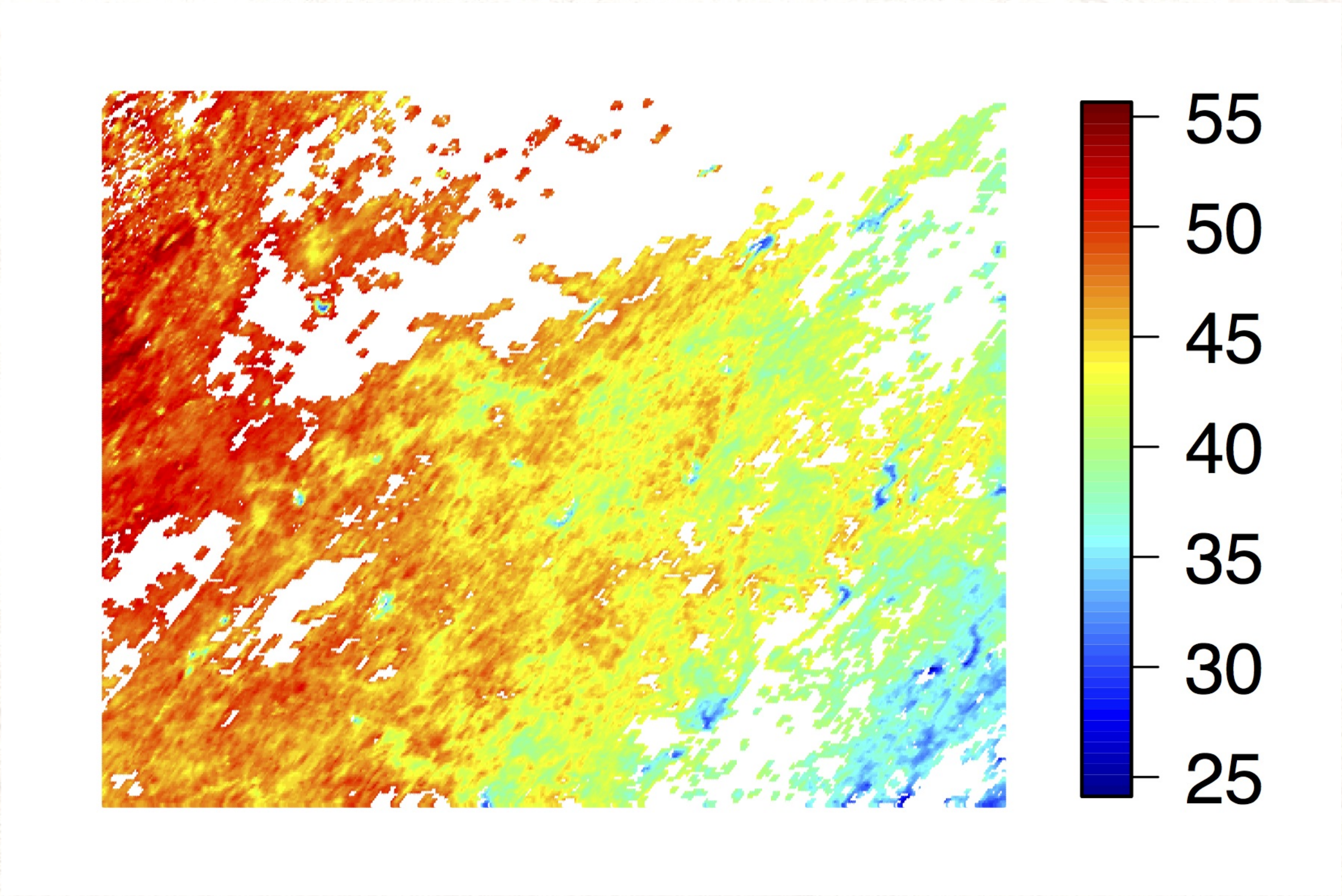
Statistical models for large spatial datasets

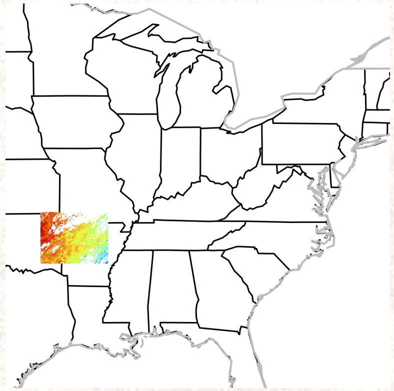
Douglas Nychka,
Colorado School of Mines

Introduction

- Examples of large spatial data and the problems
- Some cartoons and a spatial model
- Multi-resolution model
- LatticeKrig – properties
- LatticeKrig in action.

Remotely sensed temperatures





About 100K observations

Problems with large spatial data sets

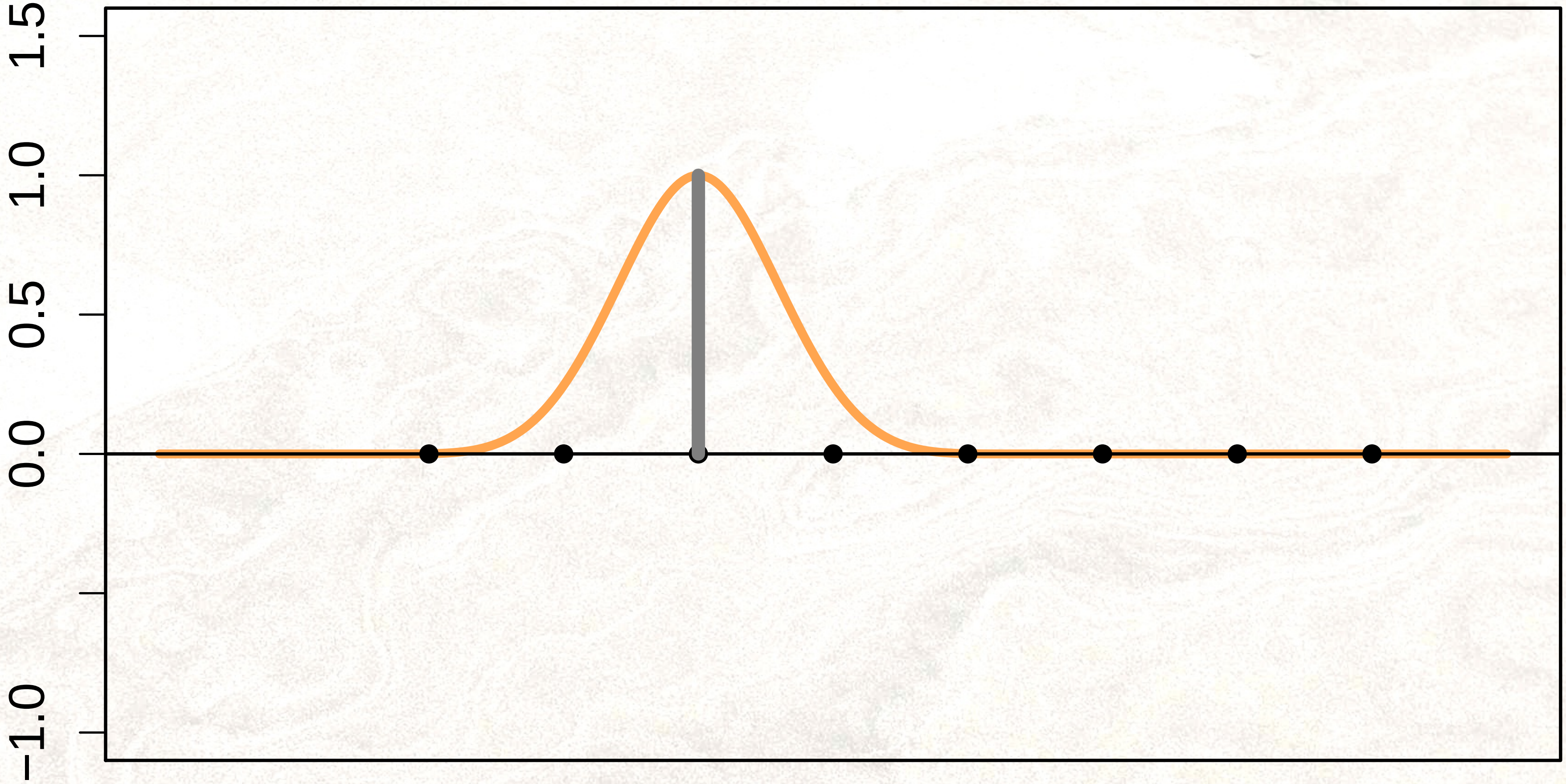
- Storage: Covariance matrices are large – size of the number of observations

- Computation: Linear algebra for the usual Kriging estimator grows as the cube of the number of observations.

- Inference: Exact prediction standard errors are not computationally feaisible.

