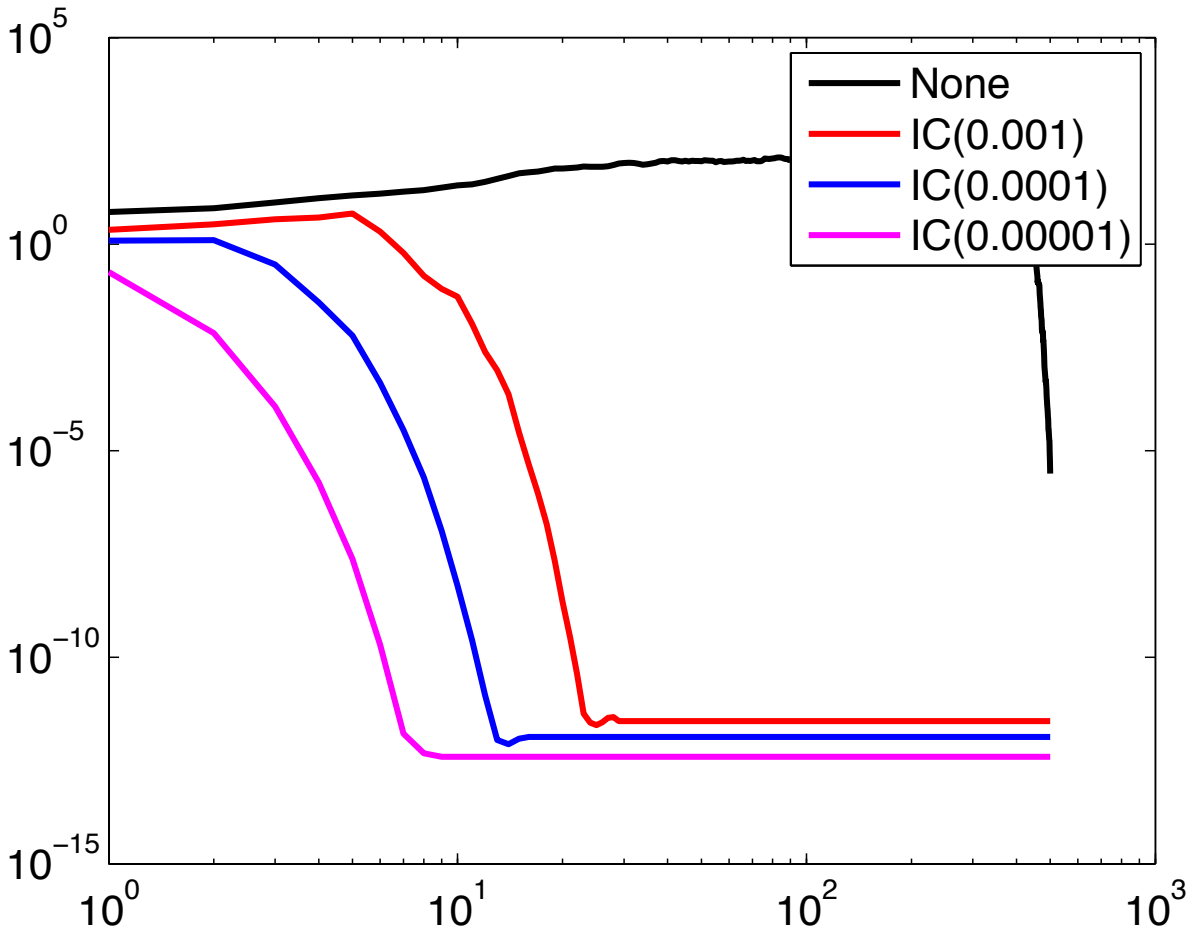
Rather than looking for a transformation that preserves the solution to the linear system, we look for one that gives the same *distribution*.

Preconditioned Sampling

Let $\boldsymbol{Q}$ and $\boldsymbol{M} = \boldsymbol{F}\boldsymbol{F}^T$ be symmetric positive definite matrices. If $\boldsymbol{y} \sim N\left(\boldsymbol{0}, \left(\boldsymbol{F}^{-1}\boldsymbol{Q}\boldsymbol{F}^{-T}\right)^{-1}\right)$, then the solution to $\boldsymbol{F}^T\boldsymbol{x} = \boldsymbol{y}$ is a zero-mean Gaussian random vector with precision matrix $\boldsymbol{Q}$.

