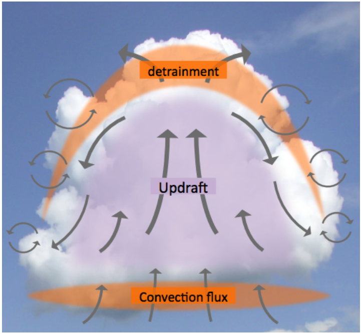
# **Adaptive Sampling of Clouds with a Fleet of UAVs**: Improving Gaussian Process Regression by Including Prior Knowledge

# Motivation: SkyScanner Project

**Adaptive Sampling of Cumulus Clouds with a Fleet of UAVs:**

Clouds remain an uncertainty in current atmospherical models:

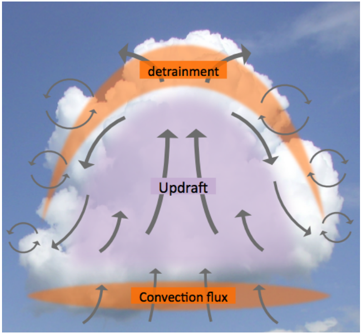
- Characterize the evolution of parameters (3D wind, liquid water content, etc.)



**Adaptive Sampling of Cumulus Clouds with a Fleet of UAVs:**

Clouds remain an uncertainty in current atmospherical models:

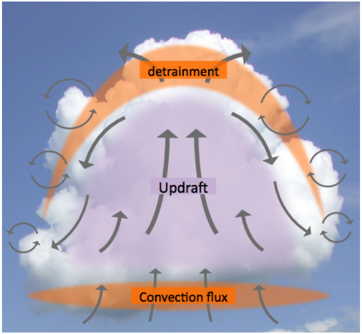
- Characterize the evolution of parameters (3D wind, liquid water content, etc.)
  $\longrightarrow$ dense spatial sampling



**Adaptive Sampling of Cumulus Clouds with a Fleet of UAVs:**

Clouds remain an uncertainty in current atmospherical models:

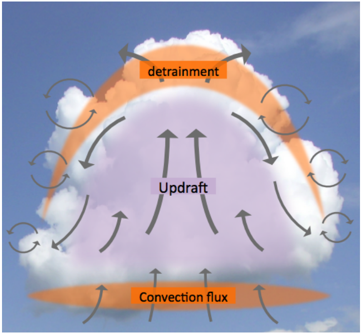
- Characterize the evolution of parameters (3D wind, liquid water content, etc.)
  $\longrightarrow$ dense spatial sampling
- **Adaptive Sampling vs. Systematic Sampling**:



**Adaptive Sampling of Cumulus Clouds with a Fleet of UAVs:**

Clouds remain an uncertainty in current atmospherical models:

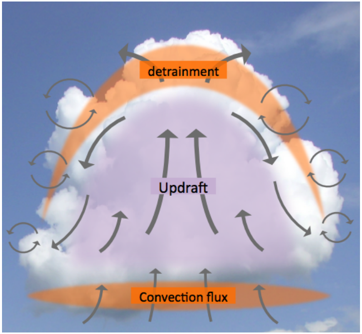
- Characterize the evolution of parameters (3D wind, liquid water content, etc.)
  $\longrightarrow$ dense spatial sampling
- **Adaptive Sampling vs. Systematic Sampling**:
  - 4D map of parameters, with only 1D manifolds available
  - Information efficiency
    $\longrightarrow$ quantification of uncertainty
  - Energy efficiency
    $\longrightarrow$ mapping and exploiting vertical wind.



**Adaptive Sampling of Cumulus Clouds with a Fleet of UAVs:**

Clouds remain an uncertainty in current atmospherical models:

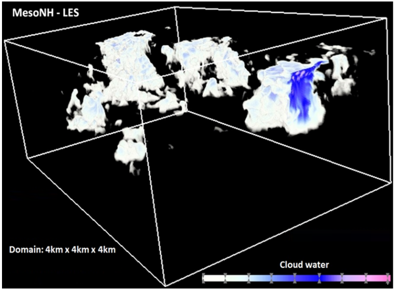
- Characterize the evolution of parameters (3D wind, liquid water content, etc.)
  $\longrightarrow$ dense spatial sampling
- **Adaptive Sampling vs. Systematic Sampling**:
  - 4D map of parameters, with only 1D manifolds available
  - Information efficiency
    $\longrightarrow$ quantification of uncertainty
  - Energy efficiency
    $\longrightarrow$ mapping and exploiting vertical wind.