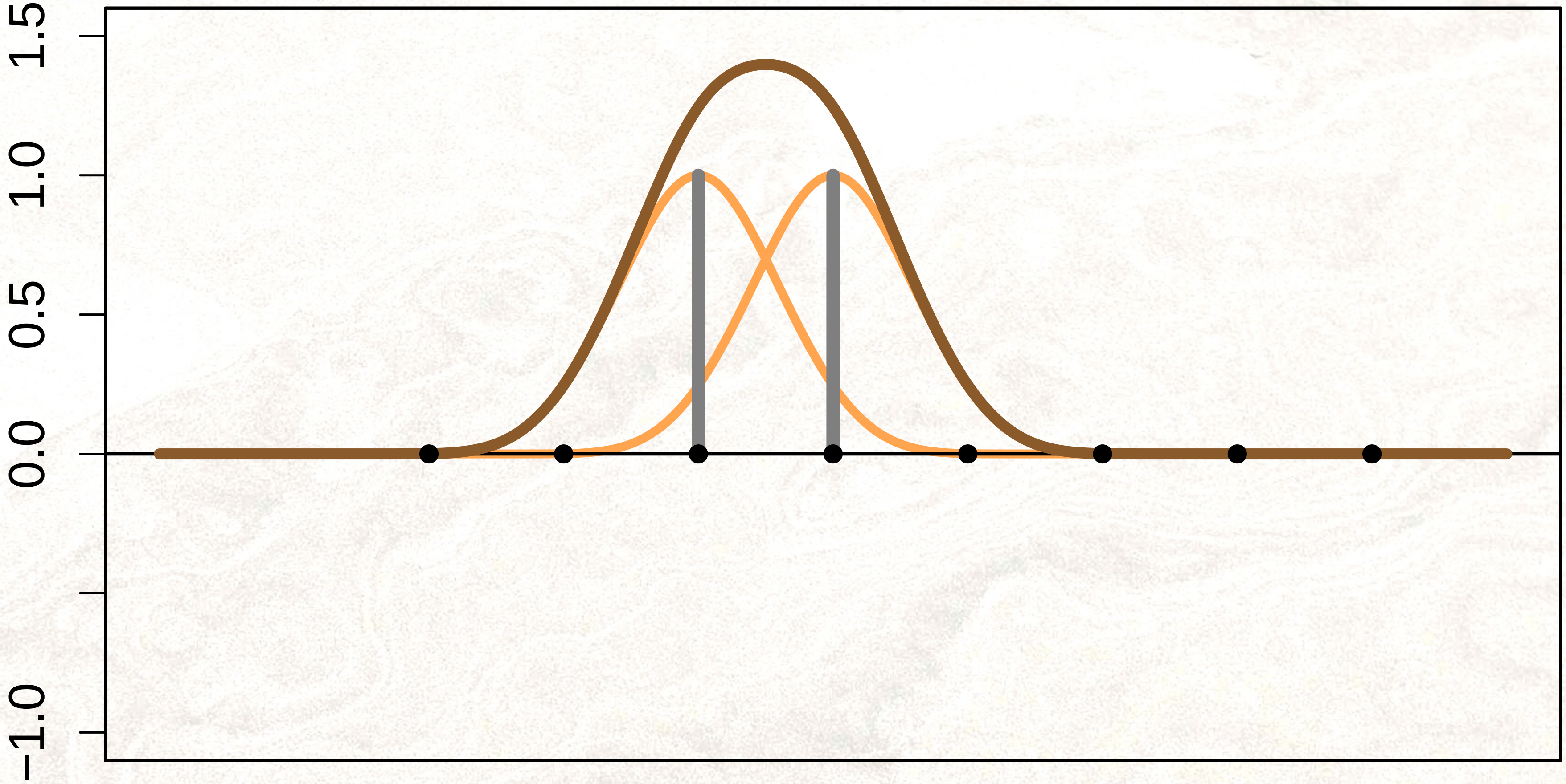
Cartoons

Building a curve from bumps



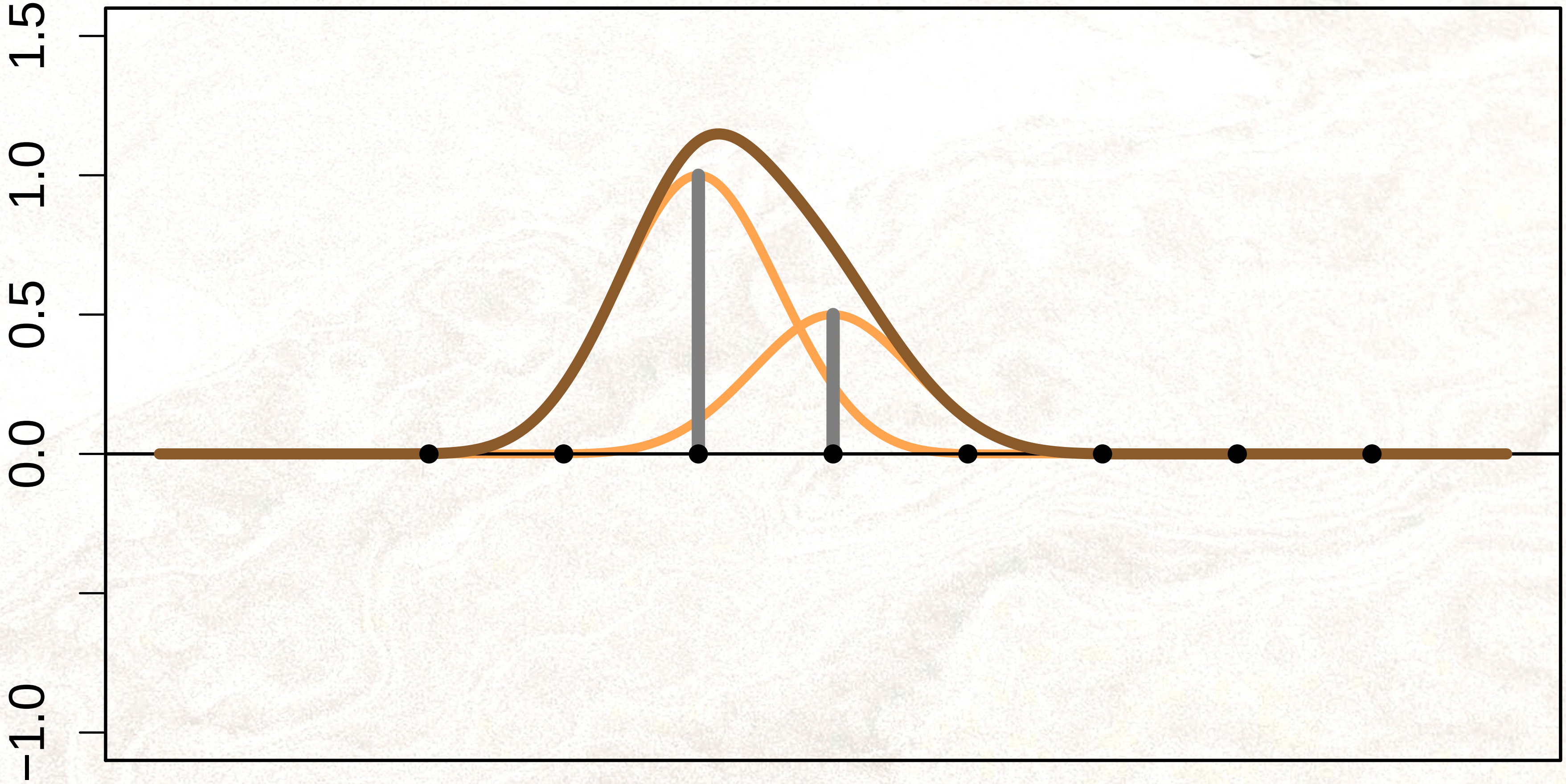
Single bump

Building a curve from bumps



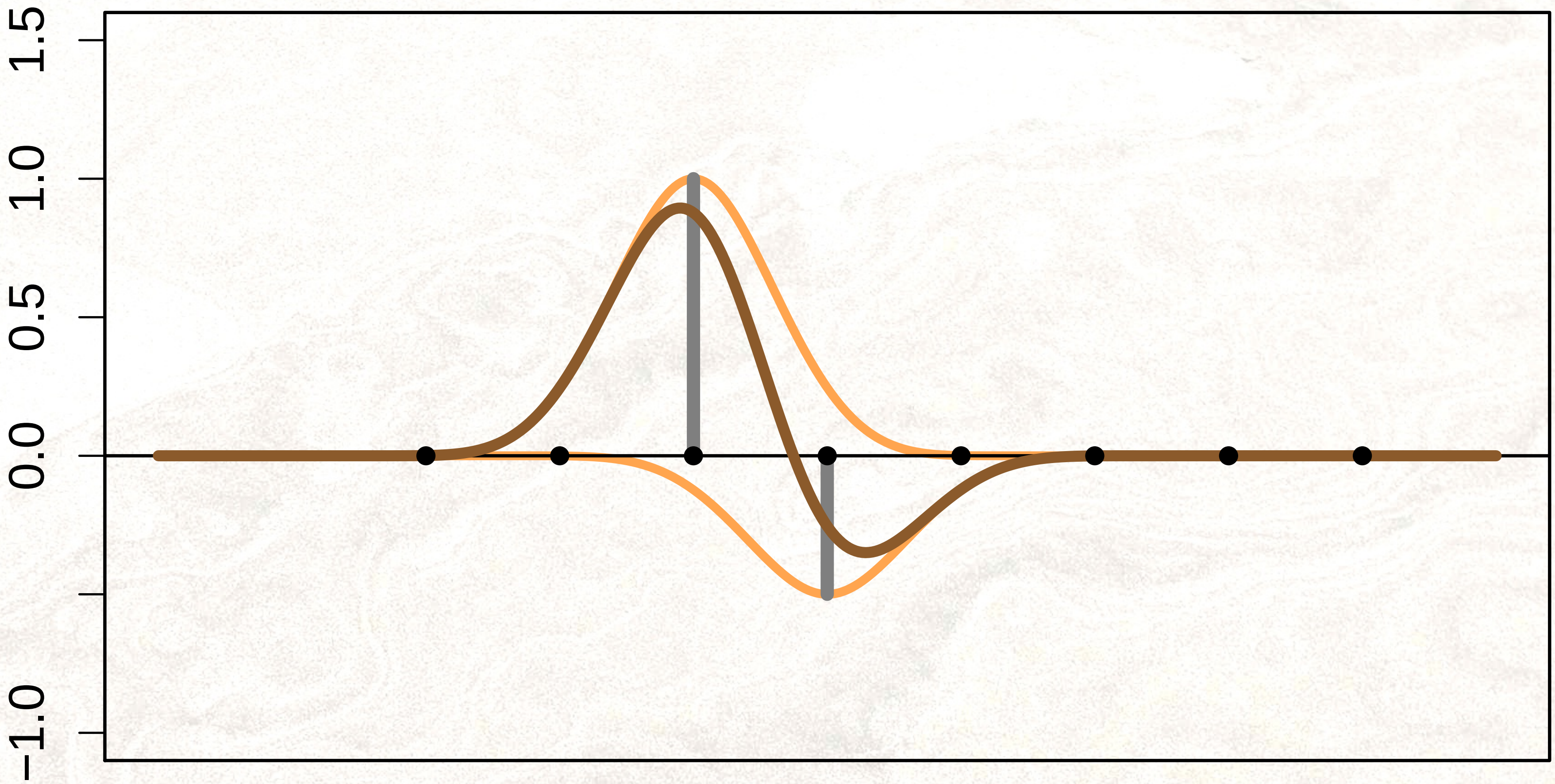
Two bumps same height

Building a curve from bumps



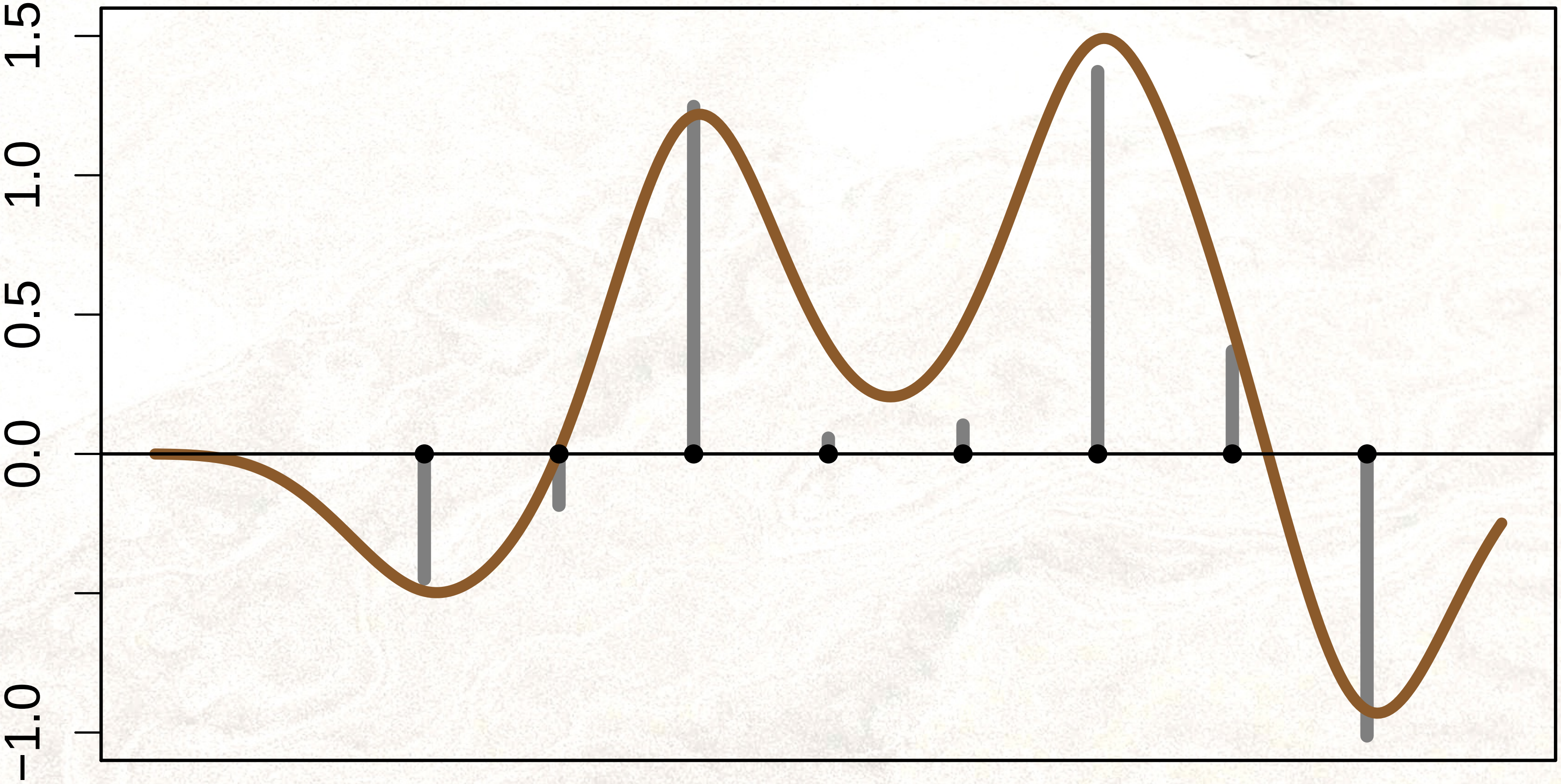
Two bumps different heights

Building a curve from bumps



