The SPDE Method for Source Inference

Colin Okasaki, Mevin Hooten

Motivation

Motivation

- Pollution

Motivation

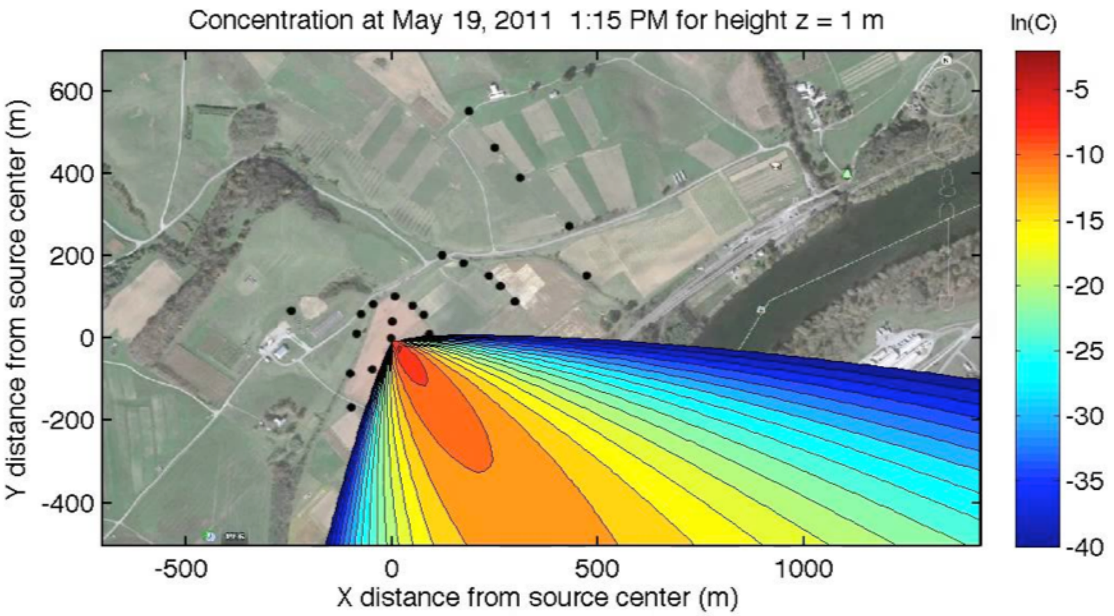
- Pollution
  - Say, PCBs in the Duwamish

Motivation

- Pollution
    - Say, PCBs in the Duwamish
    - Where did it come from?

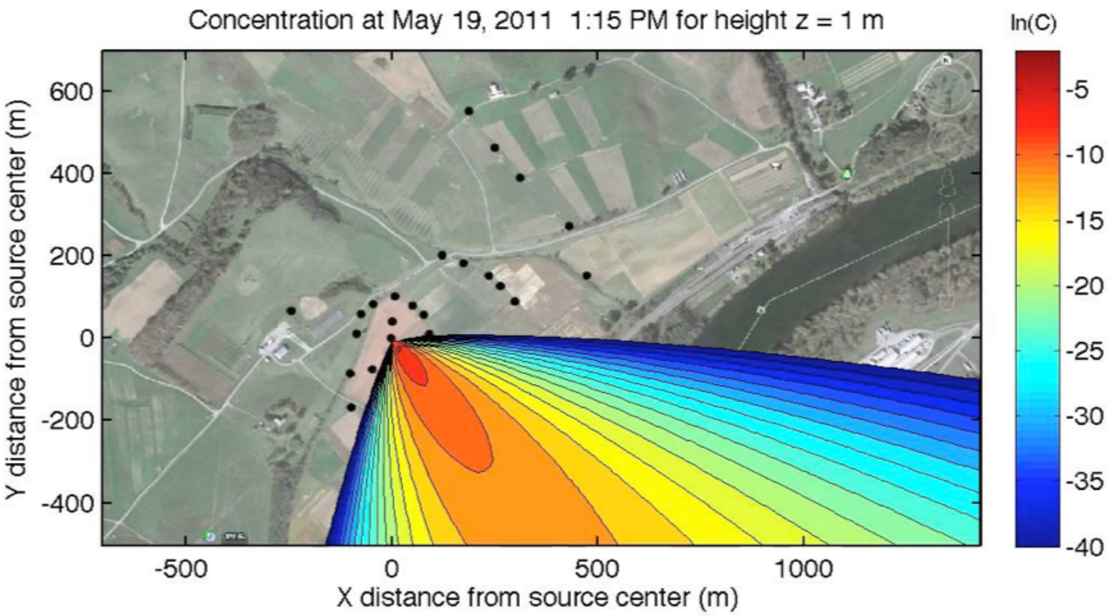
Motivation

- Pollution
  - Say, PCBs in the Duwamish
  - Where did it come from?
  - Easy case: point source

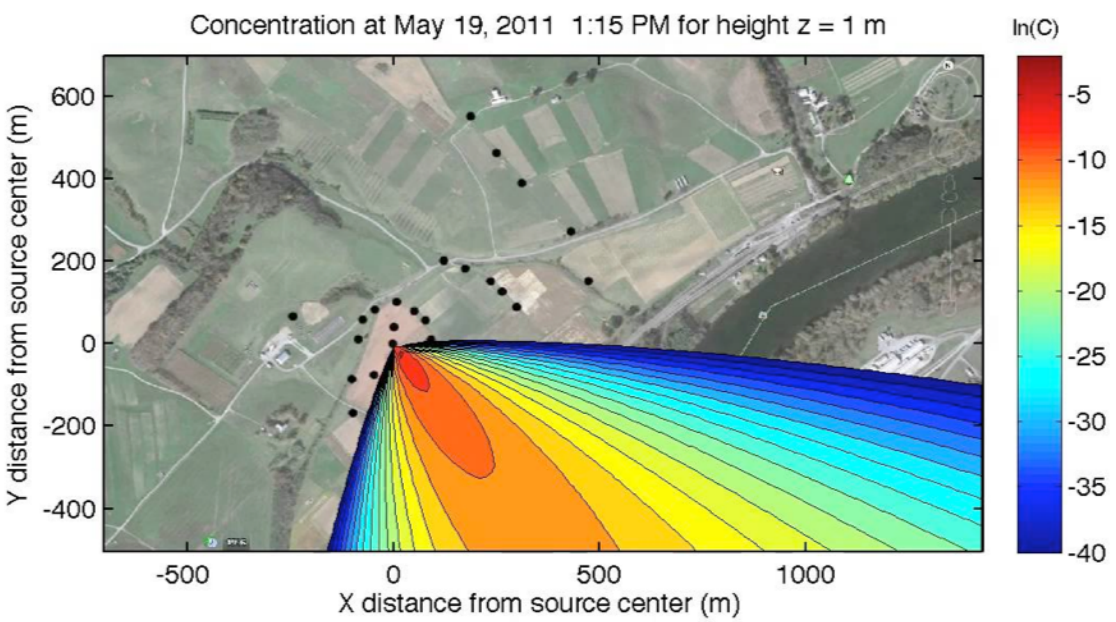


Motivation

- Pollution
  - Say, PCBs in the Duwamish
  - Where did it come from?
  - Easy case: point source
    - Measure all pipe outlets