$\rightarrow$ **Gaussian Process Regression**

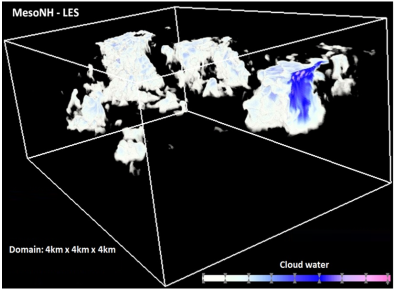
# Table of Contents

# MesoNH Simulation and Sampling Architecture



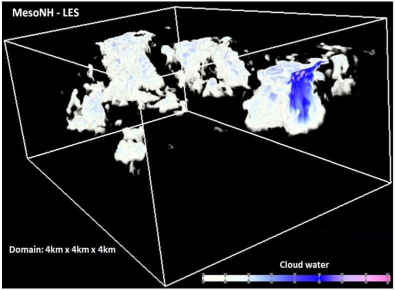

- Large Eddy Simulation(LES) of non-precipitating shallow cumulus clouds.
- **Domain**: $3540s \times 4km \times 4km \times 4km$ (3TB of data),

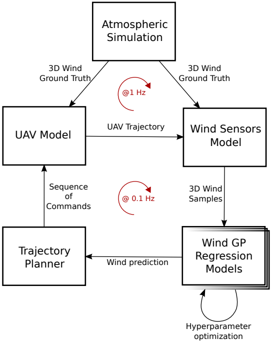
# MesoNH Simulation and Sampling Architecture




- Large Eddy Simulation(LES) of non-precipitating shallow cumulus clouds.
- **Domain**: $3540s \times 4km \times 4km \times 4km$ (3TB of data),
- **Grid**: $3540x161x400x400\,(t, z, x, y)$ and $dt = 1s$, $dx = dy = 10m$, $dz = 10m...100m$; $dz = 10m$ for boundary and convective cloud layer.
- **Variables**: 3D wind, temperature, pressure, liquid water content(LWC), etc.

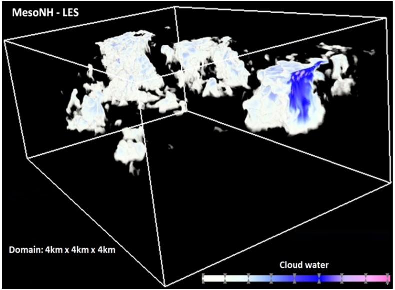
# MesoNH Simulation and Sampling Architecture

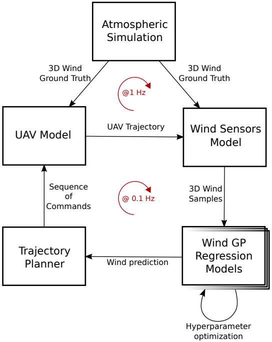



- Large Eddy Simulation(LES) of non-precipitating shallow cumulus clouds.
- **Domain**: $3540s \times 4km \times 4km \times 4km$ (3TB of data),
- **Grid**: $3540x161x400x400\,(t, z, x, y)$ and $dt = 1s$, $dx = dy = 10m$, $dz = 10m...100m$; $dz = 10m$ for boundary and convective cloud layer.
- **Variables**: 3D wind, temperature, pressure, liquid water content(LWC), etc.

# MesoNH Simulation and Sampling Architecture



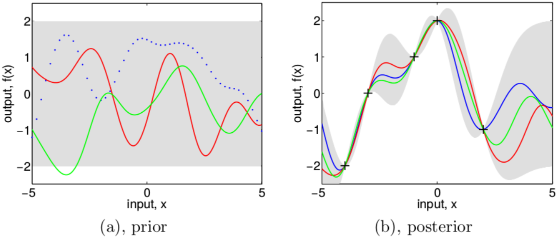

- Large Eddy Simulation(LES) of non-precipitating shallow cumulus clouds.
- **Domain**: $3540s \times 4km \times 4km \times 4km$ (3TB of data),
- **Grid**: $3540x161x400x400\,(t, z, x, y)$ and $dt = 1s$, $dx = dy = 10m$, $dz = 10m...100m$; $dz = 10m$ for boundary and convective cloud layer.
- **Variables**: 3D wind, temperature, pressure, liquid water content(LWC), etc.