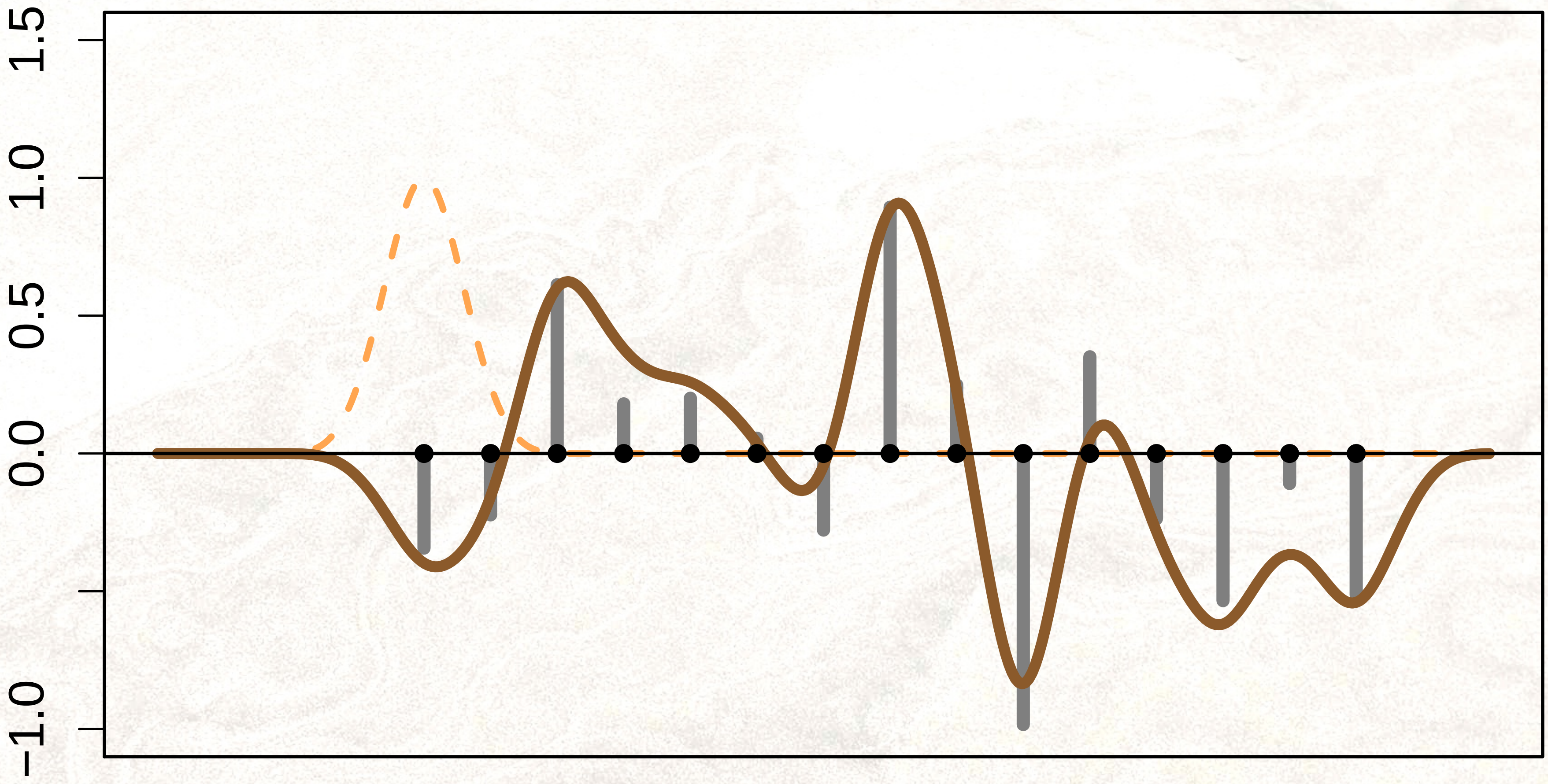
Two bumps different heights

Building a curve from bumps



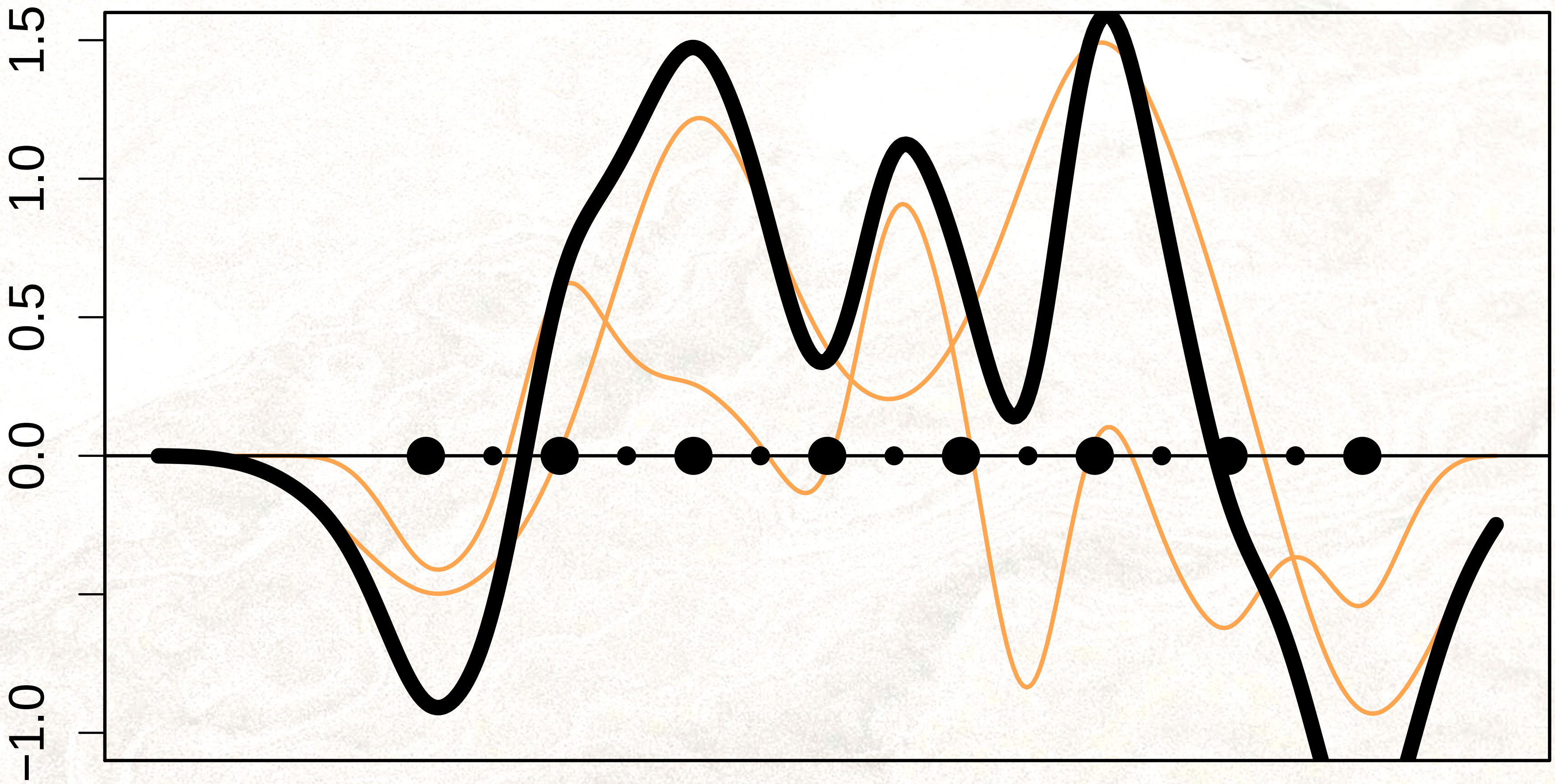
Eight bumps – all different heights

Building a curve from bumps



16 bumps – all different heights

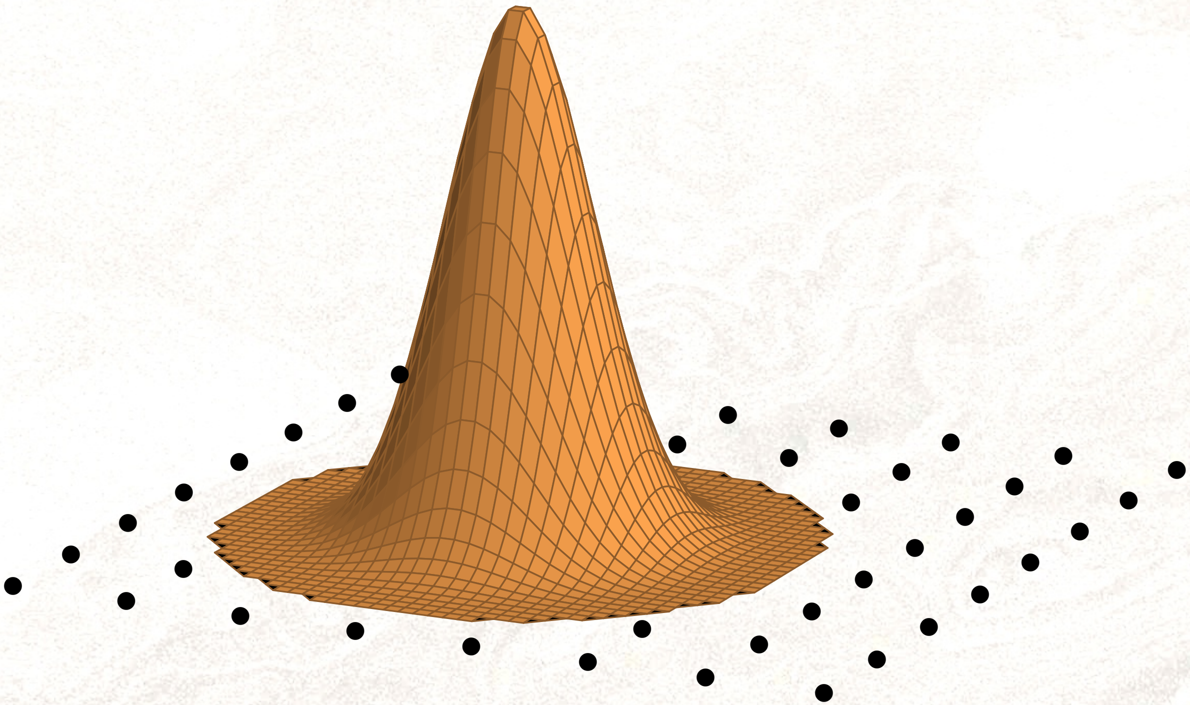
Building a curve from bumps



Adding them together

bumps = basis functions, bump heights = coefficients