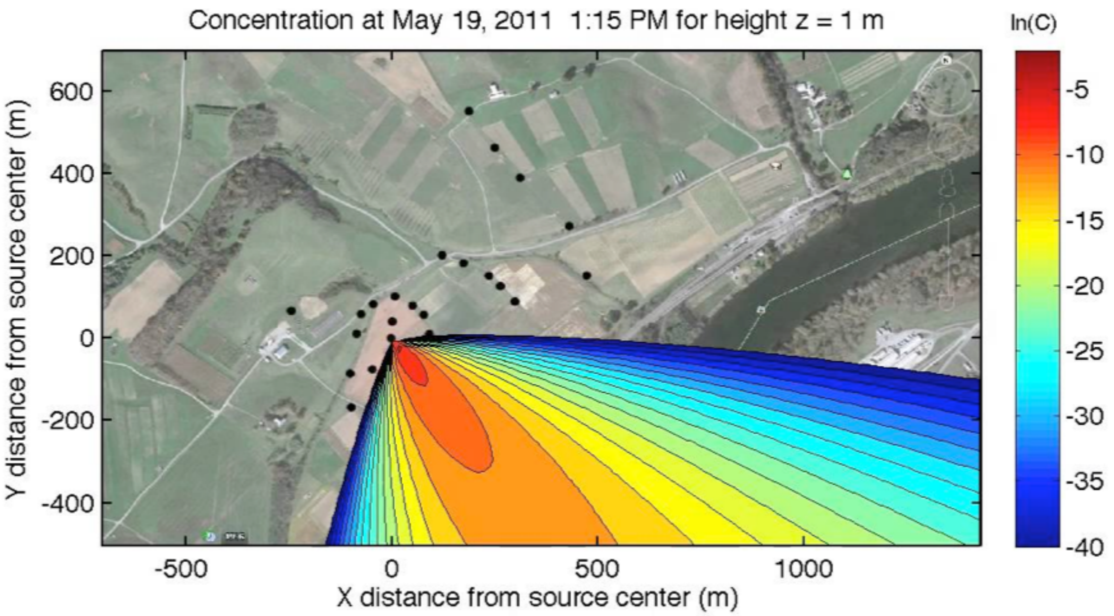
Motivation

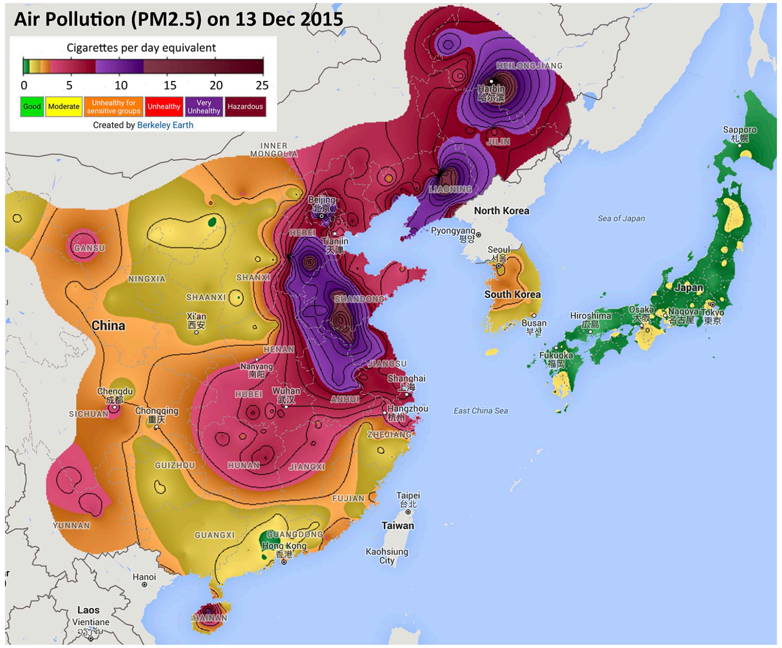


- Pollution
  - Say, PCBs in the Duwamish
  - Where did it come from?
  - Easy case: point source
    - Measure all pipe outlets
    - Use “inverse problems” method for unknown source

Motivation

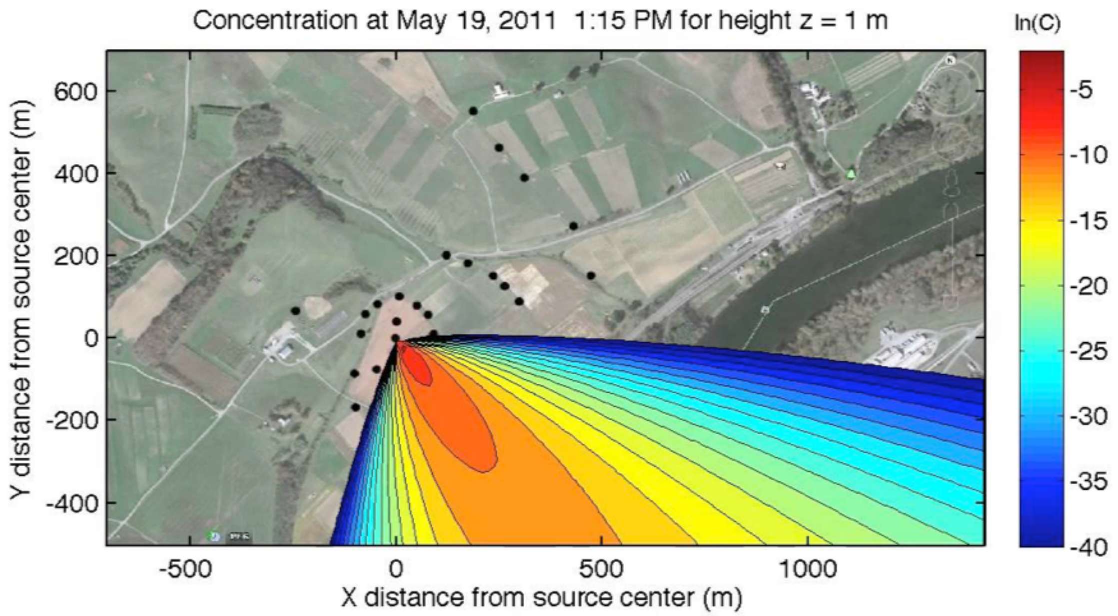
- Pollution
  - Say, PCBs in the Duwamish
  - Where did it come from?
  - Easy case: point source
    - Measure all pipe outlets
    - Use “inverse problems” method for unknown source
  - Hard case: non-point source

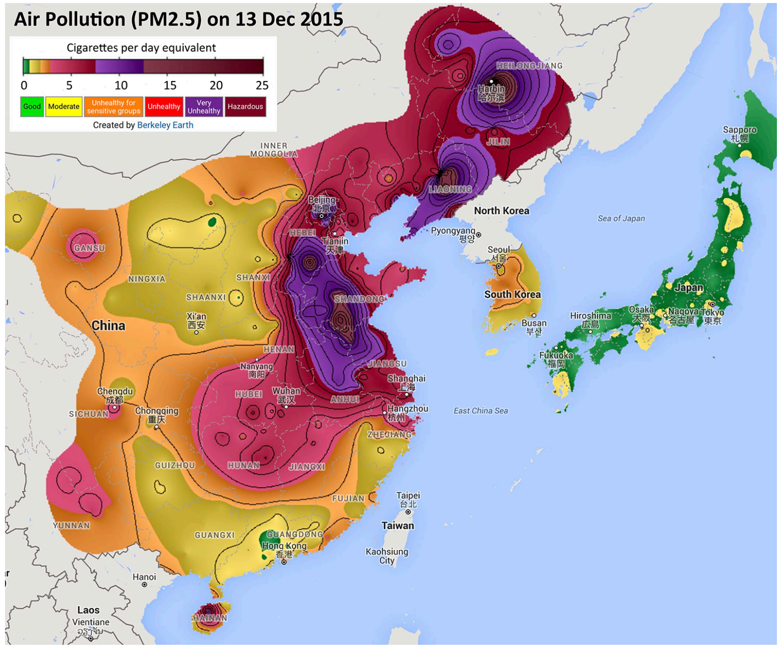




Motivation

- Pollution
  - Say, PCBs in the Duwamish
  - Where did it come from?
  - Easy case: point source
    - Measure all pipe outlets
    - Use “inverse problems” method for unknown source
  - Hard case: non-point source
    - ???





Possible Approaches

Possible Approaches

- Basically this is the advection-diffusion equation

Possible Approaches

- Basically this is the advection-diffusion equation
    - Possibly with linear decay

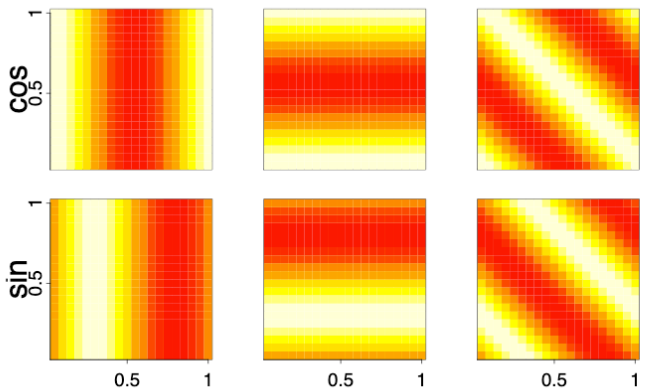
Possible Approaches

- Basically this is the advection-diffusion equation
  - Possibly with linear decay
- Existing methods

Possible Approaches

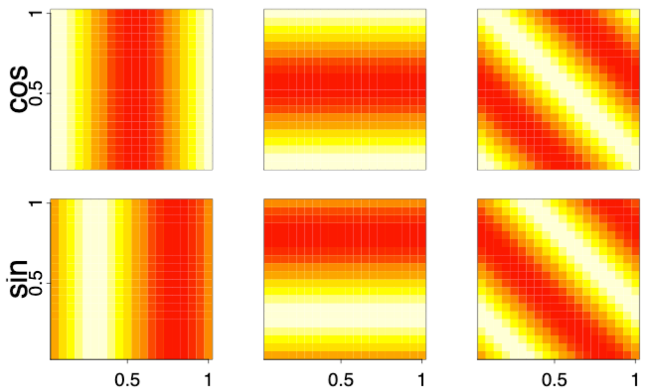
- Basically this is the advection-diffusion equation
  - Possibly with linear decay
- Existing methods
  - Fast Fourier Transform/Kalman filter (Sigrist et. al 2014)