$\rightarrow$ Wind predictions needed under real-time constraints

# *Introduction to Gaussian Process Regression*

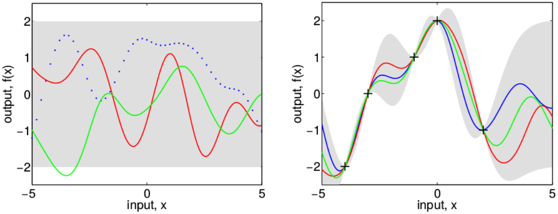
- Bayesian Machine Learning framework
- Generalization of the M-dim. Gaussian distribution to stochastic processes(functions), i.e. a Gaussian distribution over functions:

- Bayesian Machine Learning framework
- Generalization of the M-dim. Gaussian distribution to stochastic processes(functions), i.e. a Gaussian distribution over functions:



# Introduction to Gaussian Process Regression

- Bayesian Machine Learning framework
- Generalization of the M-dim. Gaussian distribution to stochastic processes(functions), i.e. a Gaussian distribution over functions:




(a), prior (b), posterior

**Two key ingredients**

- Mean function $m(\mathbf{x})$: Center for the distribution of functions
- Covariance function, matrix $k(\mathbf{x}, \mathbf{x}')$, $\Sigma$:
  Defines smoothness and variability. Quantifies similarity.
  If $\mathbf{x}$,$\mathbf{x}'$ similar $\longrightarrow$ outputs similar

**Making predictions**

With training data: $\mathbf{X}, \mathbf{Y}$ | new input vector $\mathbf{x}_\star$| mean function $m(\mathbf{x})$ | covariance matrices $\Sigma_{\mathbf{X},\mathbf{X}} = [k(\mathbf{x}_i, \mathbf{x}_i)]\,,\; i, j = 1, ..., n$ | $\Sigma_{\mathbf{x}_\star,\mathbf{X}} = [k(\mathbf{x}_\star, \mathbf{x}_i)]\,,\; i = 1, ..., n$ |

**Making predictions**

With training data: $\mathbf{X}, \mathbf{Y}$ | new input vector $\mathbf{x}_\star$| mean function $m(\mathbf{x})$ | covariance matrices $\Sigma_{\mathbf{X},\mathbf{X}} = [k(\mathbf{x}_i, \mathbf{x}_j)]$ , $i, j = 1, ..., n$ | $\Sigma_{\mathbf{x}_\star,\mathbf{X}} = [k(\mathbf{x}_\star, \mathbf{x}_i)]$ , $i = 1, ..., n$ |

$$p\left(y_\star|\mathbf{x}_\star, \mathbf{X}, \mathbf{Y}\right) = \mathcal{N}\left(\bar{y}_\star, \mathbb{V}[y_\star]\right), \quad (1)$$

$$\bar{y}_\star = m(\mathbf{x}_\star) + \Sigma_{\mathbf{x}_\star,\mathbf{X}}\Sigma_{X,X}^{-1}(\mathbf{Y} - m(\mathbf{X})), \quad (2)$$

$$\mathbb{V}[y_\star] = k(\mathbf{x}_\star, \mathbf{x}_\star) - \Sigma_{\mathbf{x}_\star,\mathbf{X}}\Sigma_{X,X}^{-1}\Sigma_{\mathbf{x}_\star,\mathbf{X}}^{T} \quad (3)$$

**Making predictions**

With training data: $\mathbf{X}, \mathbf{Y}$ | new input vector $\mathbf{x}_\star$| mean function $m(\mathbf{x})$ | covariance matrices $\Sigma_{\mathbf{X},\mathbf{X}} = [k(\mathbf{x}_i, \mathbf{x}_j)]$ , $i, j = 1, ..., n$ | $\Sigma_{\mathbf{x}_\star,\mathbf{X}} = [k(\mathbf{x}_\star, \mathbf{x}_i)]$ , $i = 1, ..., n$ |

$$p\left(y_\star | \mathbf{x}_\star, \mathbf{X}, \mathbf{Y}\right) = \mathcal{N}\left(\bar{y}_\star, \mathbb{V}[y_\star]\right), \tag{1}$$

$$\bar{y}_\star = m(\mathbf{x}_\star) + \Sigma_{\mathbf{x}_\star,\mathbf{X}} \Sigma_{X,X}^{-1} (\mathbf{Y} - m(\mathbf{X})), \tag{2}$$

$$\mathbb{V}[y_\star] = k(\mathbf{x}_\star, \mathbf{x}_\star) - \Sigma_{\mathbf{x}_\star,\mathbf{X}} \Sigma_{X,X}^{-1} \Sigma_{\mathbf{x}_\star,\mathbf{X}}^T \tag{3}$$