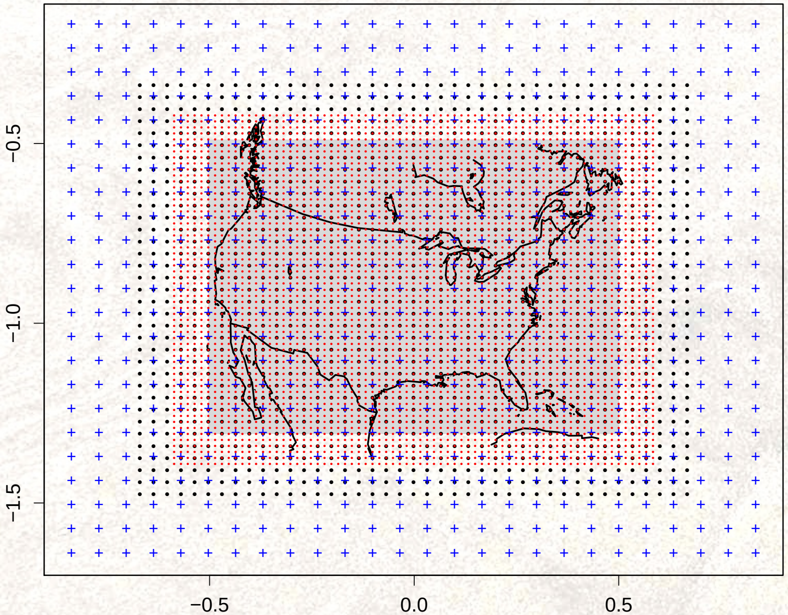
A (Wendland Basis function

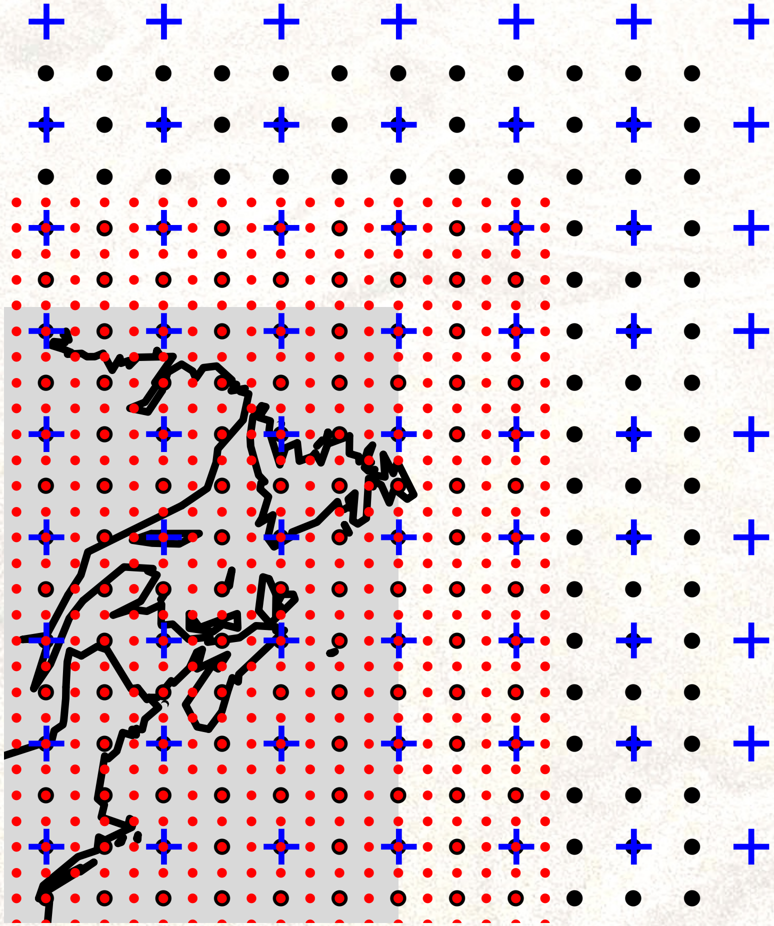


Example of a 2-d bump

A lattice example

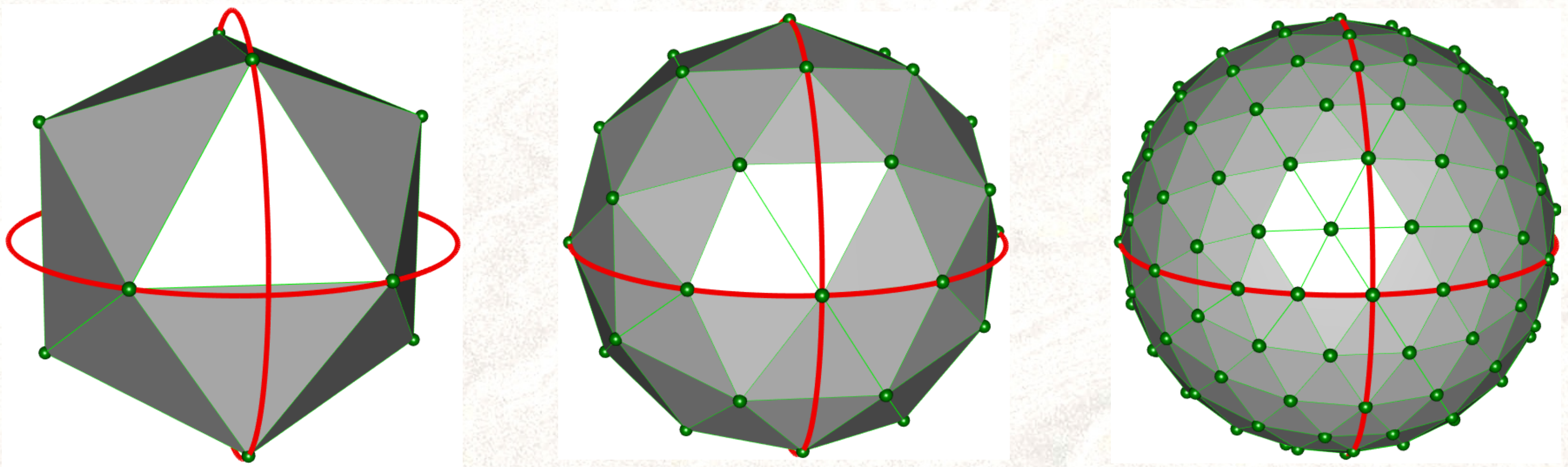
- Three levels
- Extra points on margins to minimize edges effects
- About 4000 total lattice points





Another lattice

Icosohedra grids for the sphere.