Possible Approaches



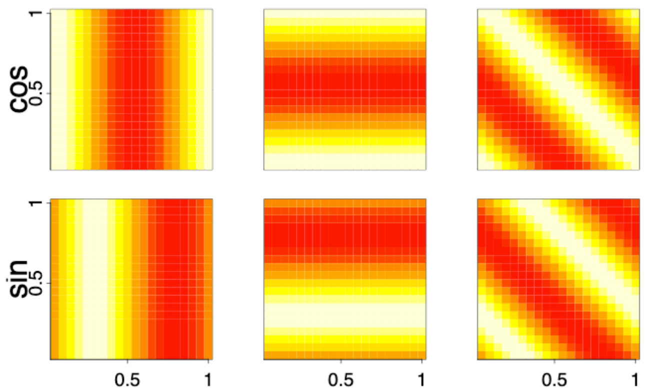
- Basically this is the advection-diffusion equation
    - Possibly with linear decay
- Existing methods
    - Fast Fourier Transform/Kalman filter (Sigrist et. al 2014)
        - Relatively complex, non-local basis functions

Possible Approaches



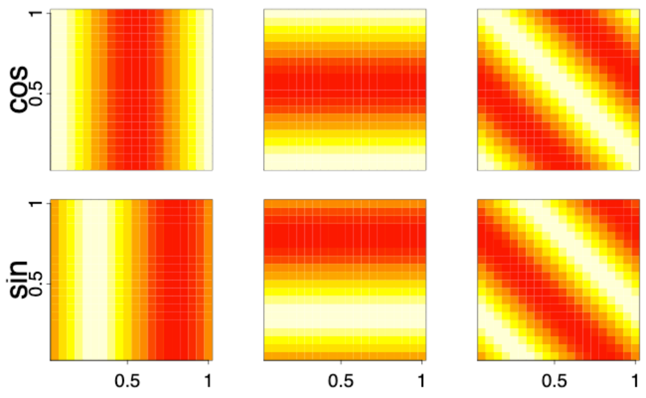
- Basically this is the advection-diffusion equation
  - Possibly with linear decay
- Existing methods
  - Fast Fourier Transform/Kalman filter (Sigrist et. al 2014)
    - Relatively complex, non-local basis functions
  - Finite-difference/finite-element method (Stroud et. al 2010)

Possible Approaches

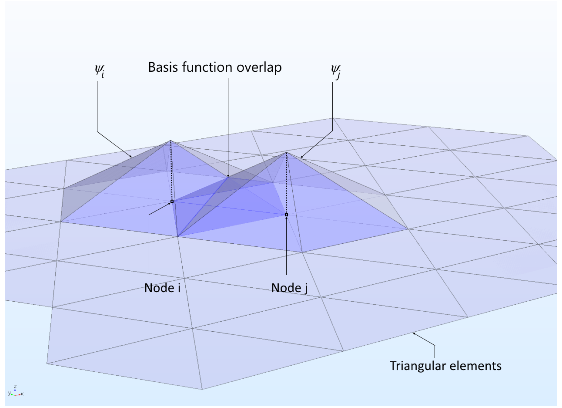


- Basically this is the advection-diffusion equation
  - Possibly with linear decay
- Existing methods
  - Fast Fourier Transform/Kalman filter (Sigrist et. al 2014)
    - Relatively complex, non-local basis functions
  - Finite-difference/finite-element method (Stroud et. al 2010)
    - Comparatively simple, fairly popular

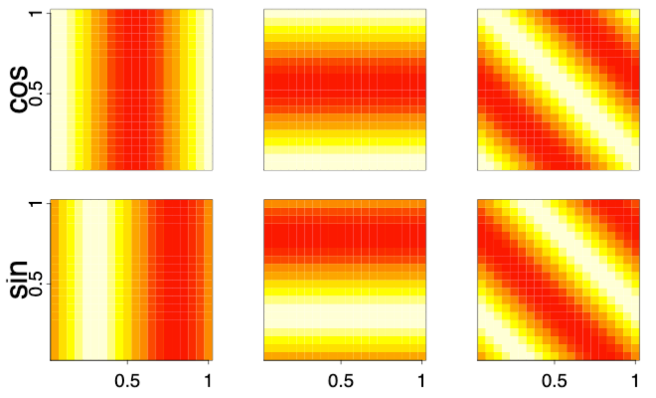
Possible Approaches



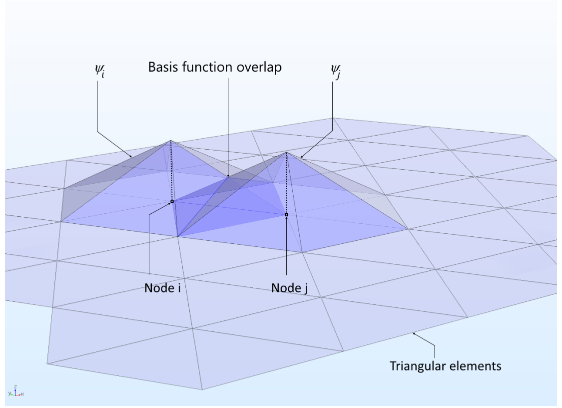
- Basically this is the advection-diffusion equation
  - Possibly with linear decay
- Existing methods
  - Fast Fourier Transform/Kalman filter (Sigrist et. al 2014)
    - Relatively complex, non-local basis functions
  - Finite-difference/finite-element method (Stroud et. al 2010)
    - Comparatively simple, fairly popular
    - Local basis functions



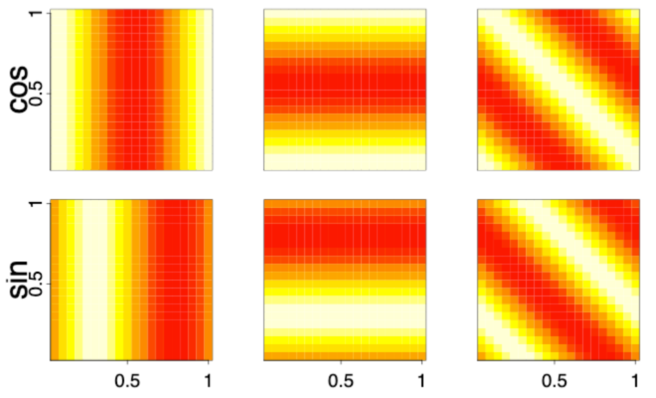
Possible Approaches



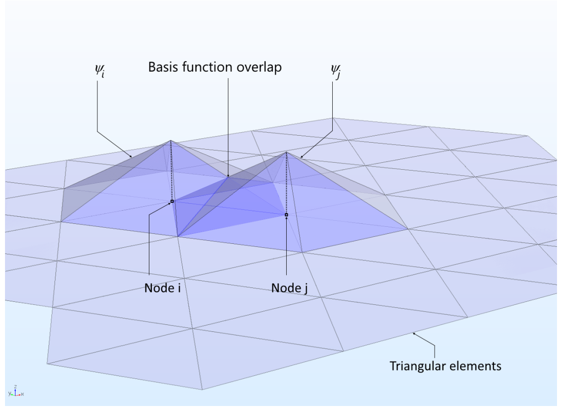
- Basically this is the advection-diffusion equation
    - Possibly with linear decay
- Existing methods
    - Fast Fourier Transform/Kalman filter (Sigrist et. al 2014)
        - Relatively complex, non-local basis functions
    - Finite-difference/finite-element method (Stroud et. al 2010)
        - Comparatively simple, fairly popular
        - Local basis functions
        - Possibly slower



