Ensemble Data Assimilation on a Non-Conservative Adaptive Mesh

Colin Guider

C.K.R.T. Jones

THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL

Alberto Carrassi and Ali Aydogdu

NERSC

Outline

1. Motivation – a dynamical/thermodynamical sea ice model (neXtSIM)

2. Challenges in developing a suitable ensemble-based data assimilation method

3. Development of algorithm and results for 1-dimensional example

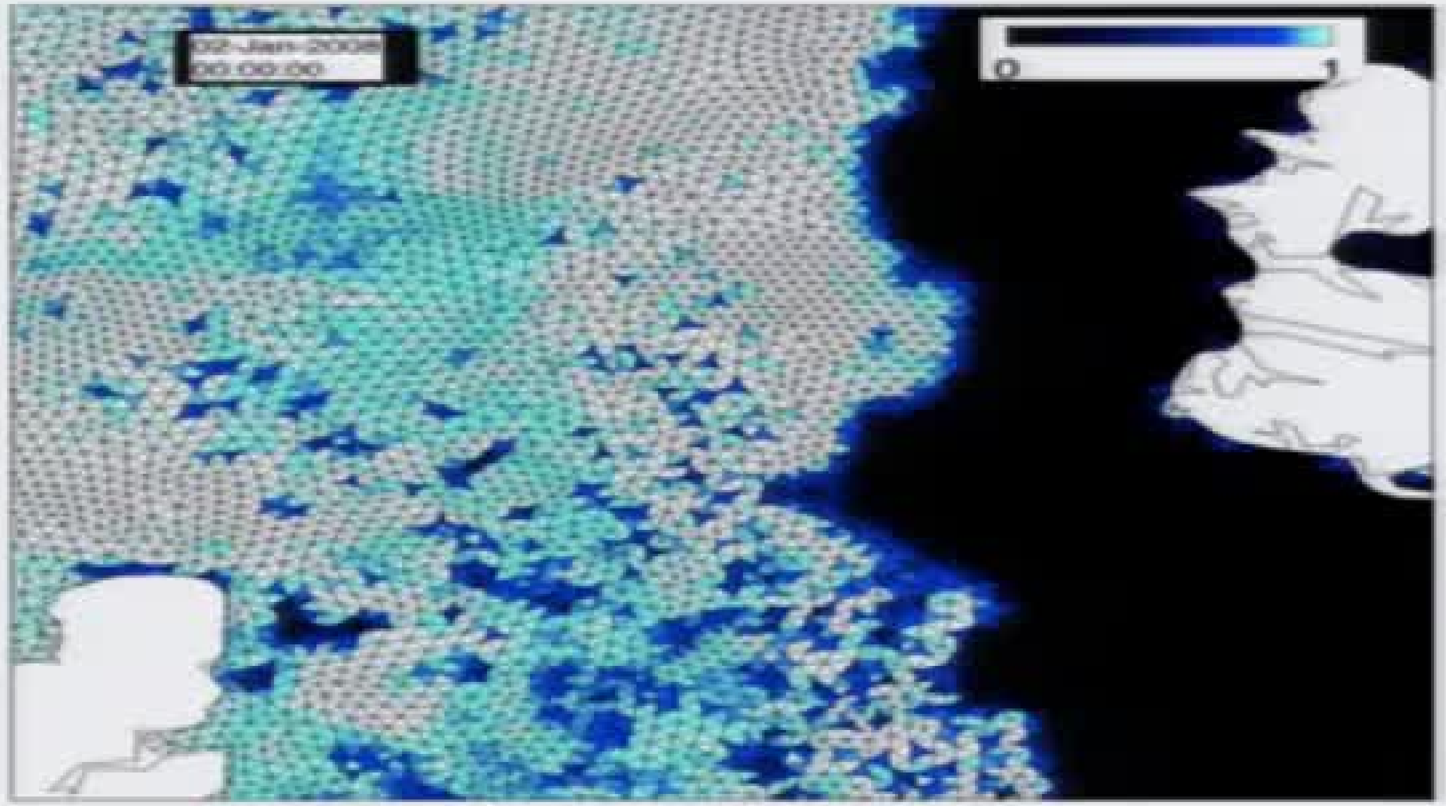
The neXtSIM Model

Variables

- Center variables
  - Ice thickness h and snow thickness h_s
  - Ice concentration A
  - Ice damage d
  - Internal stress tensor σ
- Nodal variables
  - Sea ice velocity **u**