Spatial model

A linear (random effects) model

- X a regression matrix with $X_{i,j} = \phi_j(\boldsymbol{x}_i)$

Observations:

$$\boldsymbol{y} = X\boldsymbol{c} + \boldsymbol{e} \quad \boldsymbol{e} \sim MN(0, \tau^2 I)$$

Process:

$$g(x) = \sum_j \phi_j(x) c_j, \quad \boldsymbol{c} \sim MN(0, \sigma^2 Q^{-1})$$

Potential Priors:

$$[\sigma^2, \tau^2, Q]$$

Derived Covariance

$$Cov(g(\boldsymbol{x}), g(\boldsymbol{x}')) = \sigma^2 \sum_{j,k} \phi_j(\boldsymbol{x}) \left[Q^{-1}\right]_{j,k} \phi_k(\boldsymbol{x}')$$

- The model is written so that the covariance never needs to be explicitly found.

Computing the estimate

Integrating out $\boldsymbol{c}$:
$[\boldsymbol{y}|\sigma^2, \tau^2, Q] \sim MN(0, (\sigma^2 \boldsymbol{X} Q^{-1} \boldsymbol{X}^T + \tau^2 I)$

Likelihood/posterior computation for σ^2, τ^2, Q

dominated by

$$|\sigma^2 \boldsymbol{X}^T Q^{-1} \boldsymbol{X} + \tau^2 I| \text{ or equivalently } |(\tau^2 \boldsymbol{X}^T \boldsymbol{X} + (1/\sigma^2) Q|$$

Kriging estimate of $\boldsymbol{c}$:

$$\hat{c} = (\boldsymbol{X}^T \boldsymbol{X} + (\tau^2/\sigma^2) Q)^{-1} \boldsymbol{X}^T \boldsymbol{y}$$

Conditional simulation

Based on *unconditional* simulation of $\boldsymbol{c}$ and Kriging estimate.

- Fast computation hinges on sparsity of Q and X.

Details and engineering

More about Q

At a single level

Some coefficients:

$$\begin{array}{ccccc} . & . & . & . & . \\ . & . & c_1 & . & . \\ . & c_2 & c_* & c_3 & . \\ . & . & c_4 & . & . \\ . & . & . & . & . \end{array}$$

Some weights:

$$\begin{array}{ccccc} . & . & . & . & . \\ . & . & \text{-1} & . & . \\ . & \text{-1} & a & \text{-1} & . \\ . & . & \text{-1} & . & . \\ . & . & . & . & . \end{array}$$

The filter:

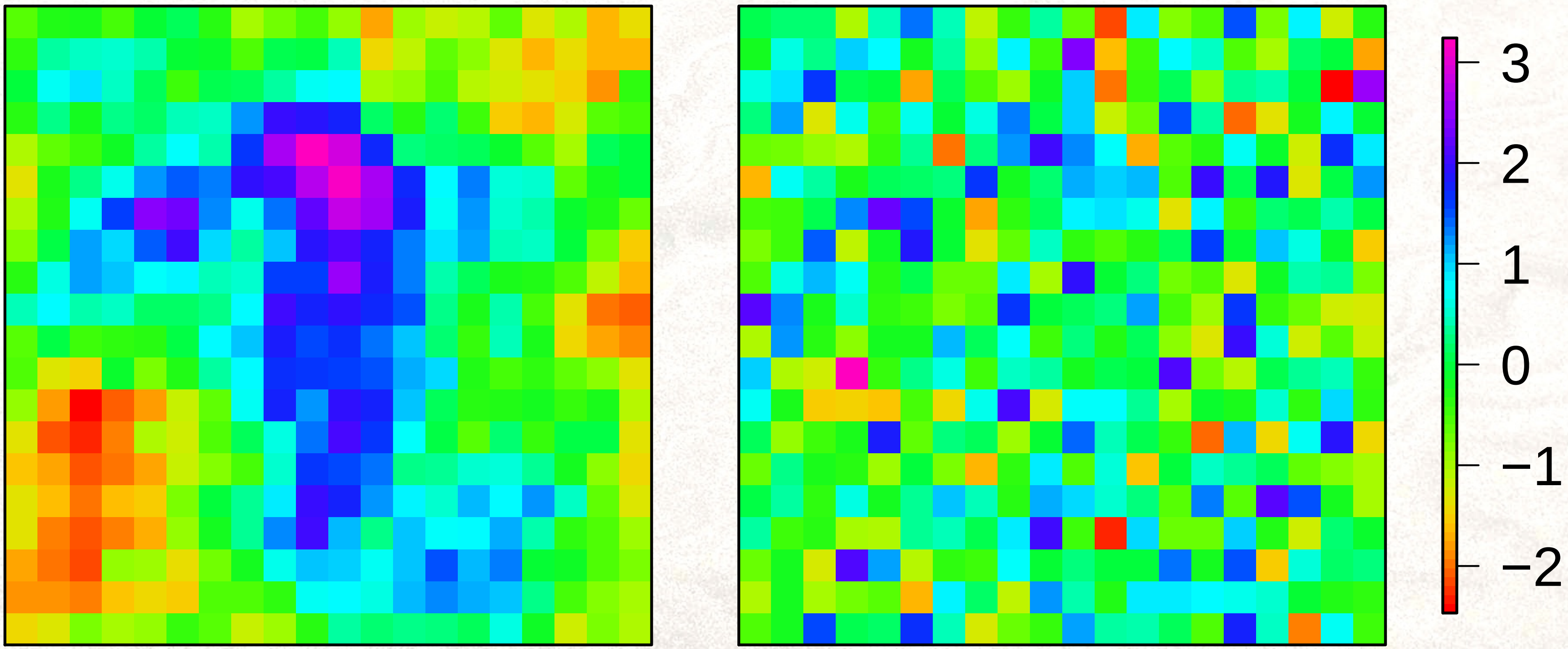
$ac_* - (c_1 + c_2 + c_3 + c_4) =$ white noise

If $B\boldsymbol{c} =$ iid $N(0,1)$, $\qquad Q = BB^T$

- a needs to be greater than 4.
- A simple discretization of the Laplacian. $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$