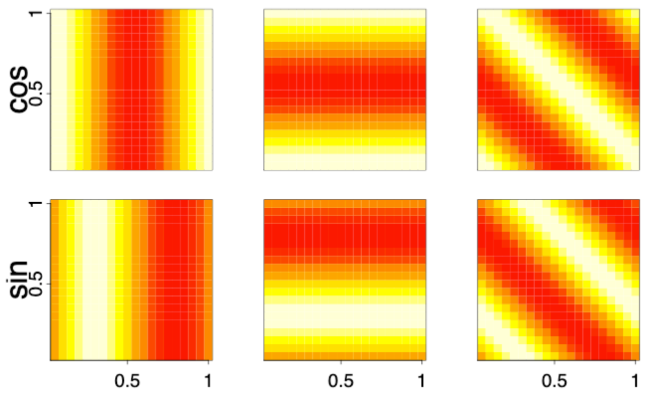
Possible Approaches



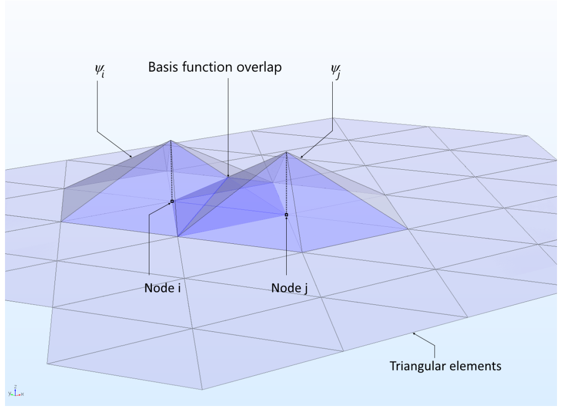
- Basically this is the advection-diffusion equation
    - Possibly with linear decay
- Existing methods
    - Fast Fourier Transform/Kalman filter (Sigrist et. al 2014)
        - Relatively complex, non-local basis functions
    - Finite-difference/finite-element method (Stroud et. al 2010)
        - Comparatively simple, fairly popular
        - Local basis functions
        - Possibly slower
    - Gaussian functional regression (Nguyen and Peraire 2015)



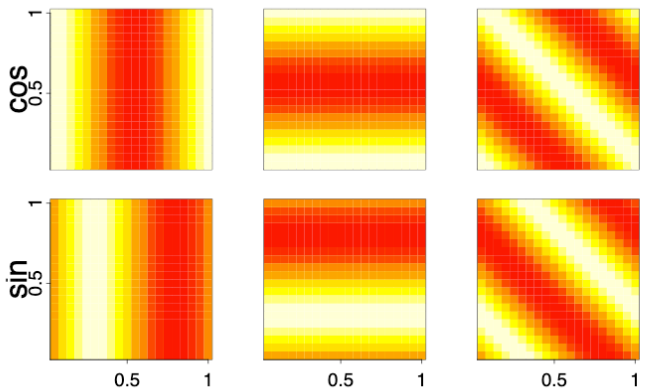
Possible Approaches



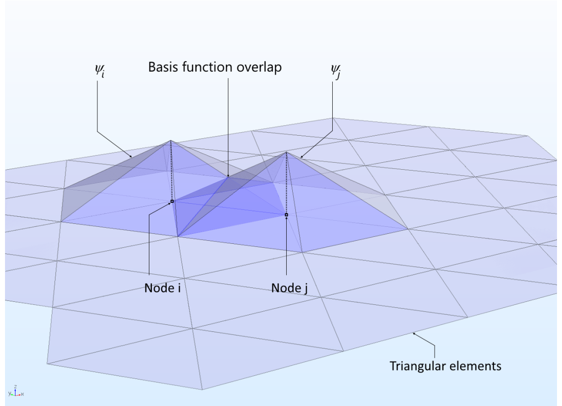
- Basically this is the advection-diffusion equation
  - Possibly with linear decay
- Existing methods
  - Fast Fourier Transform/Kalman filter (Sigrist et. al 2014)
    - Relatively complex, non-local basis functions
  - Finite-difference/finite-element method (Stroud et. al 2010)
    - Comparatively simple, fairly popular
    - Local basis functions
    - Possibly slower
  - Gaussian functional regression (Nguyen and Peraire 2015)
    - Also complex, relatively unknown



Possible Approaches



- Basically this is the advection-diffusion equation
  - Possibly with linear decay
- Existing methods
  - Fast Fourier Transform/Kalman filter (Sigrist et. al 2014)
    - Relatively complex, non-local basis functions
  - Finite-difference/finite-element method (Stroud et. al 2010)
    - Comparatively simple, fairly popular
    - Local basis functions
    - Possibly slower
  - Gaussian functional regression (Nguyen and Peraire 2015)
    - Also complex, relatively unknown
    - Basically a fancy kernel trick



The Math

The Math

- Our model is $\mathcal{L}u = f$

The Math

- Our model is $\mathcal{L}u = f$
- Assume that $\mathcal{L}$ is a linear operator

The Math

- Our model is $\mathcal{L}u = f$
- Assume that $\mathcal{L}$ is a linear operator
- Discretize this as $Ku = Lf$

The Math

- Our model is $\mathcal{L}u = f$
- Assume that $\mathcal{L}$ is a linear operator
- Discretize this as $Ku = Lf$
- We model $f \sim X\beta + Z\gamma + \epsilon$

The Math

- Our model is $\mathcal{L}u = f$
- Assume that $\mathcal{L}$ is a linear operator
- Discretize this as $Ku = Lf$
- We model $f \sim X\beta + Z\gamma + \epsilon$
- If everything is normal then

$$[f|d] \sim \mathrm{MVN}(\mu_{\mathrm{post}}, Q_{\mathrm{post}}^{-1})$$

Pros/Cons

Pros/Cons

- Relatively fast, lots of sparse matrices

Pros/Cons

- Relatively fast, lots of sparse matrices
- Incorporate linear mixed models

Pros/Cons

- Relatively fast, lots of sparse matrices
- Incorporate linear mixed models

But:

Pros/Cons

- Relatively fast, lots of sparse matrices
- Incorporate linear mixed models

But:

- Requires perfectly known physics

Pros/Cons

- Relatively fast, lots of sparse matrices
- Incorporate linear mixed models

But:

- Requires perfectly known physics
- Requires linear PDE

Pros/Cons

- Relatively fast, lots of sparse matrices
- Incorporate linear mixed models

But:

- Requires perfectly known physics
- Requires linear PDE
- Requires normal priors

Space-time distributions

Space-time distributions

- Suppose our math is instead

$$\frac{\partial u}{\partial t} + \mathcal{L}u = f$$

Space-time distributions

- Suppose our math is instead $$\frac{\partial u}{\partial t} + \mathcal{L}u = f$$
- Then we apply a finite-difference in time $$u_{t+1} = (I - K dt)u_t + L f_t dt$$

Space-time distributions

- Suppose our math is instead $\frac{\partial u}{\partial t} + \mathcal{L}u = f$
- Then we apply a finite-difference in time $u_{t+1} = (I - Kdt)u_t + Lf_tdt$
  - This gives us a vector auto-regressive model

Space-time distributions

- Suppose our math is instead $$\frac{\partial u}{\partial t} + \mathcal{L}u = f$$
- Then we apply a finite-difference in time $$u_{t+1} = (I - Kdt)u_t + Lf_tdt$$
  - This gives us a vector auto-regressive model
  - Can construct very large block-Toeplitz covariance matrix

Space-time distributions

- Suppose our math is instead $$\frac{\partial u}{\partial t} + \mathcal{L}u = f$$
- Then we apply a finite-difference in time $$u_{t+1} = (I - Kdt)u_t + Lf_t dt$$
    - This gives us a vector auto-regressive model
    - Can construct very large block-Toeplitz covariance matrix
    - Highly structured sparse precision matrix

Space-time distributions

- Suppose our math is instead $$\frac{\partial u}{\partial t} + \mathcal{L}u = f$$
- Then we apply a finite-difference in time $$u_{t+1} = (I - Kdt)u_t + Lf_t dt$$
  - This gives us a vector auto-regressive model
  - Can construct very large block-Toeplitz covariance matrix
  - Highly structured sparse precision matrix
- Initial condition?