**Advantages of GPR**

- Inbuilt estimation of uncertainty adapted to test inputs

**Limitations**

- Mean function and covariance function are parameterized
  $\longrightarrow$ Expensive optimization, usually Bayesian Marginal Log-Likelihood (several iterations of $\mathcal{O}(n^3)$)
- With no prior knowledge about process, "off-the-shelf":
  $\longrightarrow m(\mathbf{x}) = 0, \ k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(\frac{-0.5|\mathbf{x} - \mathbf{x}'|^2}{l^2}\right)$

**Types of prior knowledge to improve GPR:**

1. Determining the mean function $m(\mathbf{x})$
2. Determining type and parameter distribution of covariance function $k(\mathbf{x}, \mathbf{x}')$
3. If output multidimensional, then determine and exploit correlations between outputs

# *Introduction to Gaussian Process Regression*

**Types of prior knowledge to improve GPR:**

1. Determining the mean function $m(\mathbf{x})$
2. Determining type and parameter distribution of covariance function $k(\mathbf{x}, \mathbf{x}')$
3. If output multidimensional, then determine and exploit correlations between outputs

**Approaches to determine prior knowledge**

- *Brute Force*:
  - Cross-validate implementations that combine several mean-functions, covariance functions and output-correlation structures
    $\longrightarrow$ No real understanding about the process
    $\longrightarrow$ Computational complexity $\mathcal{O}(n^3)$

# Introduction to Gaussian Process Regression

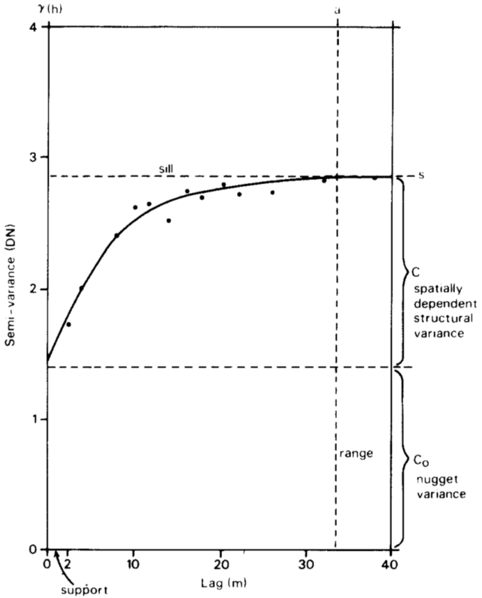
**Types of prior knowledge to improve GPR:**

1. Determining the mean function $m(\mathbf{x})$
2. Determining type and parameter distribution of covariance function $k(\mathbf{x}, \mathbf{x}')$
3. If output multidimensional, then determine and exploit correlations between outputs

**Approaches to determine prior knowledge**

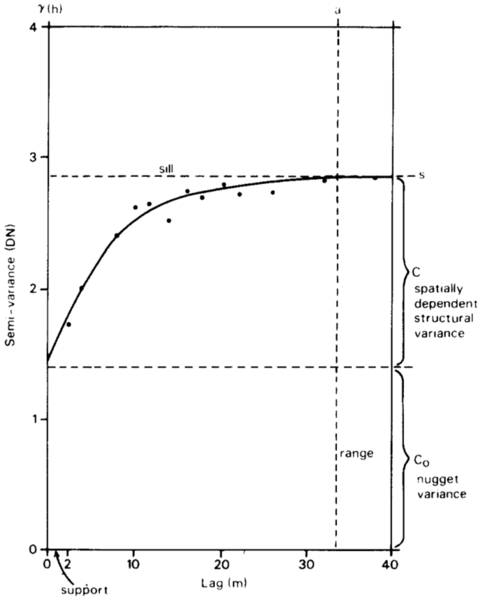
- *Brute Force*:
  - Cross-validate implementations that combine several mean-functions, covariance functions and output-correlation structures
    $\longrightarrow$ No real understanding about the process
    $\longrightarrow$ Computational complexity $\mathcal{O}(n^3)$
- *Spatial Statistics, Geostatistics*:
  - Estimate statistics from data and do regular curve fitting on these statistics to infer the priors
    $\longrightarrow$ Computational complexity: statistics $\mathcal{O}(n)$, curve fitting $\mathcal{O}(m^3)$, $m << n$