Filtering coefficients

Coefficients on the lattice *Applying the filter*

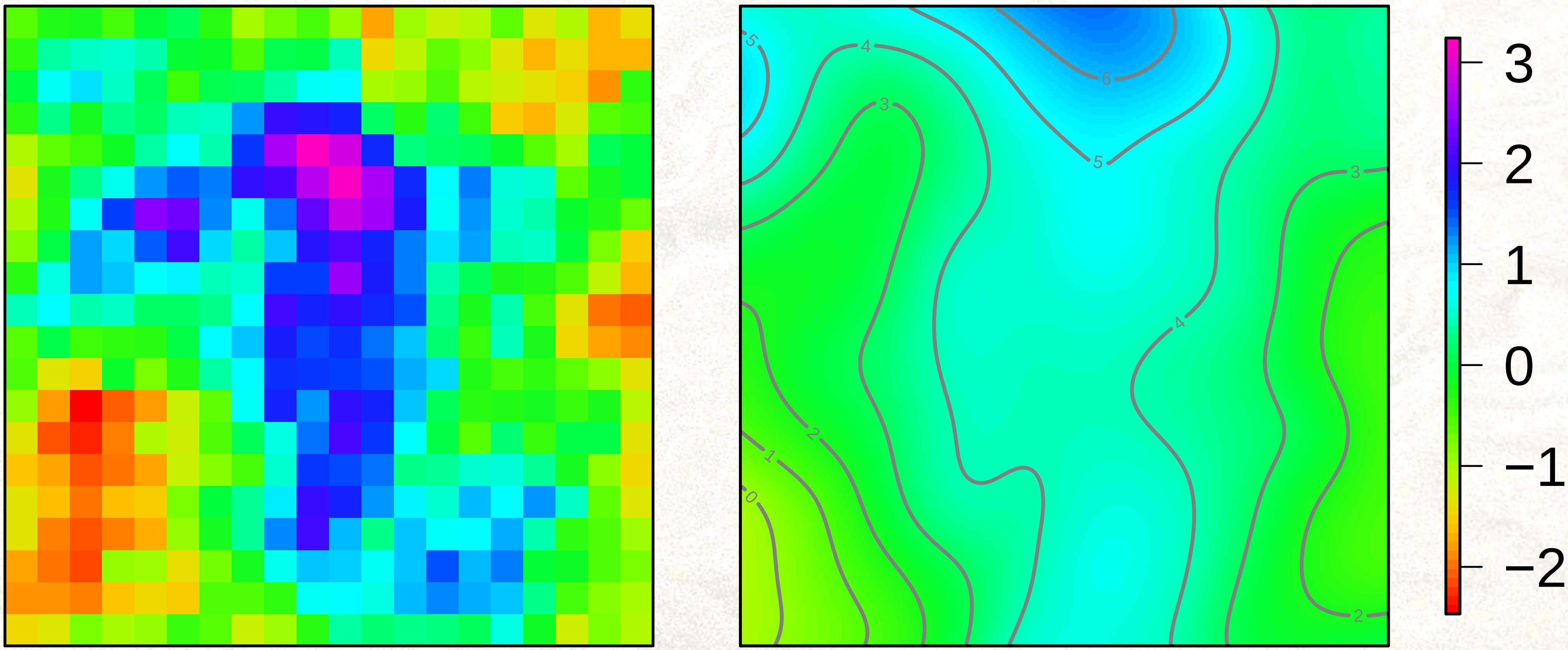


$$c_* \rightarrow a c_* - (c_1 + c_2 + c_3 + c_4)$$

$a = 4.01$

Applying the basis functions

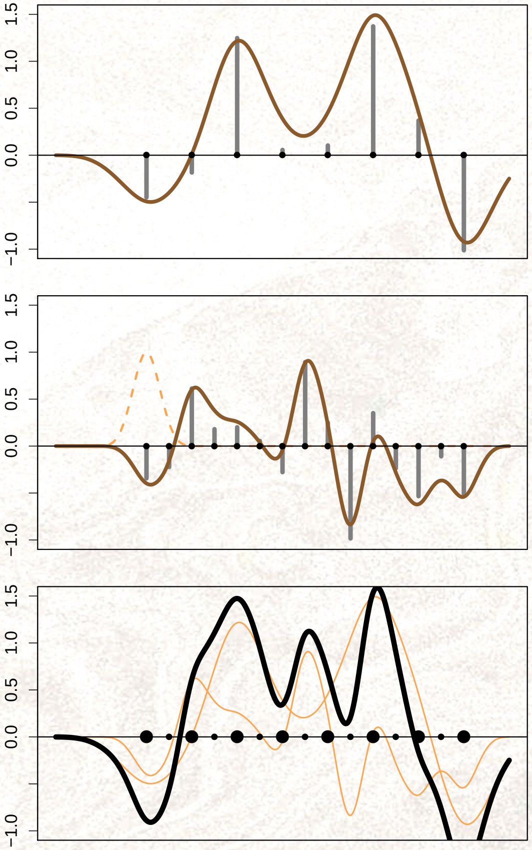
Coefficients on the lattice *Expanding with basis functions*



$$c_k \to \sum \phi_k(x) c_k = g(x)$$

More than one level:

Adding different resolutions together:



$$g(\boldsymbol{x}) = \sigma^2(\alpha_1 g_1(\boldsymbol{x}) + \alpha_2 g_2(\boldsymbol{x}) + \alpha_3 g_3(\boldsymbol{x}) + \ldots)$$

$$Q = (1/\sigma^2)\begin{bmatrix} \alpha_1 B_1^T B_1 & 0 & 0 \\ 0 & \alpha_2 B_2^T B_2 & 0 \\ 0 & 0 & \alpha_3 B_3^T B_3 \end{bmatrix}$$

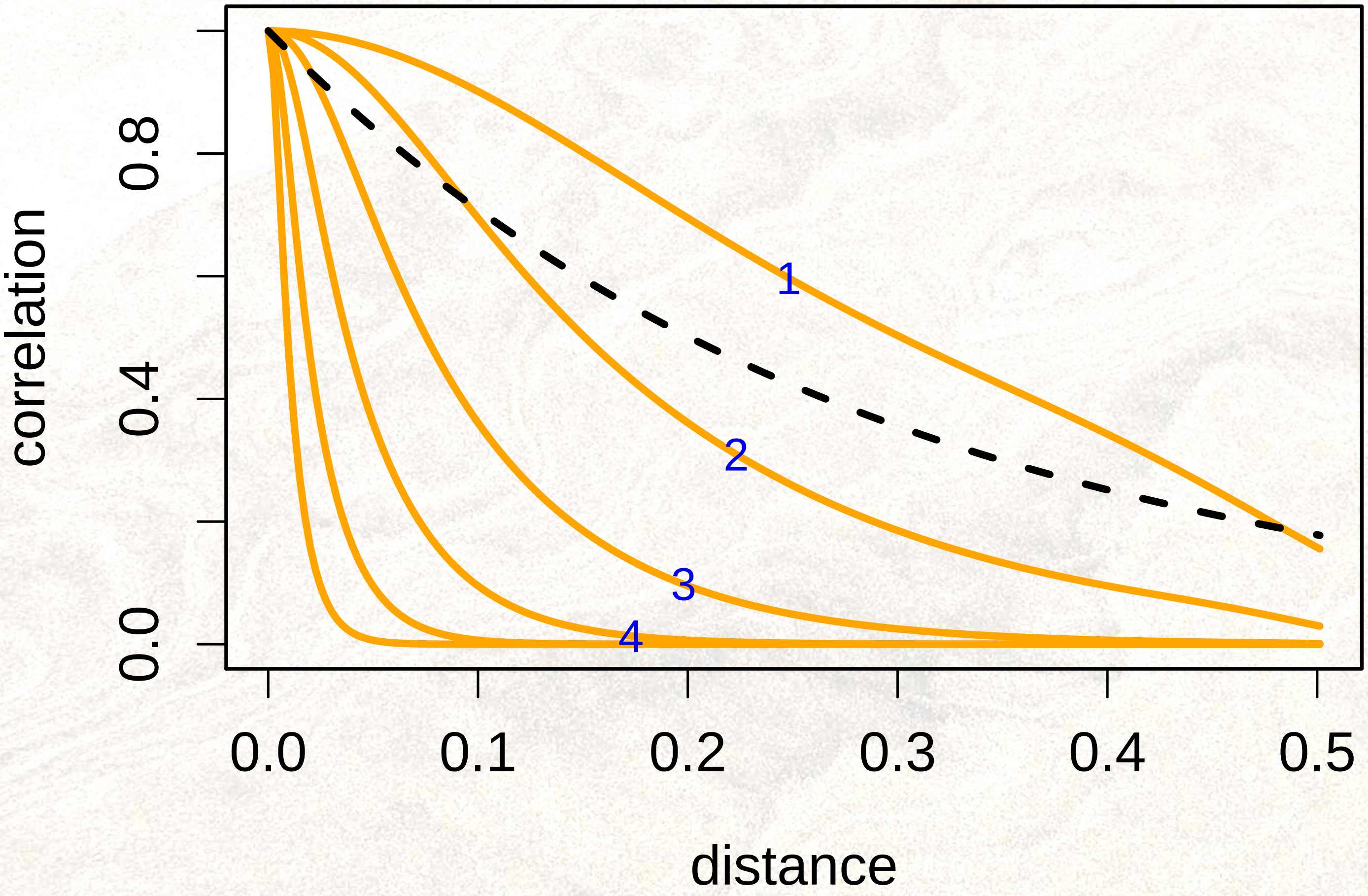
- σ^2 marginal variance of the process
- $\alpha_1, \alpha_2, \alpha_3$ relative weight for each level –
  all nonnegative and add to 1.

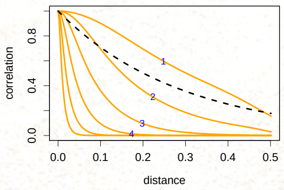
Benefits of a multi-resolution

Approximating standard covariances

Approximating an exponential covariance

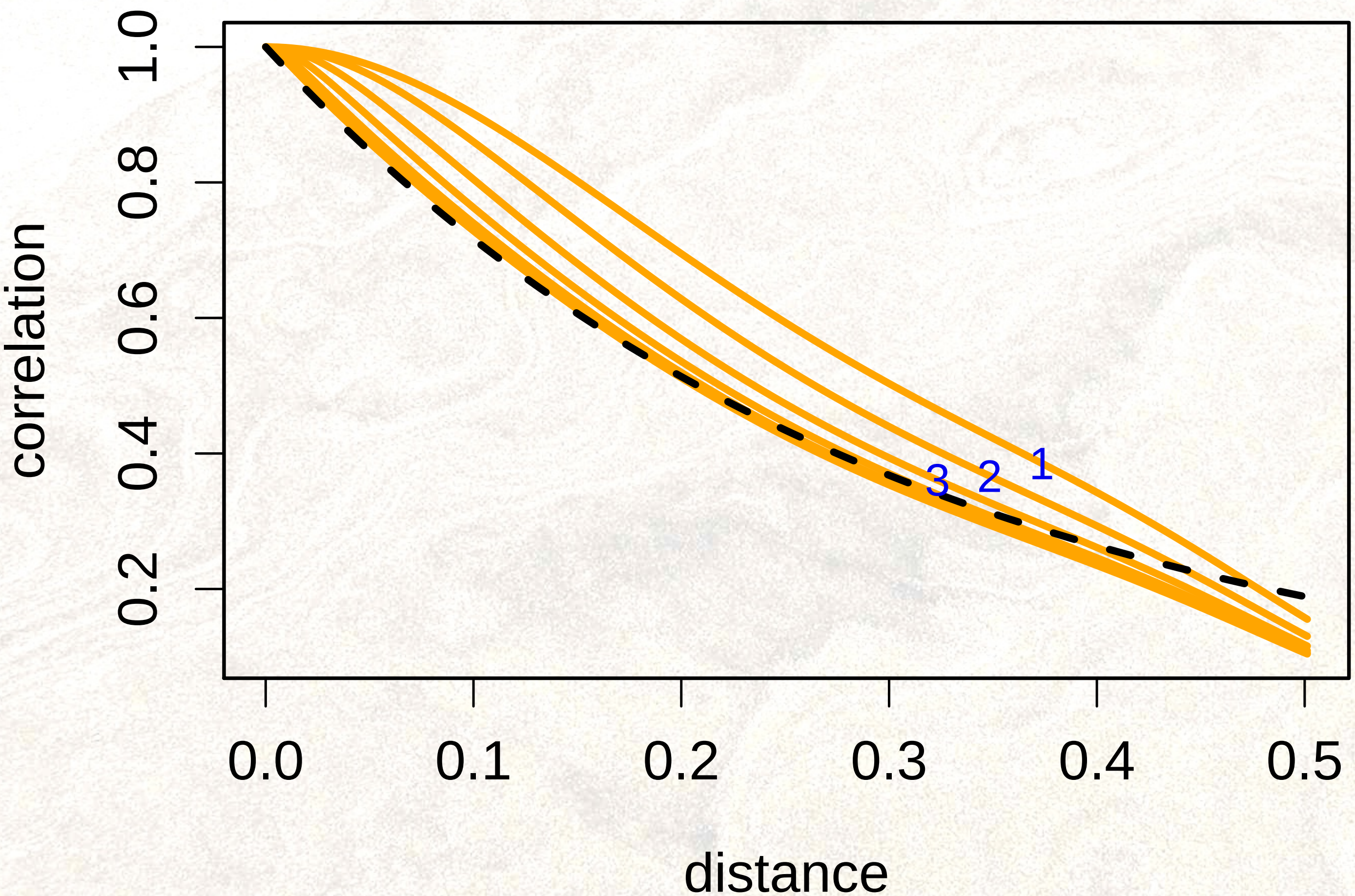
Correlation functions for 6 levels and a target exponential