Space-time distributions

- Suppose our math is instead $$\frac{\partial u}{\partial t} + \mathcal{L}u = f$$
- Then we apply a finite-difference in time $$u_{t+1} = (I - Kdt)u_t + Lf_tdt$$
  - This gives us a vector auto-regressive model
  - Can construct very large block-Toeplitz covariance matrix
  - Highly structured sparse precision matrix
- Initial condition?
  - Ideally we want to use the stationary distribution

Space-time distributions

- Suppose our math is instead $\frac{\partial u}{\partial t} + \mathcal{L}u = f$
- Then we apply a finite-difference in time $u_{t+1} = (I - Kdt)u_t + Lf_t dt$
  - This gives us a vector auto-regressive model
  - Can construct very large block-Toeplitz covariance matrix
  - Highly structured sparse precision matrix
- Initial condition?
  - Ideally we want to use the stationary distribution
  - What is the limiting spatial distribution? $u(x, \infty)$

Space-time limiting distributions

Space-time limiting distributions

- We already have $u_{t+1} = (I - Kdt)u_t + Lf_t dt$

Space-time limiting distributions

- We already have $u_{t+1} = (I - K dt) u_t + L f_t dt$
- Then taking some limits we get two possibilities

$$K \Sigma_u + \Sigma_u K^T = L \Sigma_f L^T$$

or

$$K_f \Sigma_{uf} + \Sigma_{uf} K_f^T = L_u \Sigma_{ff}$$
$$K_u \Sigma_{uu} + \Sigma_{uu} K_u^T = L_u \Sigma_{fu} + \Sigma_{uf} L_u^T$$

Space-time limiting distributions

Space-time limiting distributions

- Basically need to solve $$K\Sigma + \Sigma K^T = M$$

Space-time limiting distributions

- Basically need to solve
- But we want to do this with sparsity

$$K\Sigma + \Sigma K^T = M$$

$$QK + K^T Q = QMQ$$

Space-time limiting distributions

- Basically need to solve $$K\Sigma + \Sigma K^T = M$$
- But we want to do this with sparsity $$QK + K^T Q = QMQ$$
- Minimize matrix quadratic form $$(Q - K^T)M(Q - K) - K^T M K = 0$$

Space-time limiting distributions

- Basically need to solve $$K\Sigma + \Sigma K^T = M$$
- But we want to do this with sparsity $$QK + K^T Q = QMQ$$
- Minimize matrix quadratic form $$(Q - K^T)M(Q - K) - K^T M K = 0$$
- Ideas

Space-time limiting distributions

- Basically need to solve $$K\Sigma + \Sigma K^T = M$$
- But we want to do this with sparsity $$QK + K^T Q = QMQ$$
- Minimize matrix quadratic form $$(Q - K^T)M(Q - K) - K^T M K = 0$$
- Ideas
  - quadratic minimization techniques (i.e. conjugate gradient descent)

Space-time limiting distributions

- Basically need to solve $$K\Sigma + \Sigma K^T = M$$
- But we want to do this with sparsity $$QK + K^T Q = QMQ$$
- Minimize matrix quadratic form $$(Q - K^T)M(Q - K) - K^T MK = 0$$
- Ideas
  - quadratic minimization techniques (i.e. conjugate gradient descent)
  - if M is diagonal then the solution is as sparse as $K + K^T$

Space-time limiting distributions

- Basically need to solve $$K\Sigma + \Sigma K^T = M$$
- But we want to do this with sparsity $$QK + K^T Q = QMQ$$
- Minimize matrix quadratic form $$(Q - K^T)M(Q - K) - K^T MK = 0$$
- Ideas
    - quadratic minimization techniques (i.e. conjugate gradient descent)
    - if M is diagonal then the solution is as sparse as $K + K^T$
    - otherwise assume structure is $(K + K^T)^m$

Space-time limiting distributions

- Basically need to solve $$K\Sigma + \Sigma K^T = M$$
- But we want to do this with sparsity $$QK + K^T Q = QMQ$$
- Minimize matrix quadratic form $$(Q - K^T)M(Q - K) - K^T M K = 0$$
- Ideas
  - quadratic minimization techniques (i.e. conjugate gradient descent)
  - if M is diagonal then the solution is as sparse as $K + K^T$
  - otherwise assume structure is $(K + K^T)^m$
  - minimization in the Loewner order???

Stochastic Perturbation Analysis

Stochastic Perturbation Analysis

- Recall that we assume:

Stochastic Perturbation Analysis

- Recall that we assume:
  - perfectly known dynamics

Stochastic Perturbation Analysis

- Recall that we assume:
  - perfectly known dynamics
  - linear dynamics

Stochastic Perturbation Analysis

- Recall that we assume:
  - perfectly known dynamics
  - linear dynamics
- These are strong assumptions

Stochastic Perturbation Analysis

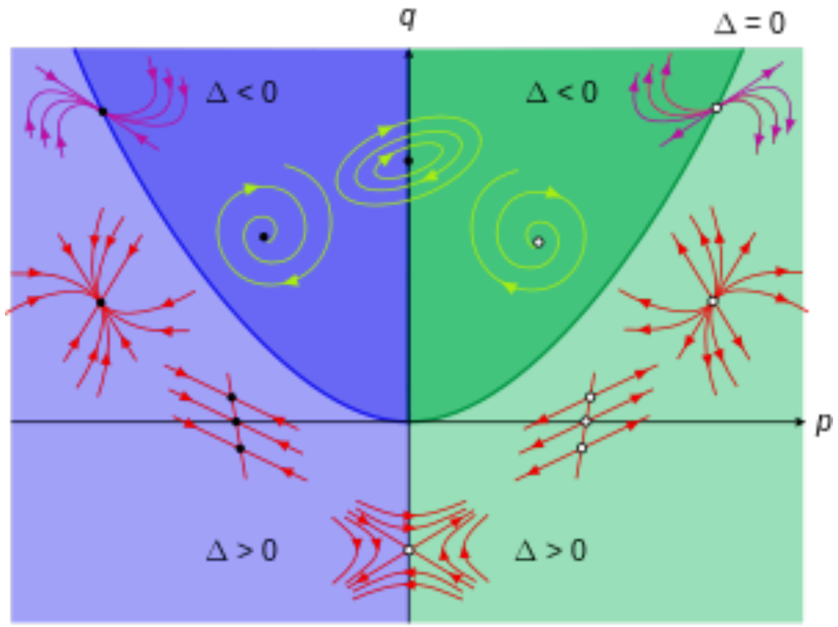
- Recall that we assume:
  - perfectly known dynamics
  - linear dynamics
- These are strong assumptions
  - linearize around a solution to a nonlinear PDE

Stochastic Perturbation Analysis

- Recall that we assume:
  - perfectly known dynamics
  - linear dynamics
- These are strong assumptions
  - linearize around a solution to a nonlinear PDE
- Review: Linearization

Stochastic Perturbation Analysis

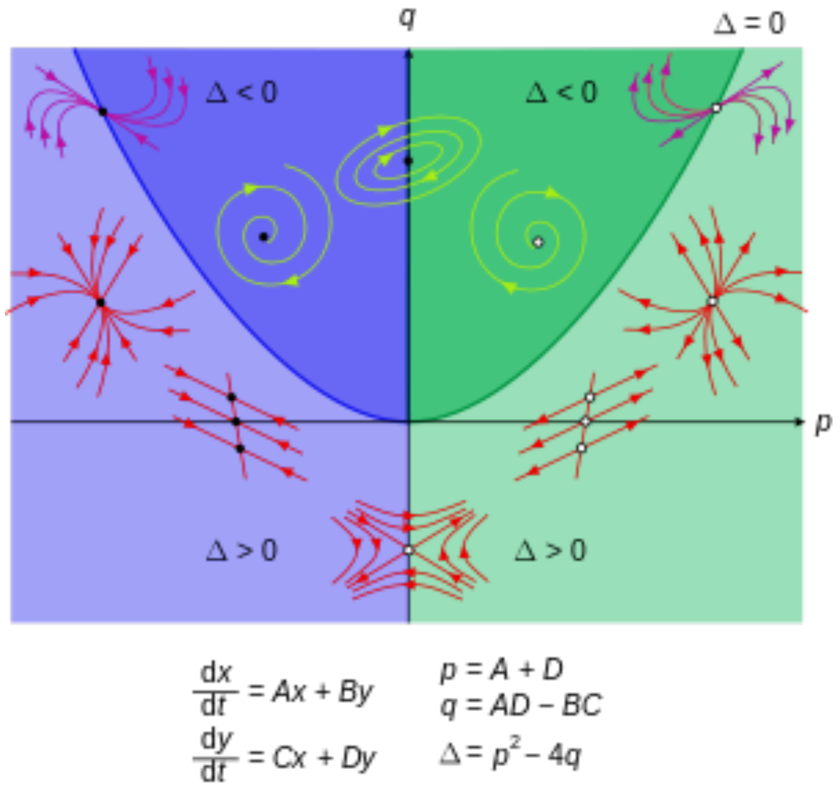
- Recall that we assume:
  - perfectly known dynamics
  - linear dynamics
- These are strong assumptions
  - linearize around a solution to a nonlinear PDE
- Review: Linearization
  - ODEs and linearized fixed points



$$\frac{dx}{dt} = Ax + By \qquad p = A + D$$
$$q = AD - BC$$
$$\frac{dy}{dt} = Cx + Dy \qquad \Delta = p^2 - 4q$$

Stochastic Perturbation Analysis

- Recall that we assume:
  - perfectly known dynamics
  - linear dynamics
- These are strong assumptions
  - linearize around a solution to a nonlinear PDE
- Review: Linearization
  - ODEs and linearized fixed points
  - PDEs can be linearized as well



