NC STATE UNIVERSITY

# Conquer Big Spatial Data: The Stochastic PDE Approach

Group members: Cole Adams,
Sibo Peng,
Jasmine Wang

# Motivation

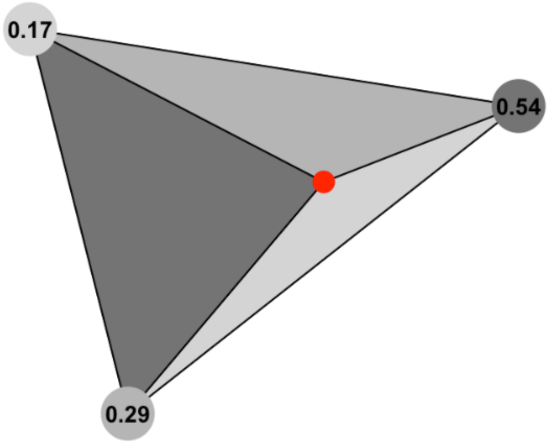
- The method of approximating a continuous Gaussian field using GMRFs was theoretically good but less practical. The SPDE method represents a Gaussian field with Matern covariance by representing a solution of stochastic partial differential equations as a GMRF using the finite element method.
- The GMRF representation of the Gaussian field, which can be computed explicitly, provides a sparse representation of the spatial effect through a sparse precision matrix. This enables the nice computational properties of the GMRFs which can then be implemented in the INLA package.

# The SPDE Model

- $\mathbf{y} \mid \beta_0, \mathbf{u}, \sigma_e^2 \sim N(\beta_0 + \mathbf{Au}, \sigma_e^2)$
- $\mathbf{u} \sim GF(0,\Sigma)$
- $y_i$ are observations at location $s_i$.
- $\beta_0$ is the intercept
- **u** is a spatial Gaussian random field with mean zero and standard deviation Σ
- **A** is the projector matrix

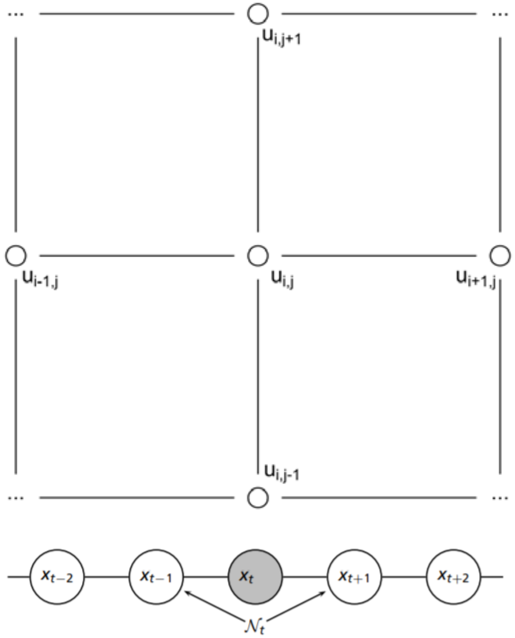
# Mesh

- Points are often distributed irregularly. Need to construct mesh
- Mesh is used in the Finite Element Method to provide a solution to a SPDE.
- To better approximate the field, need to have more triangles.
- Shiny application within INLA package: meshbuilder()



# Understanding the Method

- A Gaussian field with a generalized covariance function obtained in the Matérn correlation function when v>0 is a solution to a SPDE.
- Consider a regular two-dimensional lattice with number of sites tending to infinity.
- The right figure is an example of a GMRF. Another example is AR(1) process
- The GMRF representation is a convolution of processes with precision matrix for distinct values of smoothness v
- As the smoothness parameter v increases, the precision matrix in the GMRF representation becomes denser. This is because the conditional distributions depend on a wider neighborhood.