Methodology

- Lagrangian model running on triangular, unstructured mesh
- When mesh becomes too distorted, a remeshing process



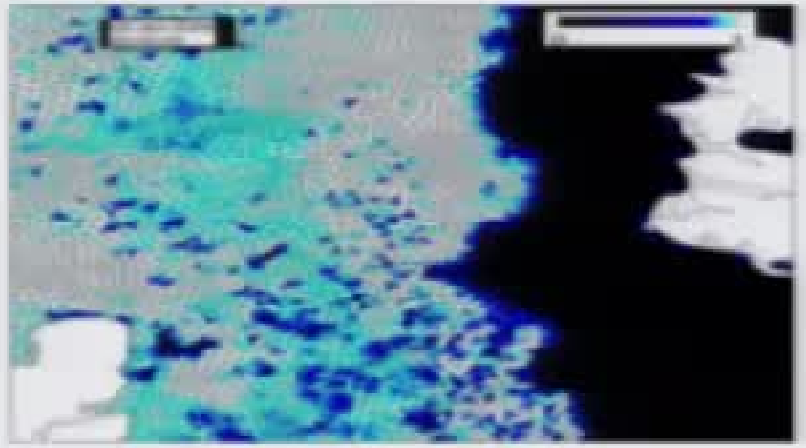
The neXtSIM Model

Variables

- ➢ Center variables
  - Ice thickness h and snow thickness h_s
  - Ice concentration A
  - Ice damage d
  - Internal stress tensor σ
- ➢ Nodal variables
  - Sea ice velocity $\mathbf{u}$

Methodology

- ➢ Lagrangian model running on triangular, unstructured mesh
- ➢ When mesh becomes too distorted, a remeshing process