Case Study

Case Study

- Pollution source inference

Case Study

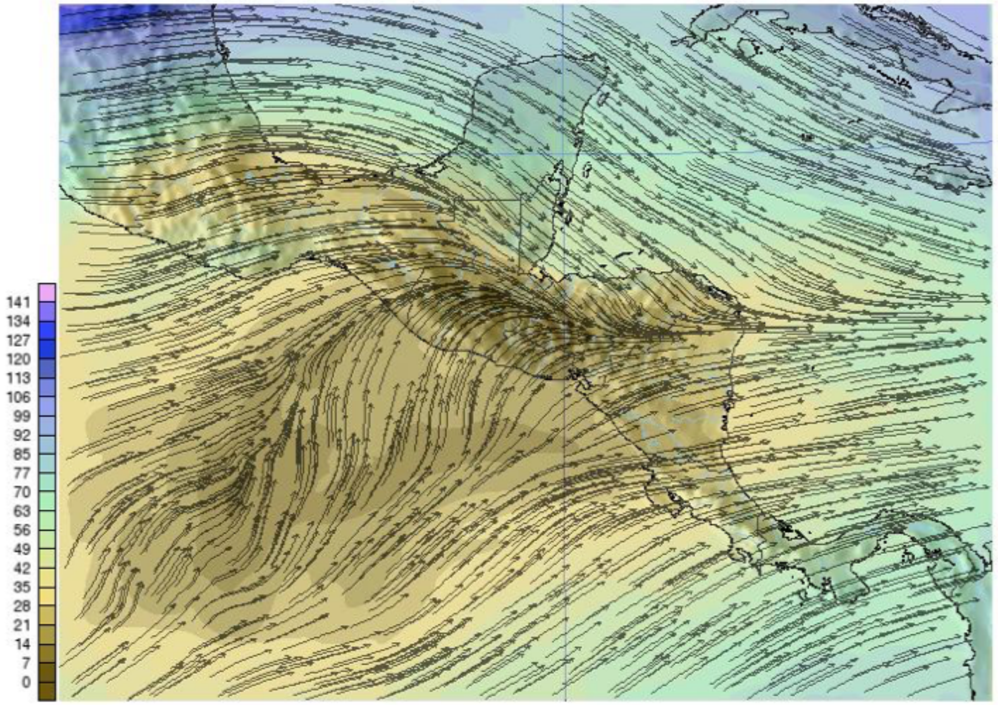
- Pollution source inference
- Data available for:

Case Study

- Pollution source inference
- Data available for:
  - Source concentration

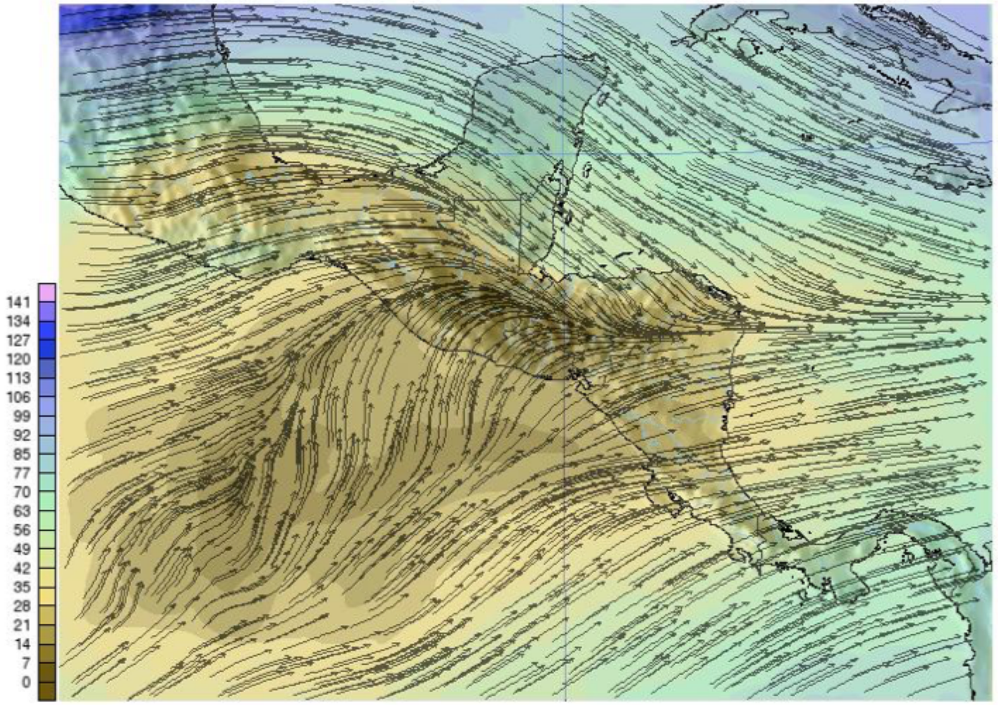
Case Study

- Pollution source inference
- Data available for:
  - Source concentration
  - Wind speed/pressure/etc



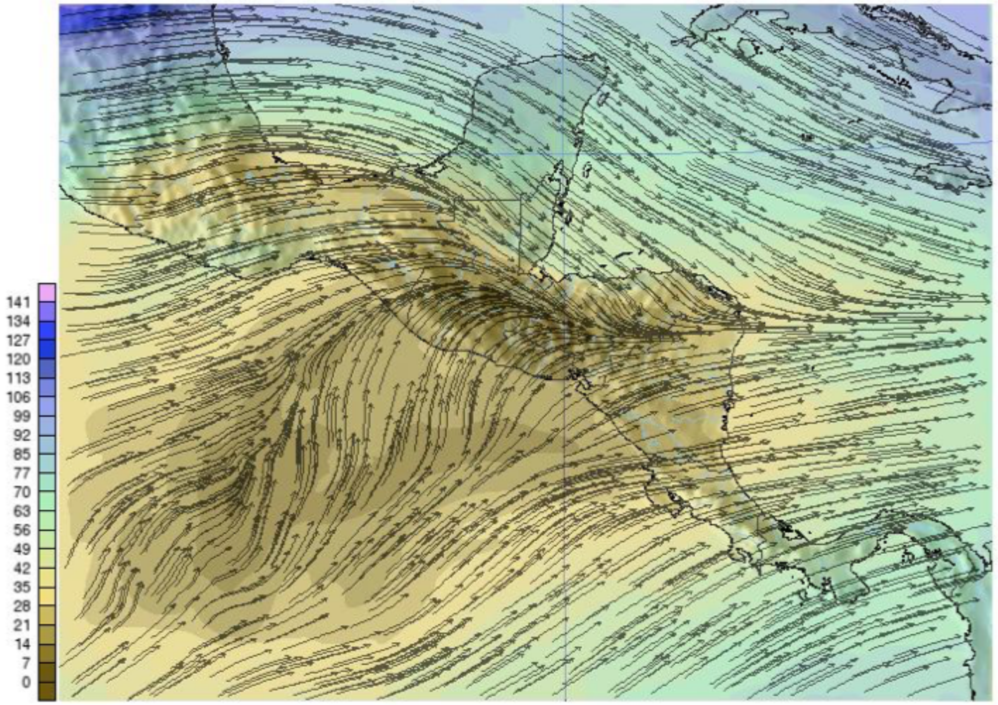
Case Study

- Pollution source inference
- Data available for:
    - Source concentration
    - Wind speed/pressure/etc
- Wind speed governed by Navier-Stokes equations