# Spatial Statistics: The Variogram



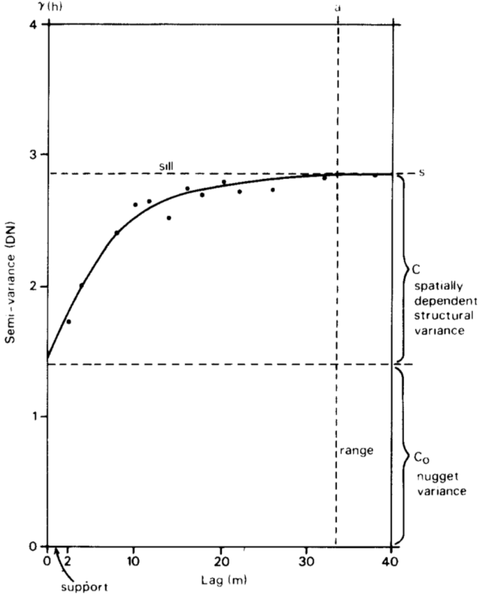

- $2\gamma(\mathbf{x}, \mathbf{x}')$ is a measure of dissimilarity between $\mathbf{x}$ and $\mathbf{x}'$

# Spatial Statistics: The Variogram




- $2\gamma(\mathbf{x}, \mathbf{x}')$ is a measure of dissimilarity between $\mathbf{x}$ and $\mathbf{x}'$
- With assumptions *Stationarity* and *Isotropy*: $2\gamma(h)$, distance $h = |\mathbf{x} - \mathbf{x}'|$
- Estimated from the data and then fitted with a model

# Spatial Statistics: The Variogram




- $2\gamma(\mathbf{x}, \mathbf{x}')$ is a measure of dissimilarity between $\mathbf{x}$ and $\mathbf{x}'$
- With assumptions *Stationarity* and *Isotropy*: $2\gamma(h)$, distance $h = |\mathbf{x} - \mathbf{x}'|$
- Estimated from the data and then fitted with a model
- Basis for spatial prediction in Geostatistics, i.e. *Kriging*
- Non-converging empirical variograms indicate problems with *Stationarity*

**Estimating**

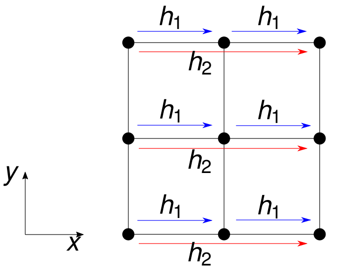
$$2\hat{\gamma}(\mathbf{h}) \equiv \frac{1}{|N(\mathbf{h})|} \sum_{N(\mathbf{h})} (Z(\mathbf{s}_i) - Z(\mathbf{s}_j))^2, \mathbf{h} \in \mathbb{R}^d, \tag{4}$$

$$N(\mathbf{h}) \equiv \{(\mathbf{s}_i, \mathbf{s}_j) : \mathbf{s}_i - \mathbf{s}_j = \mathbf{h}; i, j = 1, ..., n\} \tag{5}$$

**Estimating**

$$2\hat{\gamma}(\mathbf{h}) \equiv \frac{1}{|N(\mathbf{h})|} \sum_{N(\mathbf{h})} (Z(\mathbf{s}_i) - Z(\mathbf{s}_j))^2, \mathbf{h} \in \mathbb{R}^d, \quad (4)$$

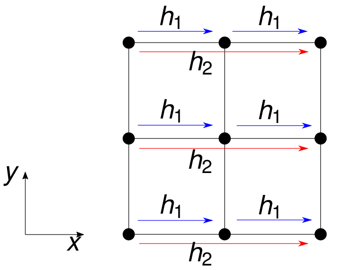
$$N(\mathbf{h}) \equiv \{(\mathbf{s}_i, \mathbf{s}_j) : \mathbf{s}_i - \mathbf{s}_j = \mathbf{h}; i, j = 1, ..., n\} \quad (5)$$

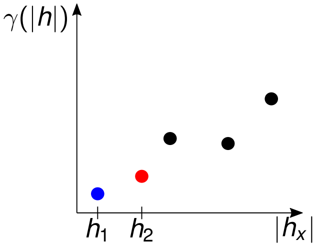


## Estimating

$$2\hat{\gamma}(\mathbf{h}) \equiv \frac{1}{|N(\mathbf{h})|} \sum_{N(\mathbf{h})} (Z(\mathbf{s}_i) - Z(\mathbf{s}_j))^2, \mathbf{h} \in \mathbb{R}^d, \quad (4)$$

$$N(\mathbf{h}) \equiv \{(\mathbf{s}_i, \mathbf{s}_j) : \mathbf{s}_i - \mathbf{s}_j = \mathbf{h}; i, j = 1, ..., n\} \quad (5)$$