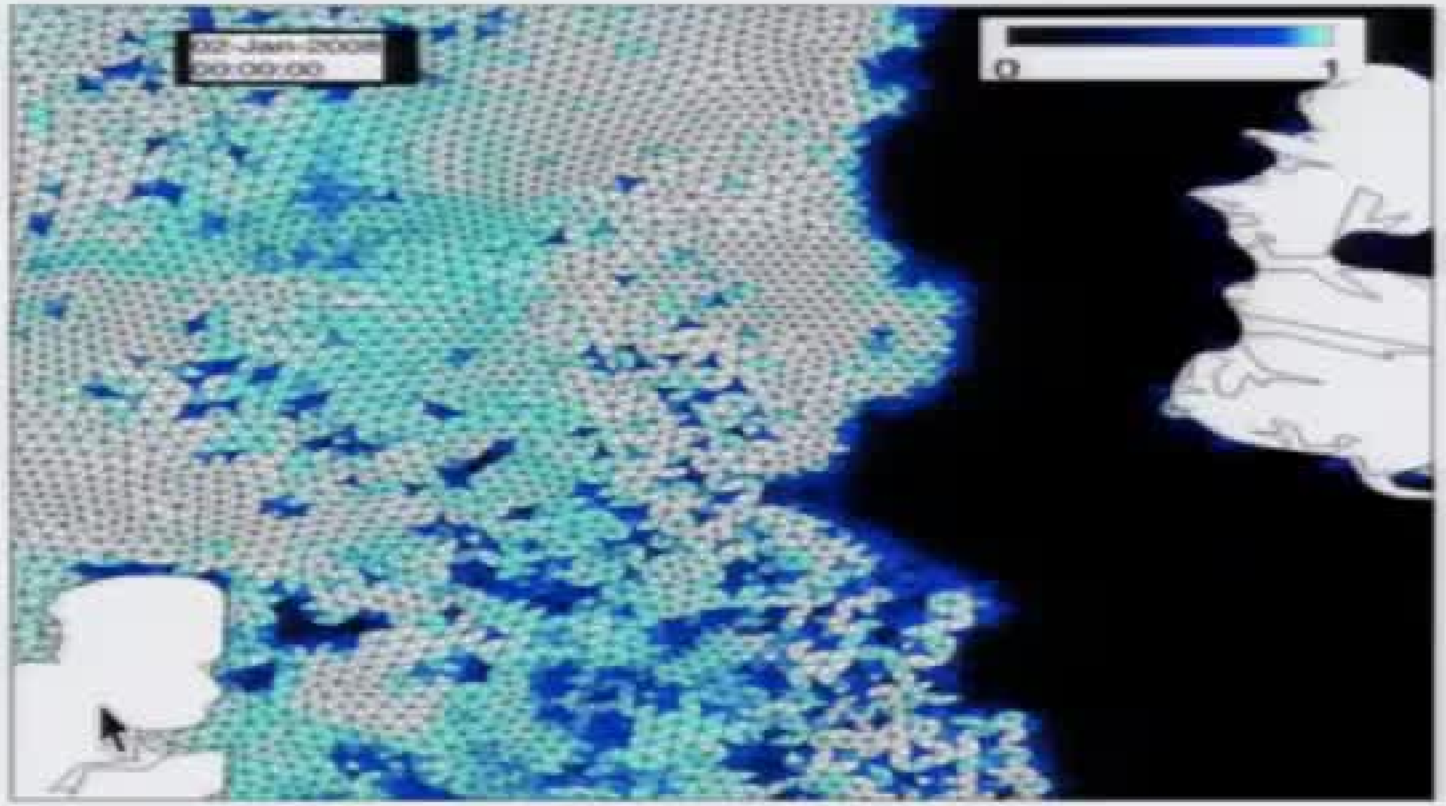
The neXtSIM Model

Variables

- Center variables
  - Ice thickness h and snow thickness h_s
  - Ice concentration A
  - Ice damage d
  - Internal stress tensor σ
- Nodal variables
  - Sea ice velocity $\mathbf{u}$

Methodology

- Lagrangian model running on triangular, unstructured mesh
- When mesh becomes too distorted, a remeshing process



Data Assimilation with neXtSIM

How do we implement an ensemble-based method (EnKF-like) method for neXtSIM?

Mesh-based Challenges

- Each ensemble member will have its own adaptive mesh
- Remeshing occurs independently for each ensemble member, and may occur at different times
- Ensemble meshes will have different sizes – how can we compute ensemble statistics?

Observation-based Challenges

- Combination of satellite observations and in-situ observations
- How do we incorporate Eulerian and Lagrangian observations in same model?
- How will we assimilate different types of observations at different times?