Weighting by $2^{-level/2}$

Correlation functions adding levels and the target exponential



Timing

On my mac laptop and in R

— i.e. a single core and `LatticeKrig` pacakge

Computation may be dominated by :

- matrix setup
- normalization to stationarity
- *Cholesky decomposition*

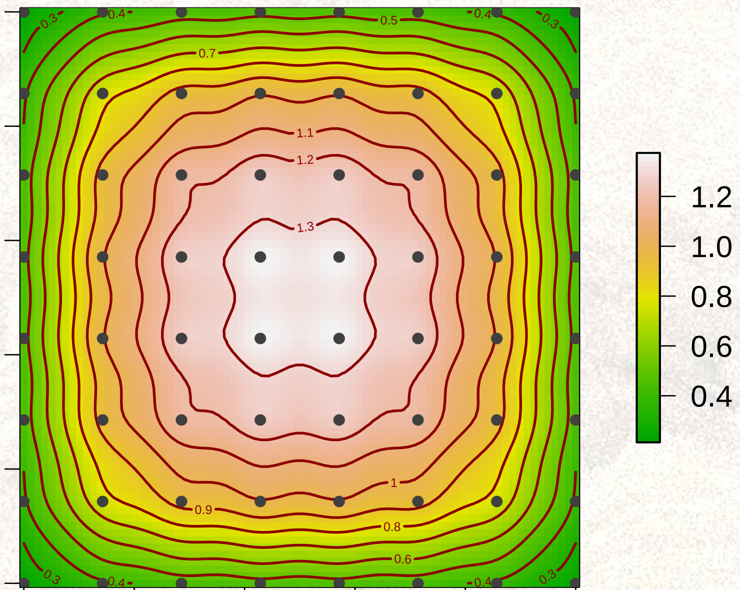
For 20,000 observations, single likelihood evaluation:

- standard Kriging (dense Cholesky) is $\approx$ 20 minutes
- `LatticeKrig` (sparse Cholesky) is $\approx$ 10 seconds.

Stationarity?

Recall $Cov(g(\boldsymbol{x}), g(\boldsymbol{x}')) = \sigma^2 \sum_{j,k} \phi_j(\boldsymbol{x}) \left[Q^{-1}\right]_{j,k} \phi_k(\boldsymbol{x}')$

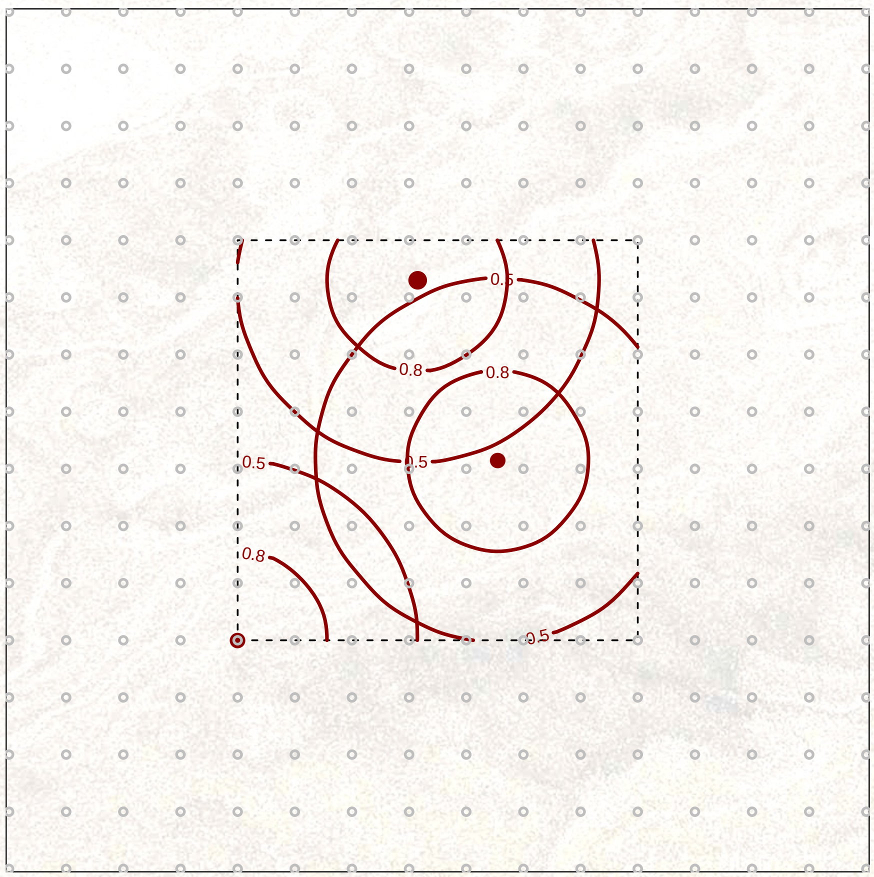
Correlation function for a single level $a = 4.2$
Marginal variance $VAR(g(x)$ from a 8$\times$ 8 grid