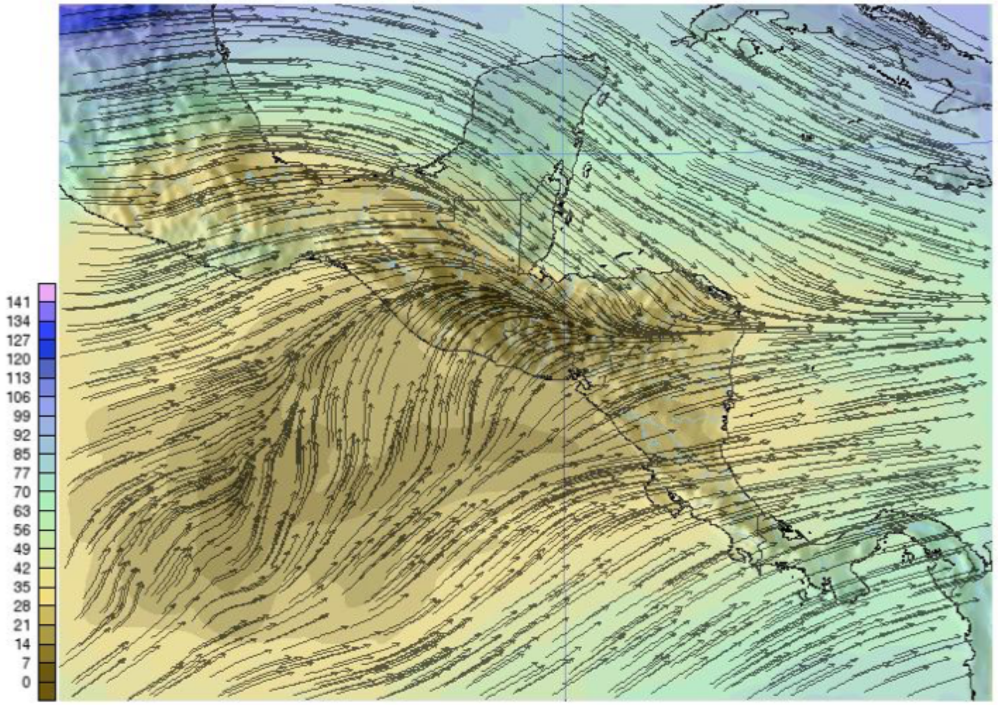


Case Study

- Pollution source inference
- Data available for:
  - Source concentration
  - Wind speed/pressure/etc
- Wind speed governed by Navier-Stokes equations
  - Highly nonlinear

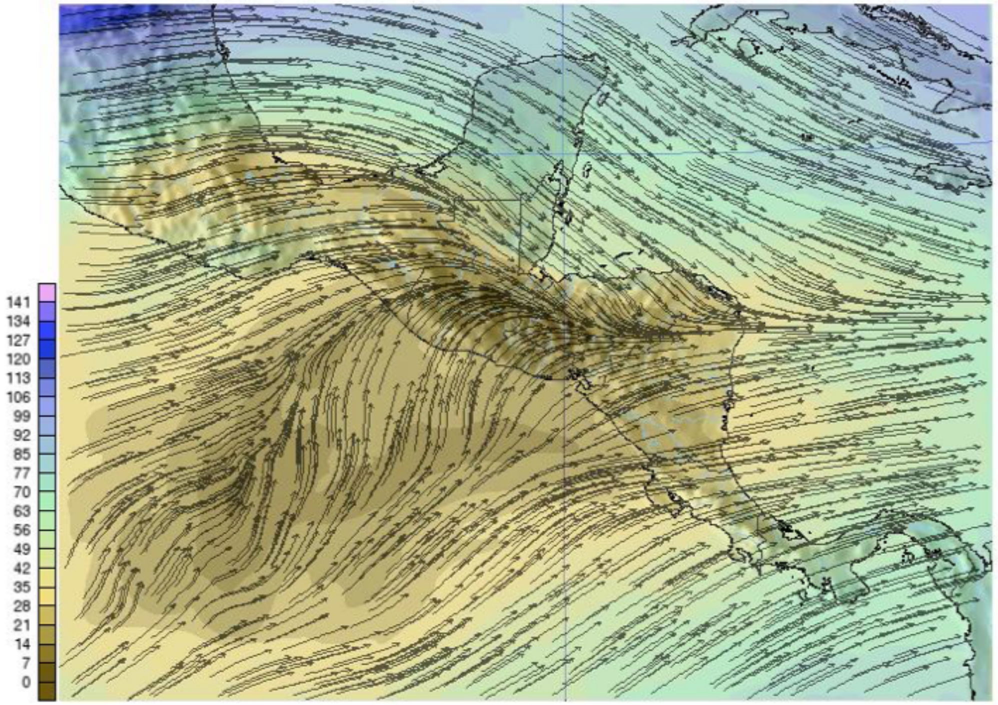


Case Study



- Pollution source inference
- Data available for:
  - Source concentration
  - Wind speed/pressure/etc
- Wind speed governed by Navier-Stokes equations
  - Highly nonlinear
- But suppose we have enough data for a really good guess

Case Study



- Pollution source inference
- Data available for:
  - Source concentration
  - Wind speed/pressure/etc
- Wind speed governed by Navier-Stokes equations
  - Highly nonlinear
- But suppose we have enough data for a really good guess
  - Just a little variation around it

The New Math

The New Math

$$\mathcal{N}\theta = 0$$

- Our model is

$$\mathcal{L}[\theta]u = f$$

The New Math

$$\mathcal{N}\theta = 0$$

- Our model is $$\mathcal{L}[\theta]u = f$$
- Assume that $\mathcal{L}[\theta]$ is a linear operator

The New Math

$$\mathcal{N}\theta = 0$$

$$\mathcal{L}[\theta]u = f$$

- Our model is
- Assume that $\mathcal{L}[\theta]$ is a linear operator
- Assume we have a really good guess θ^*

The New Math

$$\mathcal{N}\theta = 0$$

$$\mathcal{L}[\theta]u = f$$

- Our model is
- Assume that $\mathcal{L}[\theta]$ is a linear operator
- Assume we have a really good guess θ^*
- Then we conduct inference as usual, get GP u^*

The New Math

$$\mathcal{N}\theta = 0$$

- Our model is $\mathcal{L}[\theta]u = f$
- Assume that $\mathcal{L}[\theta]$ is a linear operator
- Assume we have a really good guess θ^*
- Then we conduct inference as usual, get GP u^*
- Then we might model small errors in θ as

$$\mathcal{N}(\theta^* + \Delta_\theta) = \eta$$

$$\mathcal{L}[\theta^* + \Delta_\theta](u^* + \Delta_u) = f$$

Linearization

Linearization

$$\mathcal{N}(\theta^* + \Delta_\theta) = \eta$$

$$\mathcal{L}[\theta^* + \Delta_\theta](u^* + \Delta_u) = f$$

- Now we linearize everything to go from $\Uparrow$ to $\Downarrow$

$$\partial\mathcal{N}[\theta^*]\Delta_\theta = \eta$$

$$\mathcal{L}[\theta^*]\Delta_u + \mathcal{L}[\Delta_\theta]u^* = f$$

Linearization

$$\mathcal{N}(\theta^* + \Delta_\theta) = \eta$$
$$\mathcal{L}[\theta^* + \Delta_\theta](u^* + \Delta_u) = f$$

- Now we linearize everything to go from ⬆ to ⬇

$$\partial\mathcal{N}[\theta^*]\Delta_\theta = \eta$$
$$\mathcal{L}[\theta^*]\Delta_u + \mathcal{L}[\Delta_\theta]u^* = f$$

- This gives us a fully linear system of PDEs

Linearization

$$\mathcal{N}(\theta^* + \Delta_\theta) = \eta$$

$$\mathcal{L}[\theta^* + \Delta_\theta](u^* + \Delta_u) = f$$

- Now we linearize everything to go from ⬆ to ⬇

$$\partial\mathcal{N}[\theta^*]\Delta_\theta = \eta$$

$$\mathcal{L}[\theta^*]\Delta_u + \mathcal{L}[\Delta_\theta]u^* = f$$

- This gives us a fully linear system of PDEs
- However: since u^* and Δ_θ are both GPs, their product is not

Linearization

$$\mathcal{N}(\theta^* + \Delta_\theta) = \eta$$

$$\mathcal{L}[\theta^* + \Delta_\theta](u^* + \Delta_u) = f$$

- Now we linearize everything to go from ⬆ to ⬇

$$\partial\mathcal{N}[\theta^*]\Delta_\theta = \eta$$

$$\mathcal{L}[\theta^*]\Delta_u + \mathcal{L}[\Delta_\theta]u^* = f$$

- This gives us a fully linear system of PDEs
- However: since u^* and Δ_θ are both GPs, their product is not
- Thus there is no simple analytical solution for Δ_u

Keeping everything MVN

Keeping everything MVN

- Luckily we are already in “perturbation land”

Keeping everything MVN

- Luckily we are already in “perturbation land”
- We may assume Δ_θ is small, and mean zero

Keeping everything MVN

- Luckily we are already in “perturbation land”
- We may assume Δ_θ is small, and mean zero
- Then using a Taylor series expansion of the characteristic function

$$\Delta_u \sim \mathrm{MVN}(0, \Sigma_{\Delta_\theta} \circ (\Sigma_{u^*} + \mu_{u^*}\mu_{u^*}^T))$$

Keeping everything MVN

- Luckily we are already in "perturbation land"
- We may assume Δ_θ is small, and mean zero
- Then using a Taylor series expansion of the characteristic function

$$\Delta_u \sim \mathrm{MVN}(0, \Sigma_{\Delta_\theta} \circ (\Sigma_{u^*} + \mu_{u^*}\mu_{u^*}^T))$$

- Challenge:
  - Hadamard (element-wise) product for precision matrices?