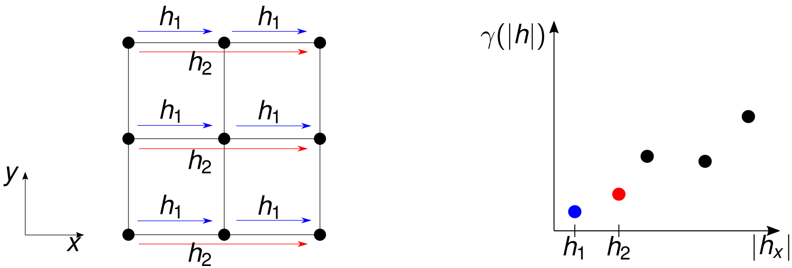


**Estimating**

$$2\hat{\gamma}(\mathbf{h}) \equiv \frac{1}{|N(\mathbf{h})|}\sum_{N(\mathbf{h})}(Z(\mathbf{s}_i) - Z(\mathbf{s}_j))^2, \mathbf{h} \in \mathbb{R}^d, \quad (4)$$

$$N(\mathbf{h}) \equiv \{(\mathbf{s}_i, \mathbf{s}_j) : \mathbf{s}_i - \mathbf{s}_j = \mathbf{h}; i, j = 1, ..., n\} \quad (5)$$




**Fitting**

$$\sum_{j=1}^{k} |N(h(j))| \left(\frac{\hat{\gamma}(h(j))}{\gamma(h(j);\theta)} - 1\right)^2 \quad (6)$$

# Spatial Statistics: The Variogram and Gaussian Process Regression

Converging variogram models and stationary covariance functions are related:

$$\gamma(\mathbf{h}) = k(\mathbf{0}) - k(\mathbf{h}), \tag{7}$$

$$k(\mathbf{h}) = \gamma(\infty) - \gamma(\mathbf{h}), \tag{8}$$

# Spatial Statistics: The Variogram and Gaussian Process Regression

Converging variogram models and stationary covariance functions are related:

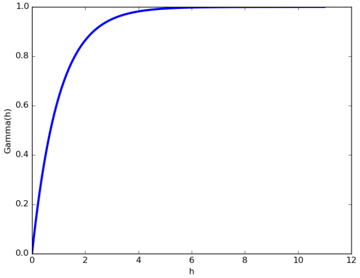
$$\gamma(\mathbf{h}) = k(\mathbf{0}) - k(\mathbf{h}), \tag{7}$$

$$k(\mathbf{h}) = \gamma(\infty) - \gamma(\mathbf{h}), \tag{8}$$

**Examples**:

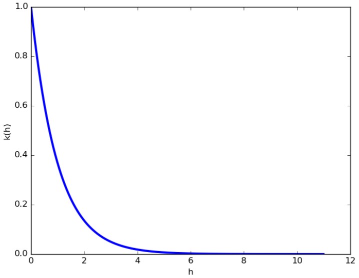
Exponential Variogram

$$\gamma(h) = \sigma^2(1 - \exp(\frac{-|h|}{l}))$$



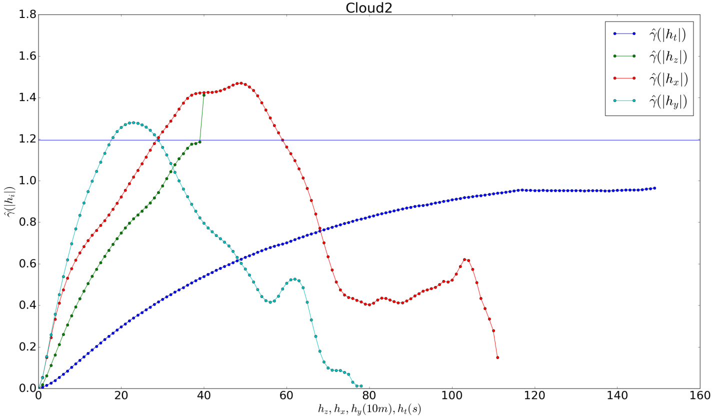

Exponential Covariance Function

$$k(h) = \sigma^2 \exp(\frac{-|h|}{l})$$



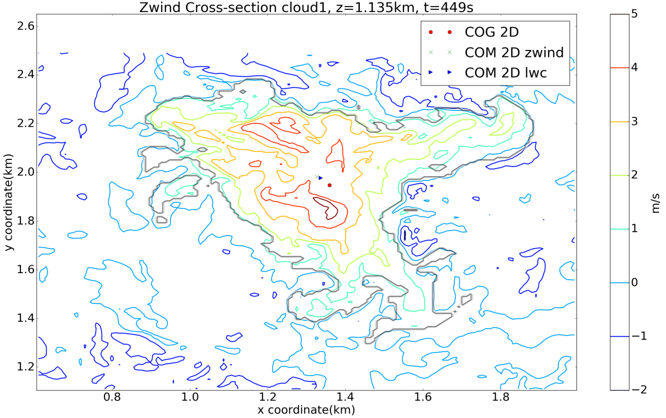
# Implementation: Vertical Wind Empirical Variograms

- 5 Clouds were segmented, and used to estimate variograms in $t, z, x, y$




- Values at big distances are very similar in $x, y$
- Variograms continue to grow over theoretical sill in $x, y$
  $\longrightarrow$ Non-stationarity, mean function?