- This replaces the problem of sampling from $N(\boldsymbol{0}, \boldsymbol{Q}^{-1})$ with sampling from $\sim N\left(\boldsymbol{0}, \left(\boldsymbol{F}^{-1}\boldsymbol{Q}\boldsymbol{F}^{-T}\right)^{-1}\right)$, which should be better behaved.
- Generic choice of $\boldsymbol{F}$ is the incomplete Cholesky factorisation of $\boldsymbol{Q}$.

Speed Lab


10^5
10^0
10^{-5}
10^{-10}
10^{-15}
10^0
10^1
10^2
10^3
None
IC(0.001)
IC(0.0001)
IC(0.00001)

Breaking Glass

For stationary Gaussian random fields on a regular lattice (on a torus), the precision matrix (and the covariance matrix) is *circulant* and all of the calculations can be done in $\mathcal{O}(n \log n)$ operations using FFTs.

- Any operation involving *data* destroys the circulant structure, leading to precision matrices of the form $\boldsymbol{Q}_{post} = \boldsymbol{Q}_{prior} + \boldsymbol{\Lambda}$.
- This means that good MCMC methods that take into account the second order properties of the likelihood *cannot* be used!
- With Krylov Sampling, you can still do everything. Matrix-vector products with $\boldsymbol{Q}_{post}$ still cost $\mathcal{O}(n \log n)$ and the number required to reach a prescribed accuracy is $\mathcal{O}(1)$ as $n \to \infty$.

All the best

Equivalent operator preconditioning

if $\left\|\mathcal{Q}^{1/2}x\right\|_H$ and $\left\|\mathcal{M}^{1/2}x\right\|_H$ are equivalent norms on $\mathcal{R}_H(Q^{-1/2})$ and $\boldsymbol{Q}_n$ and $\boldsymbol{M}_n$ are compatible discretisations of $\mathcal{Q}$ and $\mathcal{M}$, then

$$c\langle \boldsymbol{x_n}, \boldsymbol{M_n x_n}\rangle \leq \langle \boldsymbol{x_n}, \boldsymbol{Q_n x_n}\rangle \leq C\langle \boldsymbol{x_n}, \boldsymbol{M_n x_n}\rangle$$

and the Krylov sampler required $\mathcal{O}(1)$ iterations to reach the prescribed accuracy ϵ.

- It is not all that difficult to show that if $\mathcal{Q}_{prior}^{-1}\Lambda$ is nice (i.e. is Hilbert-Schmidt), then $\boldsymbol{M} = \boldsymbol{Q}_{prior}$ is an optimal preconditioner for $\boldsymbol{Q}_{post}$.

Simon Smith and his amazing dancing bear

$\boldsymbol{Q}$ is generated using the exponential covariance function on a torus and the diagonals of Λ are $U[0, 10]$.

n	$\boldsymbol{L} = \boldsymbol{Q}^{1/2}$	$\boldsymbol{L} = (\boldsymbol{Q} + \bar{\boldsymbol{\Lambda}})^{1/2}$
32	5	3
64	6	3
128	5	3
256	6	3
512	6	3
1024	6	3

Outline

Less talk, more rock

A classic latent Gaussian process in which the latent field is almost always modelled as block circulant (or block Toeplitz) is the log-Gaussian Cox process model for point pattern data.

$$
\begin{aligned}
y_i|\boldsymbol{\eta} &\sim Po(h^2 e^{\eta_i}) \\
\boldsymbol{\eta}|\boldsymbol{\theta} &\sim N(\mathbf{0}, \boldsymbol{Q}(\boldsymbol{\theta})^{-1}) \\
\boldsymbol{\theta} &\sim \pi(\boldsymbol{\theta}).
\end{aligned}
$$