Back to non-normal distributions

Back to non-normal distributions

- Perturbation may not be small enough

Back to non-normal distributions

- Perturbation may not be small enough
- Use smart MVN proposal based on perturbation for MCMC

Back to non-normal distributions

- Perturbation may not be small enough
- Use smart MVN proposal based on perturbation for MCMC
    - Rejection sampling? How to quantify tails?

Back to non-normal distributions

- Perturbation may not be small enough
- Use smart MVN proposal based on perturbation for MCMC
  - Rejection sampling? How to quantify tails?
- Perturbation is inappropriate for forecasting

Back to non-normal distributions

- Perturbation may not be small enough
- Use smart MVN proposal based on perturbation for MCMC
  - Rejection sampling? How to quantify tails?
- Perturbation is inappropriate for forecasting
  - Error blows up quickly due to nonlinearity

Back to non-normal distributions

- Perturbation may not be small enough
- Use smart MVN proposal based on perturbation for MCMC
  - Rejection sampling? How to quantify tails?
- Perturbation is inappropriate for forecasting
  - Error blows up quickly due to nonlinearity
  - Solution: interpolate with perturbation, then simulate

Applications

Applications

- Linear PDEs:

Applications

- Linear PDEs:
  - Advection-diffusion

Applications

- Linear PDEs:
  - Advection-diffusion
    - Pollution

Applications

- Linear PDEs:
  - Advection-diffusion
    - Pollution
    - eDNA

Applications

- Linear PDEs:
  - Advection-diffusion
    - Pollution
    - eDNA
  - Wave equation

Applications

- Linear PDEs:
    - Advection-diffusion
        - Pollution
        - eDNA
    - Wave equation
        - earthquake mapping?

Applications

- Linear PDEs:
    - Advection-diffusion
        - Pollution
        - eDNA
    - Wave equation
        - earthquake mapping?
    - Maxwell's equations

Applications

- Linear PDEs:
    - Advection-diffusion
        - Pollution
        - eDNA
    - Wave equation
        - earthquake mapping?
    - Maxwell's equations
        - charge mapping?

Applications

- Linear PDEs:
  - Advection-diffusion
    - Pollution
    - eDNA
  - Wave equation
    - earthquake mapping?
  - Maxwell's equations
    - charge mapping?
- Nonlinear PDEs:

Applications

- Linear PDEs:
  - Advection-diffusion
    - Pollution
    - eDNA
  - Wave equation
    - earthquake mapping?
  - Maxwell’s equations
    - charge mapping?
- Nonlinear PDEs:
  - Navier Stokes

Applications

- Linear PDEs:
  - Advection-diffusion
    - Pollution
    - eDNA
  - Wave equation
    - earthquake mapping?
  - Maxwell’s equations
    - charge mapping?
- Nonlinear PDEs:
  - Navier Stokes
    - Oceanographic processes

Applications

- Linear PDEs:
  - Advection-diffusion
    - Pollution
    - eDNA
  - Wave equation
    - earthquake mapping?
  - Maxwell's equations
    - charge mapping?
- Nonlinear PDEs:
  - Navier Stokes
    - Oceanographic processes
    - Atmospheric processes

Applications

- Linear PDEs:
  - Advection-diffusion
    - Pollution
    - eDNA
  - Wave equation
    - earthquake mapping?
  - Maxwell’s equations
    - charge mapping?
- Nonlinear PDEs:
  - Navier Stokes
    - Oceanographic processes
    - Atmospheric processes
    - Internal medicine?

Applications

- Linear PDEs:
  - Advection-diffusion
    - Pollution
    - eDNA
  - Wave equation
    - earthquake mapping?
  - Maxwell’s equations
    - charge mapping?

- Nonlinear PDEs:
  - Navier Stokes
    - Oceanographic processes
    - Atmospheric processes
    - Internal medicine?
  - Advection-diffusion-reaction

Applications

- Linear PDEs:
  - Advection-diffusion
    - Pollution
    - eDNA
  - Wave equation
    - earthquake mapping?
  - Maxwell's equations
    - charge mapping?

- Nonlinear PDEs:
  - Navier Stokes
    - Oceanographic processes
    - Atmospheric processes
    - Internal medicine?
  - Advection-diffusion-reaction
    - Ecology, chemistry

Applications

- Linear PDEs:
  - Advection-diffusion
    - Pollution
    - eDNA
  - Wave equation
    - earthquake mapping?
  - Maxwell’s equations
    - charge mapping?

- Nonlinear PDEs:
  - Navier Stokes
    - Oceanographic processes
    - Atmospheric processes
    - Internal medicine?
  - Advection-diffusion-reaction
    - Ecology, chemistry
  - Mechanical engineering

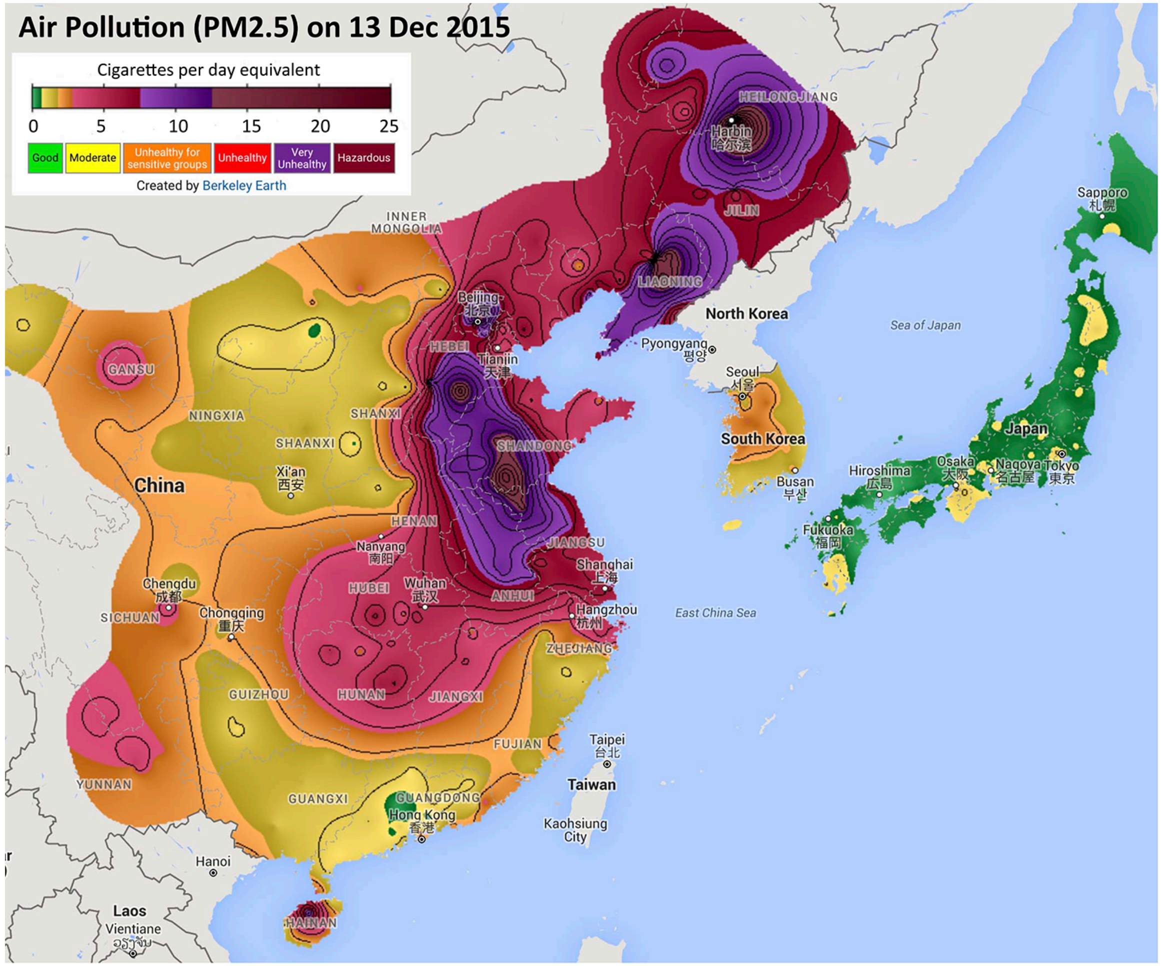
Acknowledgements

Mevin Hooten

Jim Thorson

Andrew Berdahl and Berdahl Lab

QERM and SAFS



This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1762114. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.