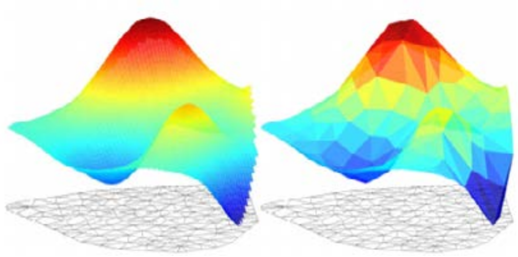


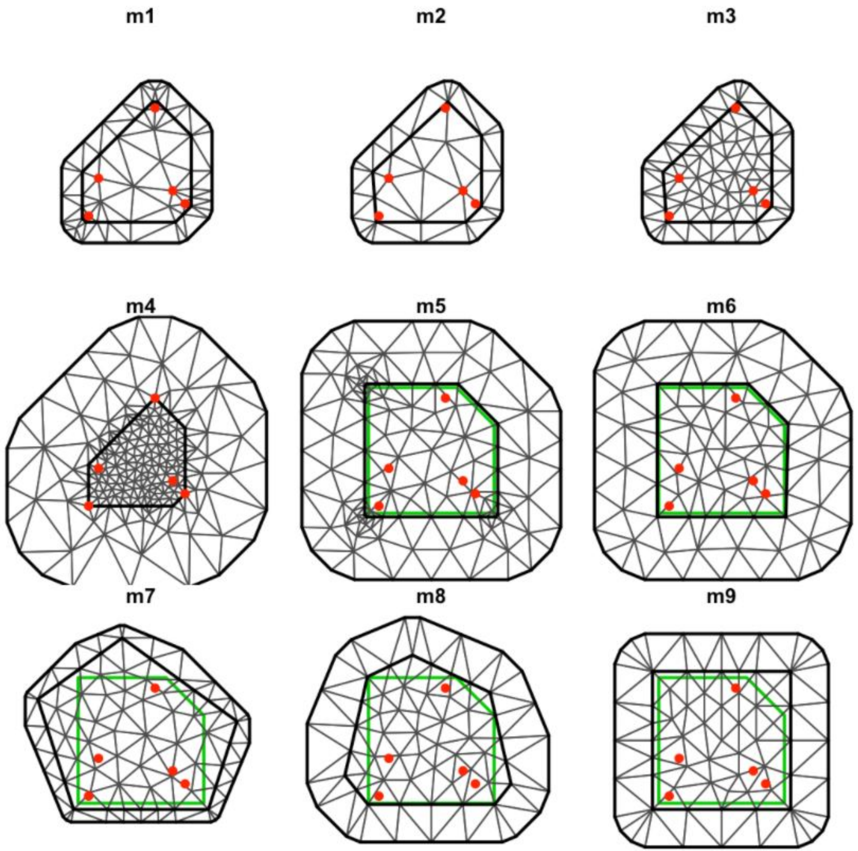
# How Approximation Works

- Illustrate the approximation in two-dimensional space considering piecewise linear basis functions.

  $u(s) = \sum_{j=1}^{m} \psi_k(s) W_k$

- where $\psi_k$ are basis functions and $W_k$ are Gaussian distributed weights, k=1,...,m with m the number of vertices in the triangulation.
- Carefully choose the basis functions to preserve the sparse structure of the resulting precision matrix for the random field at a set of mesh nodes.
- This provides an explicit link between a continuous random field and a GMRF representation, which allows efficient computations.






# To Build a 2-D Mesh

## INLA package in R

### inla.mesh.2d() function

Mandatory arguments:

- *loc*, *loc.domain*, or *boundary*
- *max.edge*: inner domain,

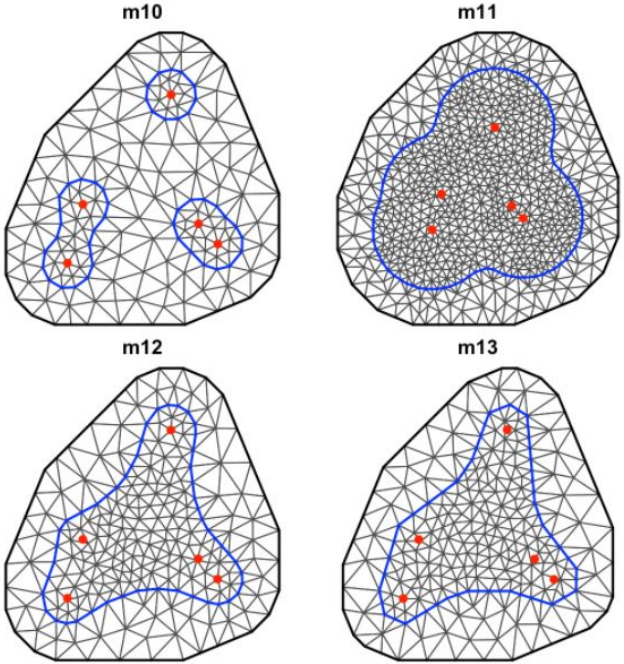
  outer extension

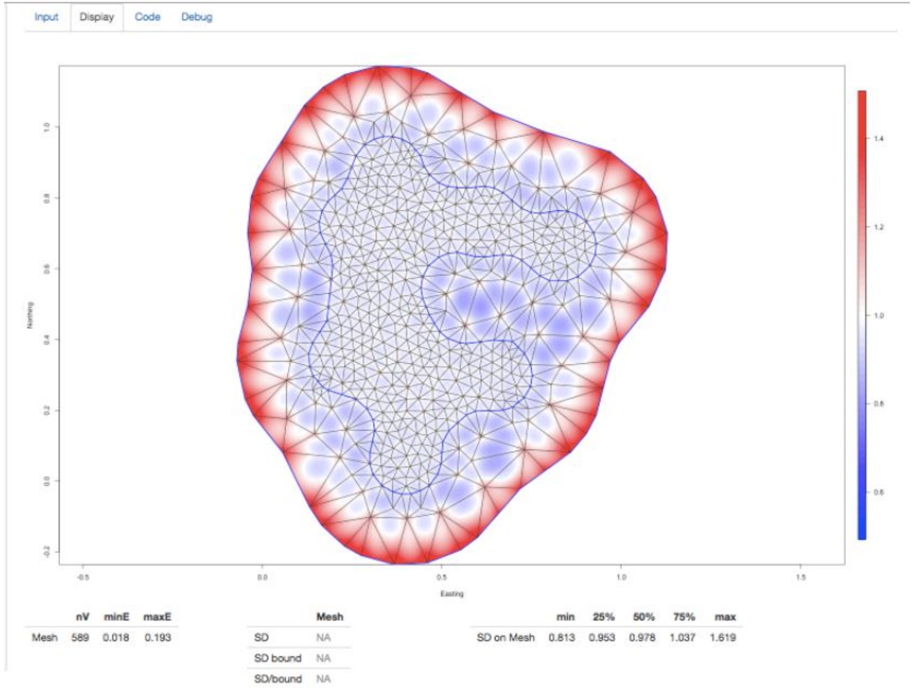
Optional arguments:

- *cutoff*
- *offset*
- *min.angle*
- *n*

* Image is taken from https://becarioprecario.bitbucket.io/spde-gitbook/index.html

# Non-convex Hull Meshes




- boundary <- **inla.nonconvex.hull(***points = , convex = , concave = , resolution = ,…***)**

**R Shiny app**: **meshbuilder()**;

* Images are taken from https://becarioprecario.bitbucket.io/spde-gitbook/index.html

# Quality of A Mesh

- **Goal**: Uniform triangle shape and size

```
mesh <- inla.mesh.2d( loc = coordinates,
                      max.edge = c(<inner domain>, <outer extension>),
                      cutoff = <a numeric value>,
                      offset = -0.10 <default> )
```

- To create the projector matrix (**A** matrix) from the mesh:

```
A <- inla.spde.make.A(mesh, loc = coordinates)
```

# Problem

- Distance is too small:

  Range(longitude) = 0.44 degrees, range(latitude) = 0.21 degrees

- Re-scale the coordinates:

  (longitude + 96) ×10; (latitude - 41) ×10

- Number of nodes for n = 20,000:

  i. node = 3,312: max.edge = c(0.1, 1), cutoff = 0.05

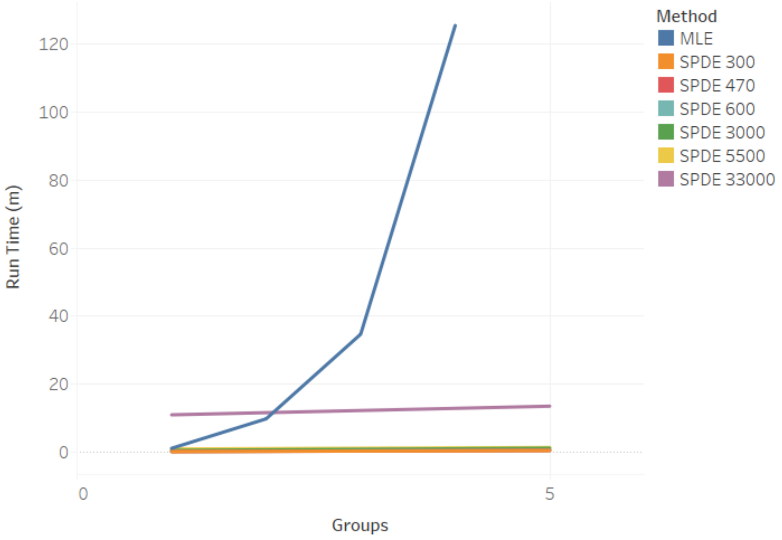
  ii. node = 5,962: max.edge = c(0.08, 0.8), cutoff = 0.03

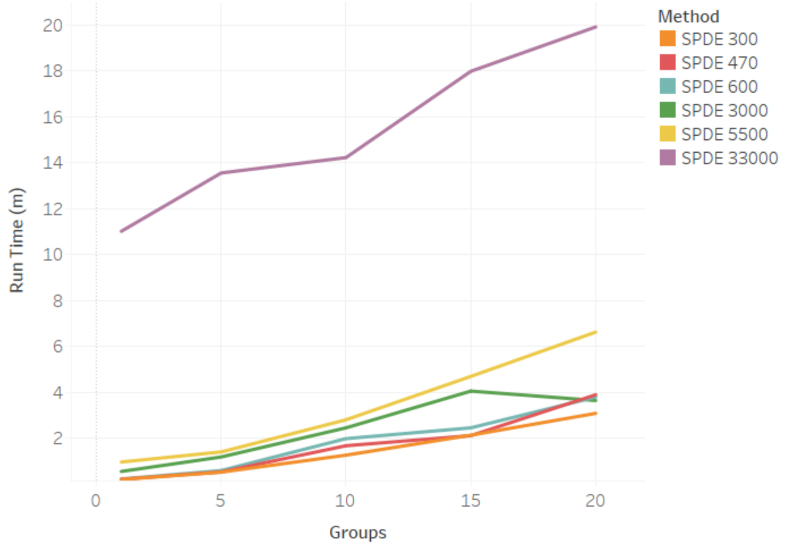
  iii. node = 34,861: max.edge = c(0.03, 0.7), cutoff = 0.01

# The SPDE Model Construction

- $\mathbf{y} \mid \beta_0, \mathbf{u}, \sigma_e^2 \sim N(\beta_0 + \mathbf{A}\mathbf{u}, \sigma_e^2)$
- $\mathbf{u} \sim GF(0, \Sigma)$
- Matern covariance with Penalized Complexity prior: **inla.spde2.pcmatern()**
  i. $\alpha \in [1, 2]; \alpha = \upsilon + {}^{d}/_{2} = 2$ <default>
  ii. $P(\sigma > \sigma_0) = p \Rightarrow P(\sigma > 1) = 0.01$:
     prior.sigma = c(1, 0.01)
  iii. $P(r < r_0) = p \Rightarrow P(r < 1.3) = 0.5$:
     prior.range = c(1.3, 0.5)