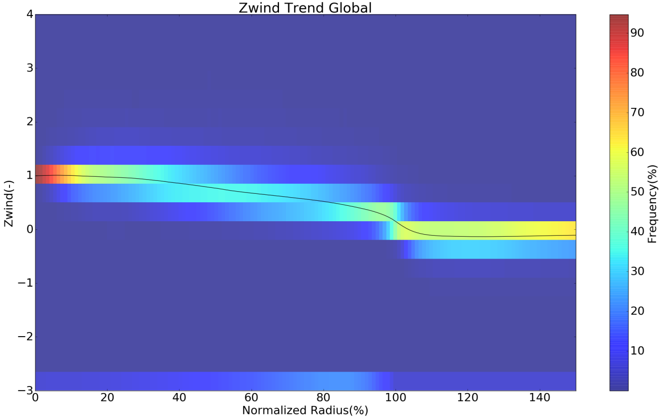
# Implementation: New coordinates




- Polar coordinates based on center of LWC more “natural”
- Vertical winds near the center are higher, near boundaries lower
  $\longrightarrow$ Radial mean function?

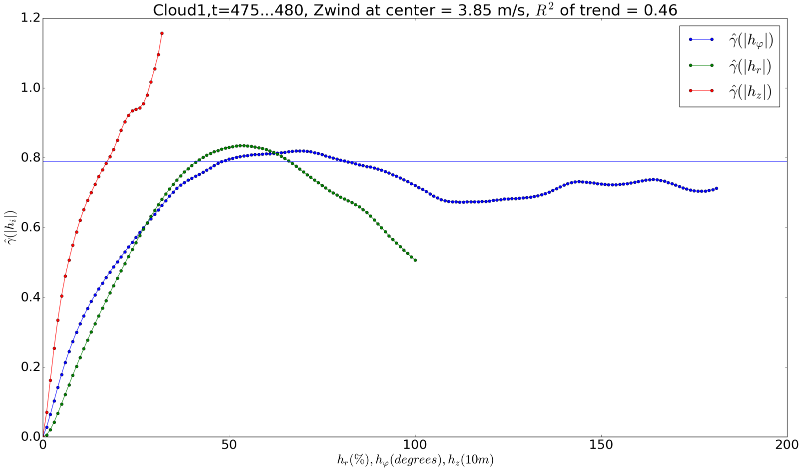
# Implementation: Estimating the Mean Function

- Normalization of radius and vertical wind at center
- Over 300.000 radial trends to estimate the median



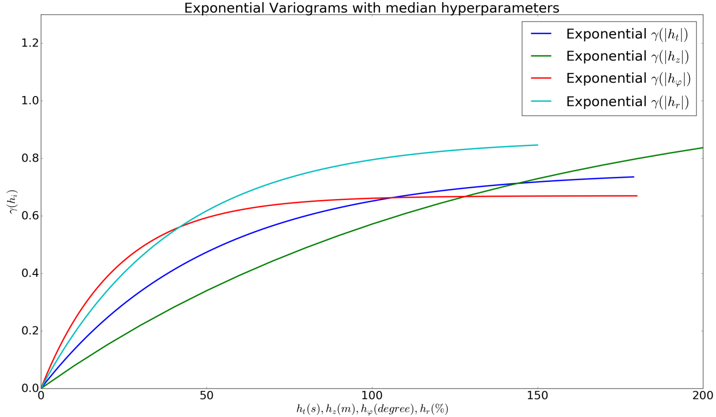
# Implementation: Detrended Empirical Variograms

- Clouds were detrended with the mean function
- New variograms were computed in the four polar directions $t, z, \varphi, r$