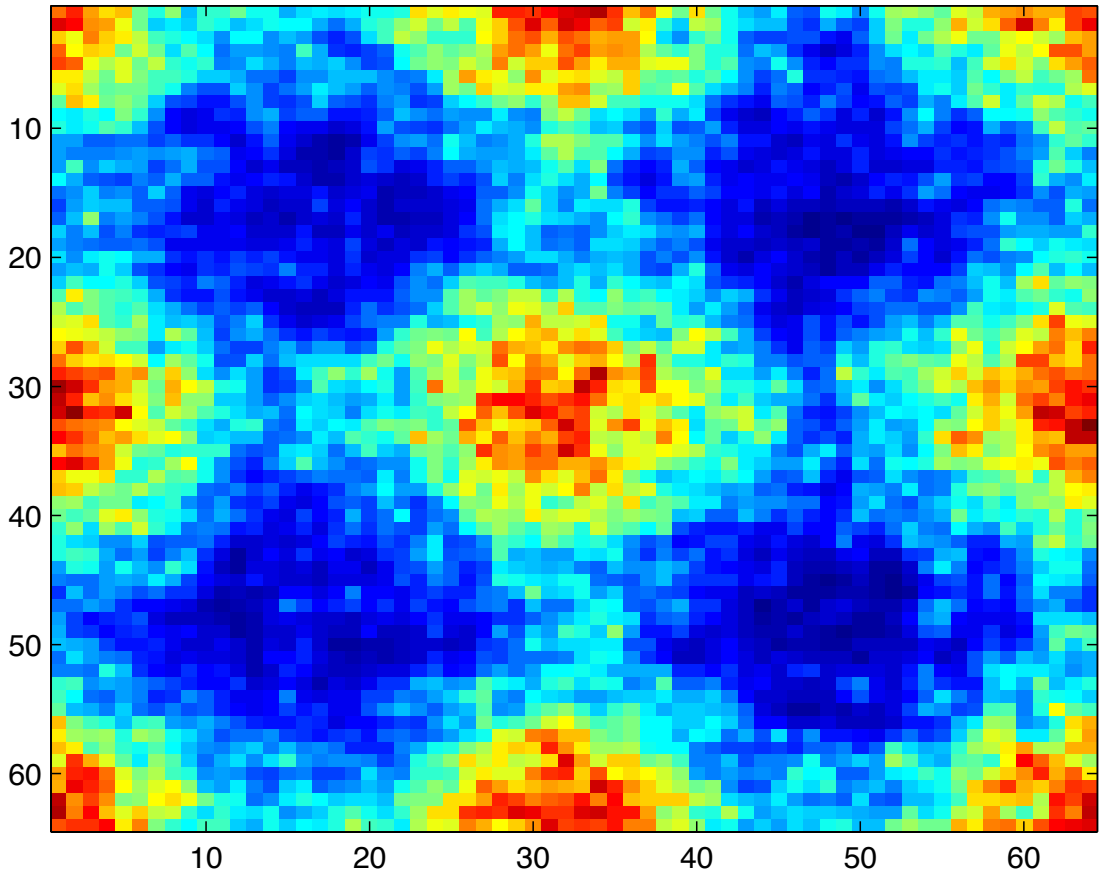
- Here $\boldsymbol{Q}$ is a circulant matrix that has possibly been extended to include fixed effects.
- The preconditioner for in the case of fixed effects is the same! (block diagonal with the circulant preconditioner and maybe a scaling for the fixed effects components).

"Further Complications."

There is one problem: computing the acceptance ratio (or the Hamiltonian) requires the computation of the determinant.

- This is trivial if you're using a direct method!
- For indirect methods, $\text{logdet}(\boldsymbol{A}) = \mathbb{E}(\boldsymbol{u}^T \log(\boldsymbol{A})\boldsymbol{u})$ can be computed (and preconditioned) in the same way as the samples.
- NB: $\text{logdet}(\boldsymbol{Q}) = \text{logdet}(\boldsymbol{F}^{-1}\boldsymbol{Q}\boldsymbol{L}^{-T}) - 2\,\text{logdet}(\boldsymbol{F})$
- Unbiased MC estimate for the log-determinant can be converted to an unbiased estimate for the determinant using results from Physics. The resulting MCMC routine still asymptotically correct, but the cost is a higher variance.

Are you the one I've been waiting for?



Outline

I could never take the place of your man

Hopefully, I have convinced you that there are a suite of iterative methods that can be used as efficient replacements for traditional methods.

That being said, these are still methods for *HARD* problems—if existing methods are satisfactory, there is no reason to change!

Now—Later—Soon

- Seeing as we can do everything as long as we have matrix-vector products, we should focus expanding our work to these models! When life is non-stationary, or unequally spaced, we should really spend some quality time with wavelet.
- We probably need more work on MCMC theory. There are results for IEEE floating point arithmetic, but in this case the cut-off function is not the same for each iteration. Is the chain still ergodic? And what does it converge to? How far are we from correct?
- We can actually do all of the things required for INLA. With some programming effort, we may get some interesting results for *HUGE* models!

Playlist: `http://spoti.fi/KubkvQ`