# MLE & SPDE Run Time

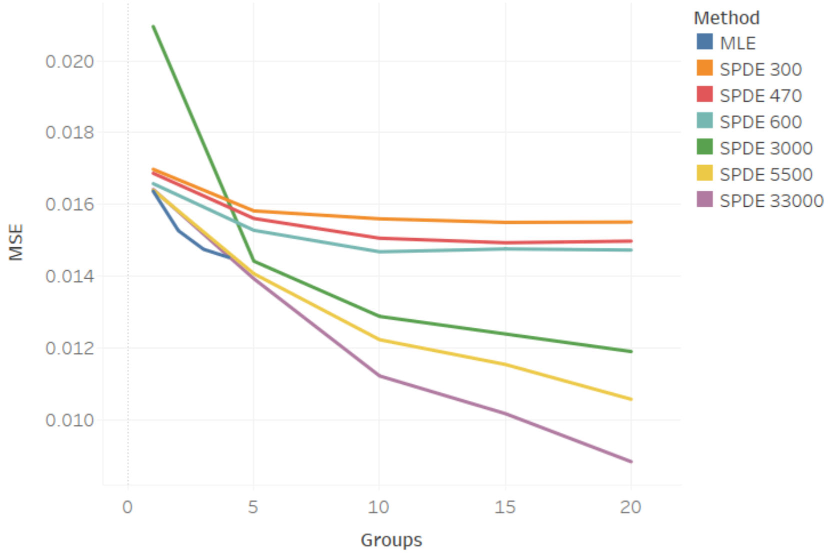


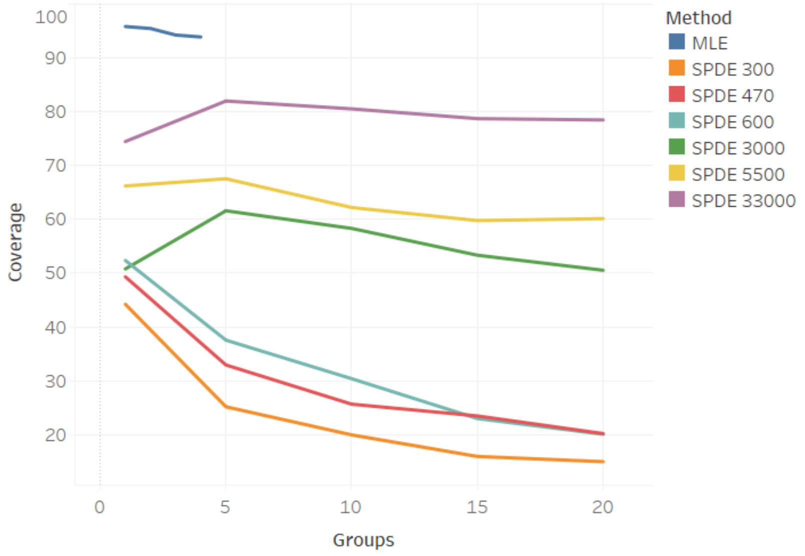




- MLE Estimation passed over one hour on fourth group
- SPDE Models runtime did not increase exponentially
- SPDE Model with most nodes (~33,000) stayed under half hour with all groups

Cole Adams

# Performance Across Group Size

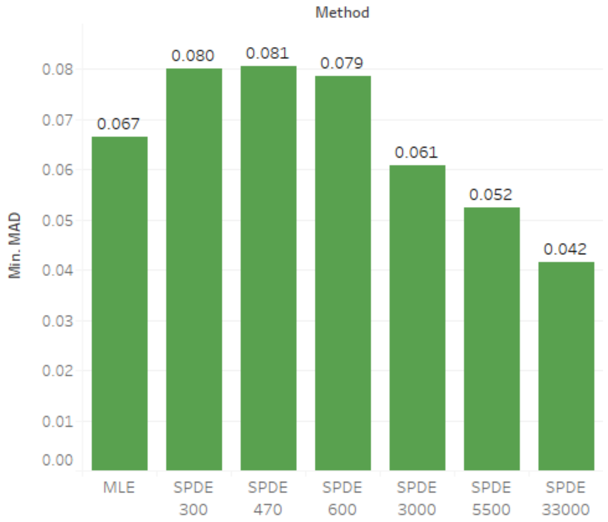


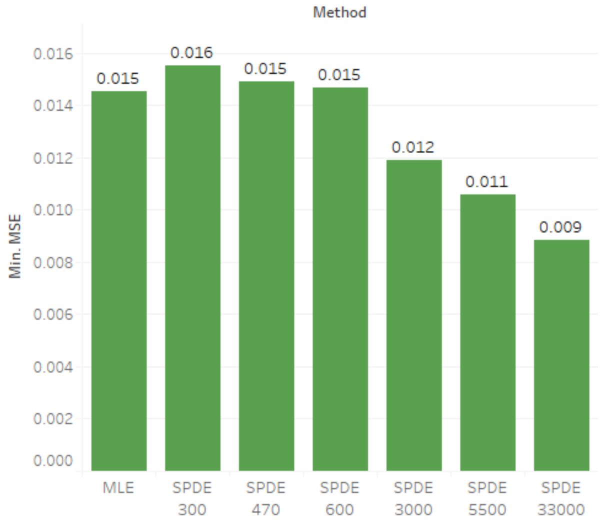




- Kriging using MLE Estimation & SPDE showed a decline in MSE as size increased
  - SPDE models with higher node counts decreased more significantly than SPDE models with smaller node counts
- Coverage stayed relatively consistent in MLE and SPDE estimation when node count was high (3000, 5500, 33000) but decreased in coverage as size increased when node counts were small