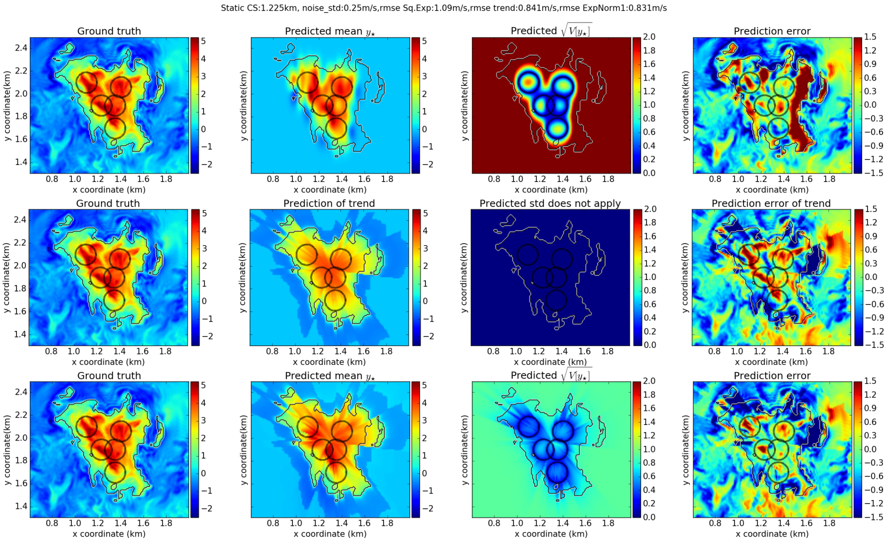


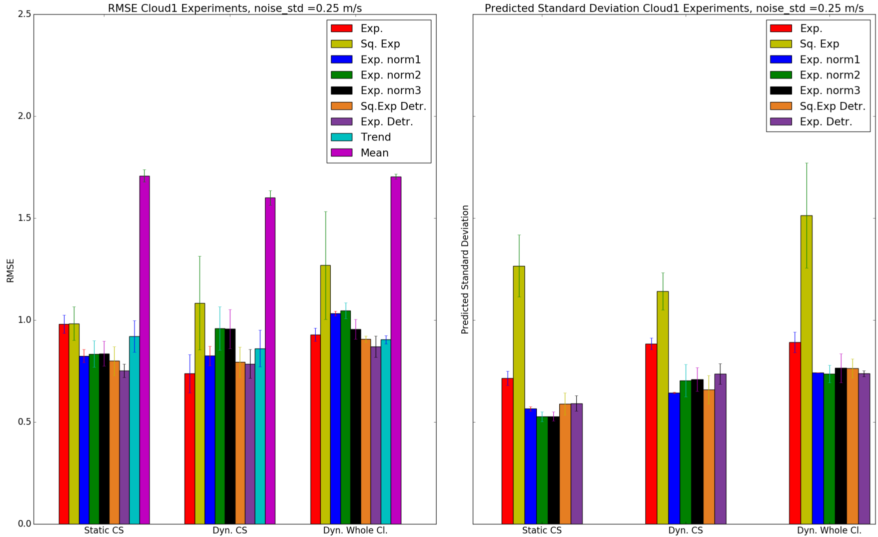
# Implementation: Best Fit Detrended Variograms

- Around 20-30 variograms with detrended vertical wind were fitted
  $\longrightarrow$ Parameters of covariance function
- Out of four possible models tested, Exponential Variogram best fit
- Similarity in sills suggests that range anisotropy is more accentuated
  $\longrightarrow \gamma(|r|), \quad r^2 \equiv \mathbf{h}^T\mathbf{M}\mathbf{h}, \quad \mathbf{M} = diag(1/l_{x_i}^2)$



# Implementation: Testing the new GPR




RMSE Cloud1 Experiments, noise_std =0.25 m/s
RMSE
0.0
0.5
1.0
1.5
2.0
2.5
Static CS
Dyn. CS
Dyn. Whole Cl.
Exp.
Sq. Exp
Exp. norm1
Exp. norm2
Exp. norm3
Sq.Exp Detr.
Exp. Detr.
Trend
Mean
Predicted Standard Deviation Cloud1 Experiments, noise_std =0.25 m/s
Predicted Standard Deviation
Static CS
Dyn. CS
Dyn. Whole Cl.
Exp.
Sq. Exp
Exp. norm1
Exp. norm2
Exp. norm3
Sq.Exp Detr.
Exp. Detr.

**Summary**

- Prior on mean function ✓
- Prior on covariance function ✓
- Improved performance vs. “off-the-shelf” GPR ✓

**Outlook**

- Repeat line of analysis on other variables, e.g. liquid water content(LWC)
- Exploit correlations between LWC and vertical wind
- Integrate polar coordinates preprocessing to current adaptive sampling scheme

Questions?