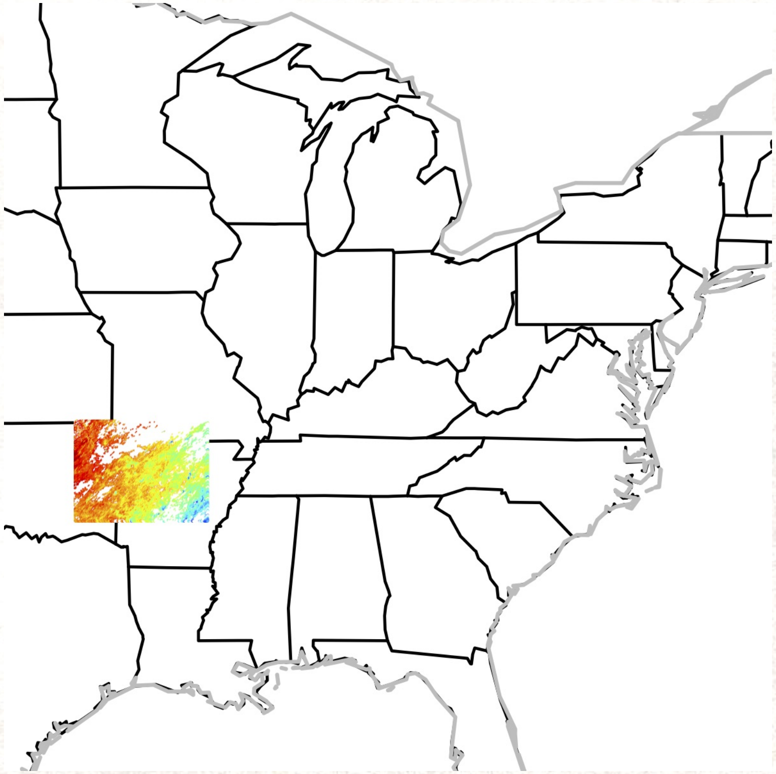
Adding buffer and normalizing

$8\times$ 8 grid with 4 grid points of buffer and normalized

Correlation function for a single level, $a = 4.2$

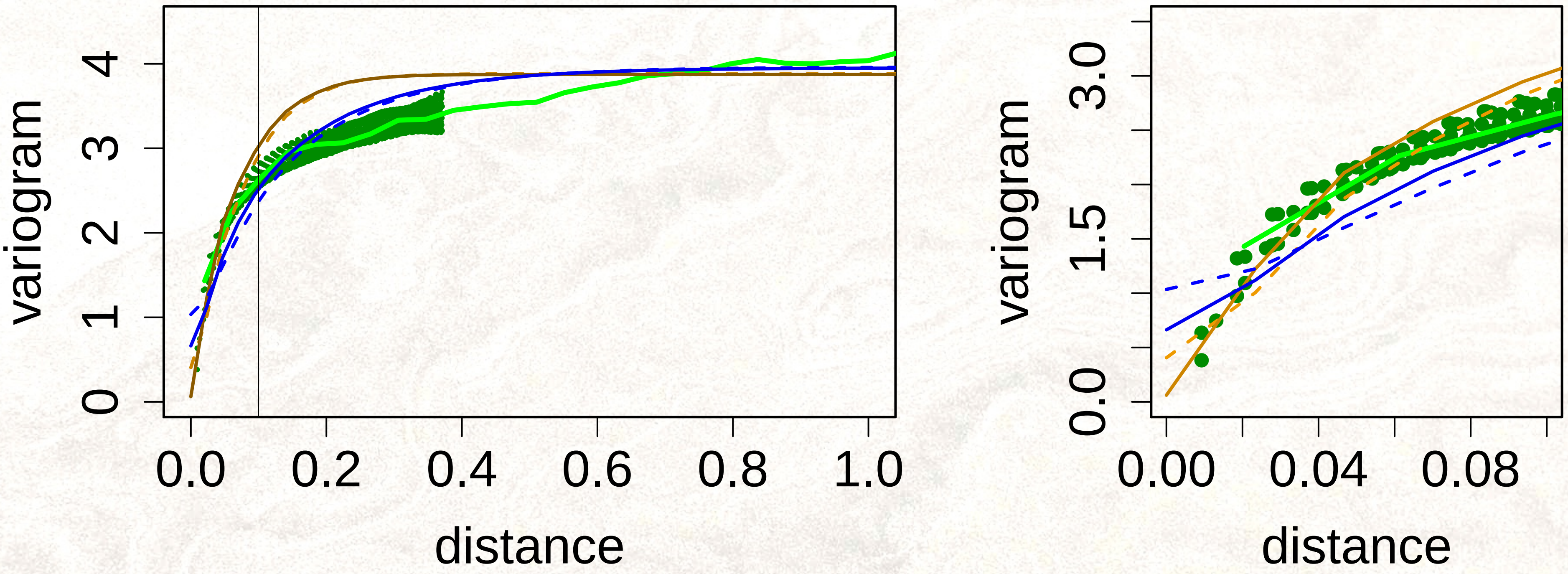


Surface temperatures



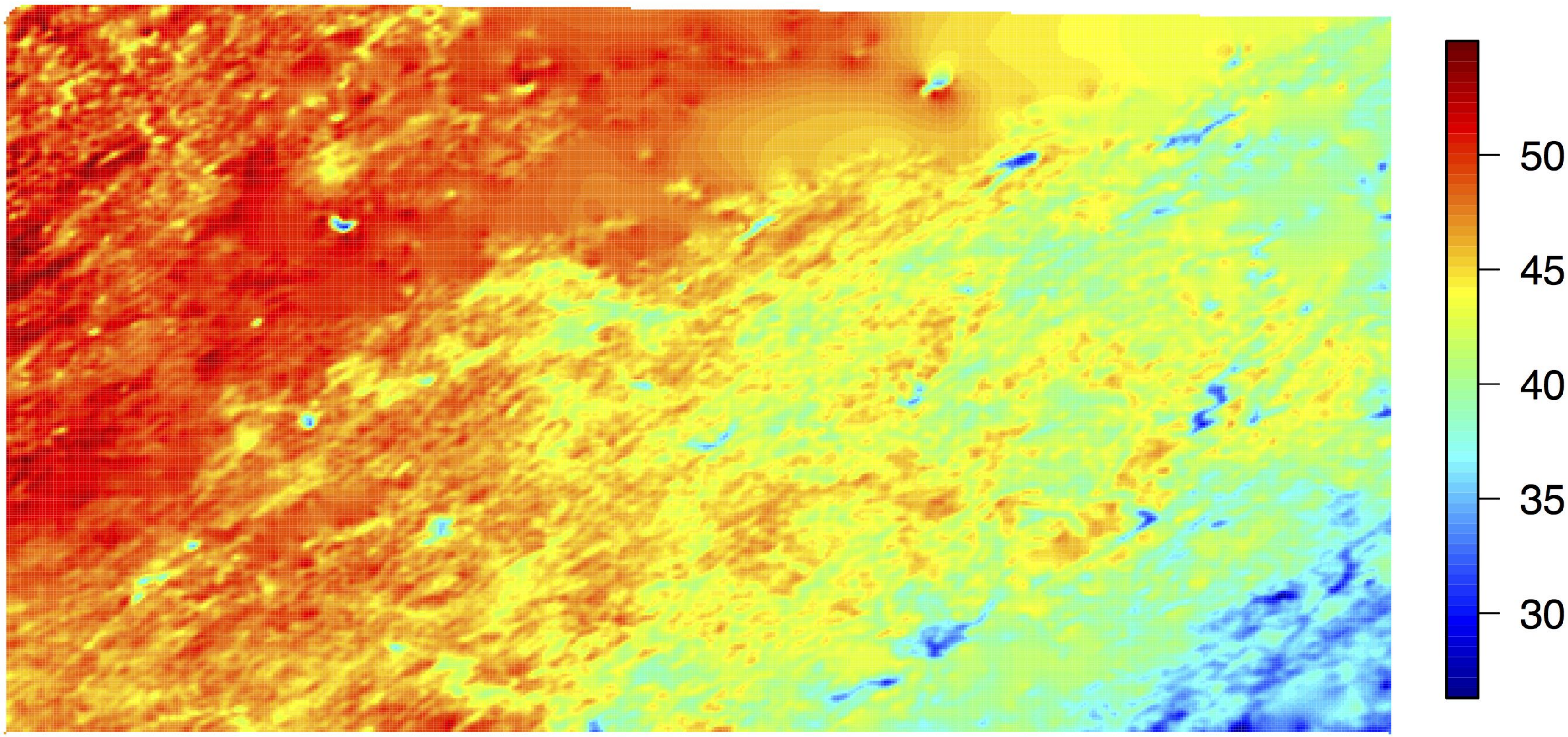
Variogram fitting

Sample Variogram with some different LatticeKrig models



- First level - 50 basis functions in longer dimension,
- a = 10, a = 4.5
- dashed - 3 levels, solid – 4 levels

Some results



Final model: `NC= 40, nlevel = 4, a.wght = 10.25, nu = .1`

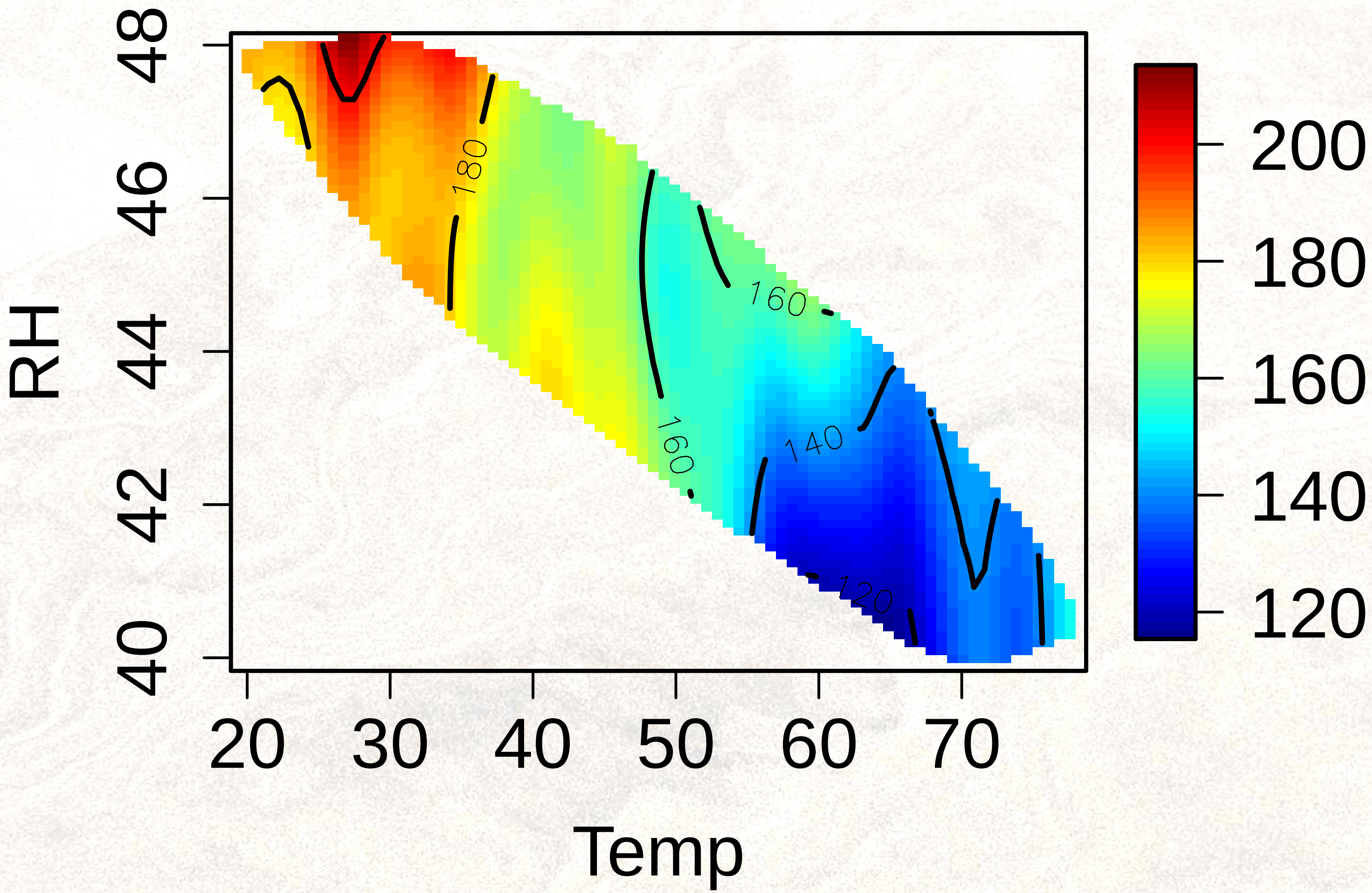
LatticeKrig package

```
# x and y are the power use data for the NCAR/WY  Supercomputing center

  fit<- LatticeKrig( x,y, NC=10, nlevel=4) # takes about 10 seconds
  print( fit)

  quilt.plot( x, fit$residuals)

  surface( fit, xlab="Temp", ylab="RH")

# Conditional simulation with fixed covariance parameters
#  50 draws 80X80 grid takes about 40 seconds

  simFit<- LKrig.sim.conditional( fit,  M=50)
```

print(fit)

```
 Call:
LatticeKrig(x = x, y = y, nlevel = 4, NC = 10)

 Number of Observations:                     1677
 Number of parameters in the fixed component 3
 Effective degrees of freedom (EDF)          32.2
    Standard Error of EDF estimate:          1.595
 MLE tau                                   30.7
 MLE sigma^2                                     21270
 MLE lambda = tau^2/sigma^2                    0.04432

Fixed part of model is a polynomial of degree 1 (m-1)
Basis function :  Radial
Basis function used:  WendlandFunction
Distance metric:  Euclidean

Lattice summary:
4  Level(s) 3112 basis functions with overlap of  2.5  (lattice units)

 Level Lattice points   Spacing
     1            240 6.5664444
     2            377 3.2832222
     3            752 1.6416111
     4           1743 0.8208056

Nonzero entries in Ridge regression matrix 357579
```

```
surface(fit, xlab="Temp", ylab="RH")
```



Final thoughts on the next steps

- Estimate parameters using score equations to avoid determinant

- Estimate parameters using cross validation.

Randomized trace Generalized Cross-Validation is amenable to iterative methods

- Parameter searching can be easily parallelized using `Rmpi`

Summary

- **Computational efficiency gained by compact basis functions and sparse roughness (precision) matrix.**

- **Multi-resolution can approximate standard covariance families (e.g. Matern)**

- **Easy to generate uncertainty measures.**

See `LatticeKrig` contributed package in R

Thank you!