Cole Adams

# MAD & MSE

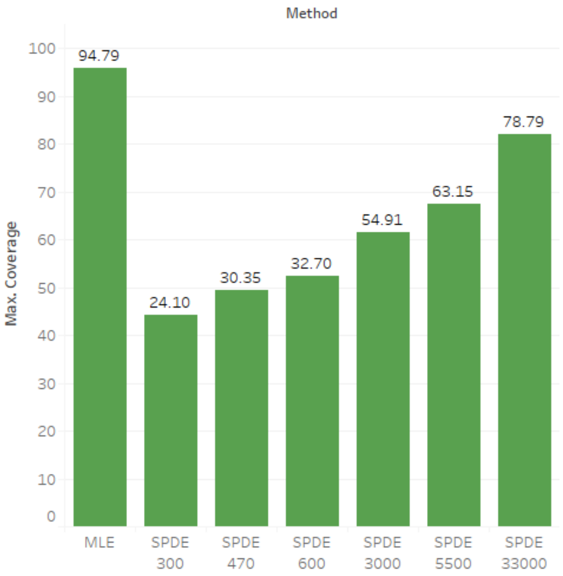



- Kriging methods using SPDE estimation with node size >3,000 performed better than MLE models
- Conversely SPDE estimation with node sizes 300, 470 and 600 performed worse than the MLE estimation

Cole Adams

# Coverage vs. Correlation

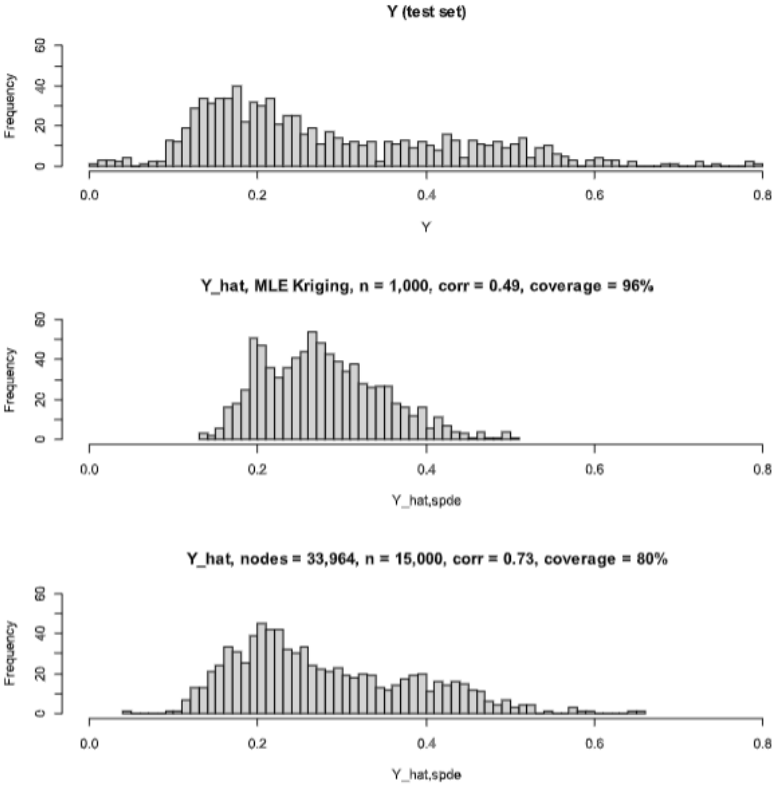




- Kriging using MLE Estimation had the highest coverage with ~95%
- As the number of nodes increased, coverage increased for SPDE-based Kriging methods
- SPDE methods using nodes 3,000, 5,500 and 33,000 resulted in higher levels of correlation than the MLE estimation

Cole Adams

# Methods Overview




- The predictions made with Kriging using the SPDE estimation better correlate with the test set.
- Additionally, it better matches the test data set in terms of the histogram, with more of the tails captured
- The predictions made with Kriging using the MLE estimation do not capture the tails of the test set and are more centered around the central value