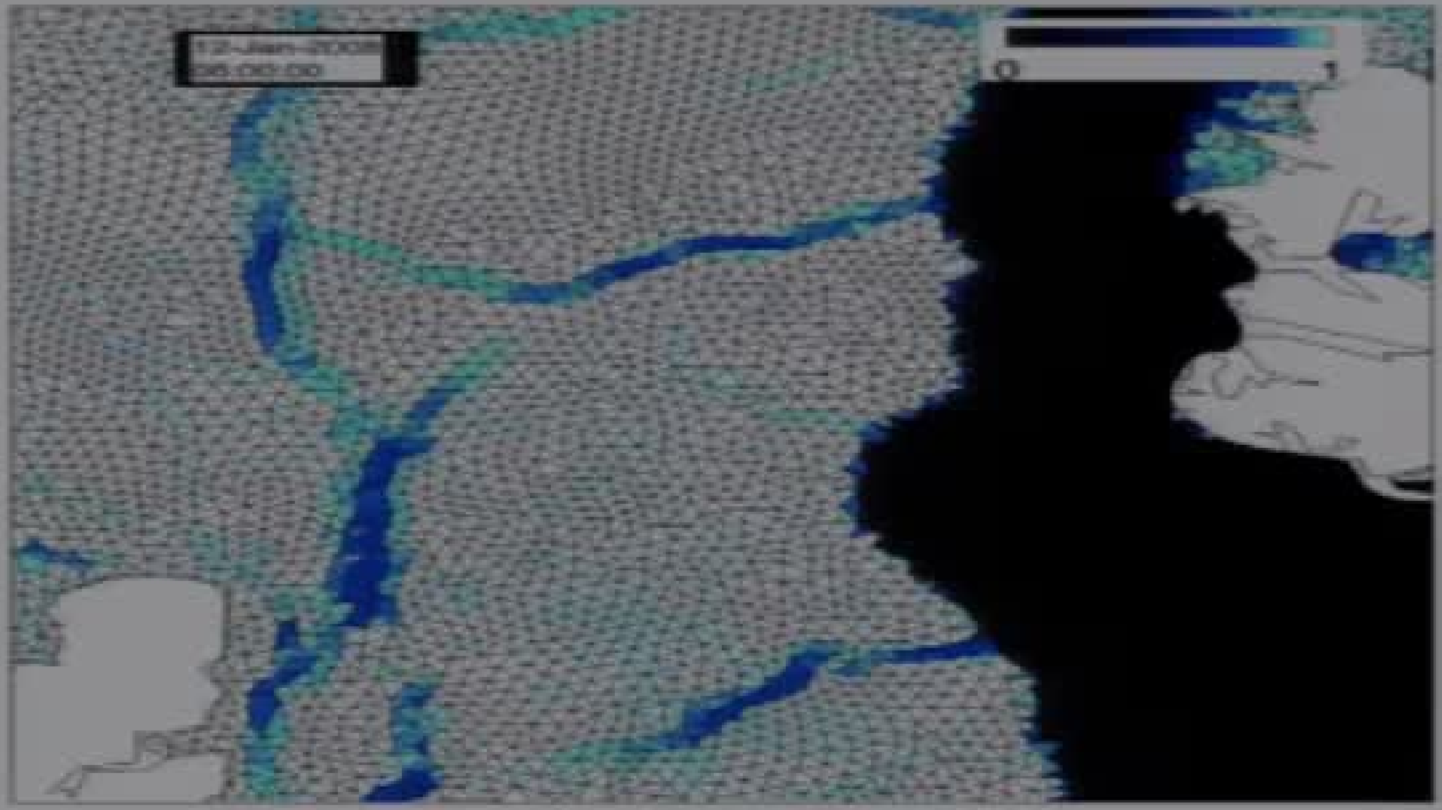
The neXtSIM Model

Variables

- Center variables
  - Ice thickness h and snow thickness h_s
  - Ice concentration A
  - Ice damage d
  - Internal stress tensor σ
- Nodal variables
  - Sea ice velocity $\mathbf{u}$

Methodology

- Lagrangian model running on triangular, unstructured mesh
- When mesh becomes too distorted, a remeshing process



Play from Start | Play from Current Slide | Presenter View | Custom Show | Set Up Slide Show | Hide Slide | Rehearse Timings | Record Slide Show

☑ Use Timings
☑ Show Media Controls

Harty_SIAM_UQ.pptx
this_file.pptx

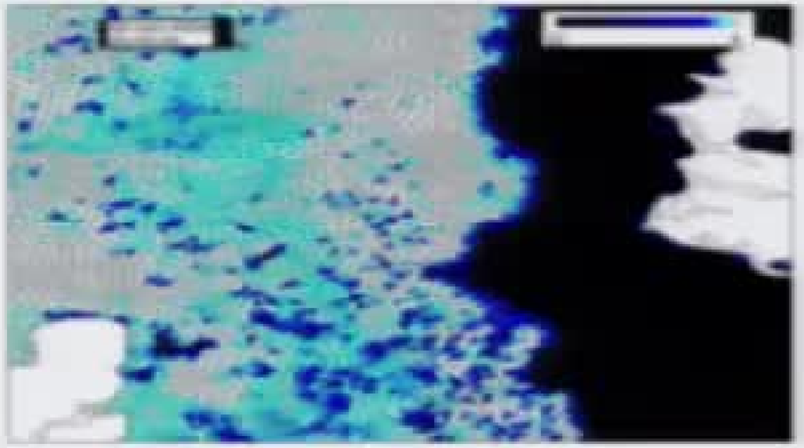
The neXtSIM Model

Variables

- Center variables
  - Ice thickness h and snow thickness h_s
  - Ice concentration A
  - Ice damage d
  - Internal stress tensor σ
- Nodal variables
  - Sea ice velocity $\mathbf{u}$

Methodology

- Lagrangian model running on triangular, unstructured mesh
- When mesh becomes too distorted, a remeshing process



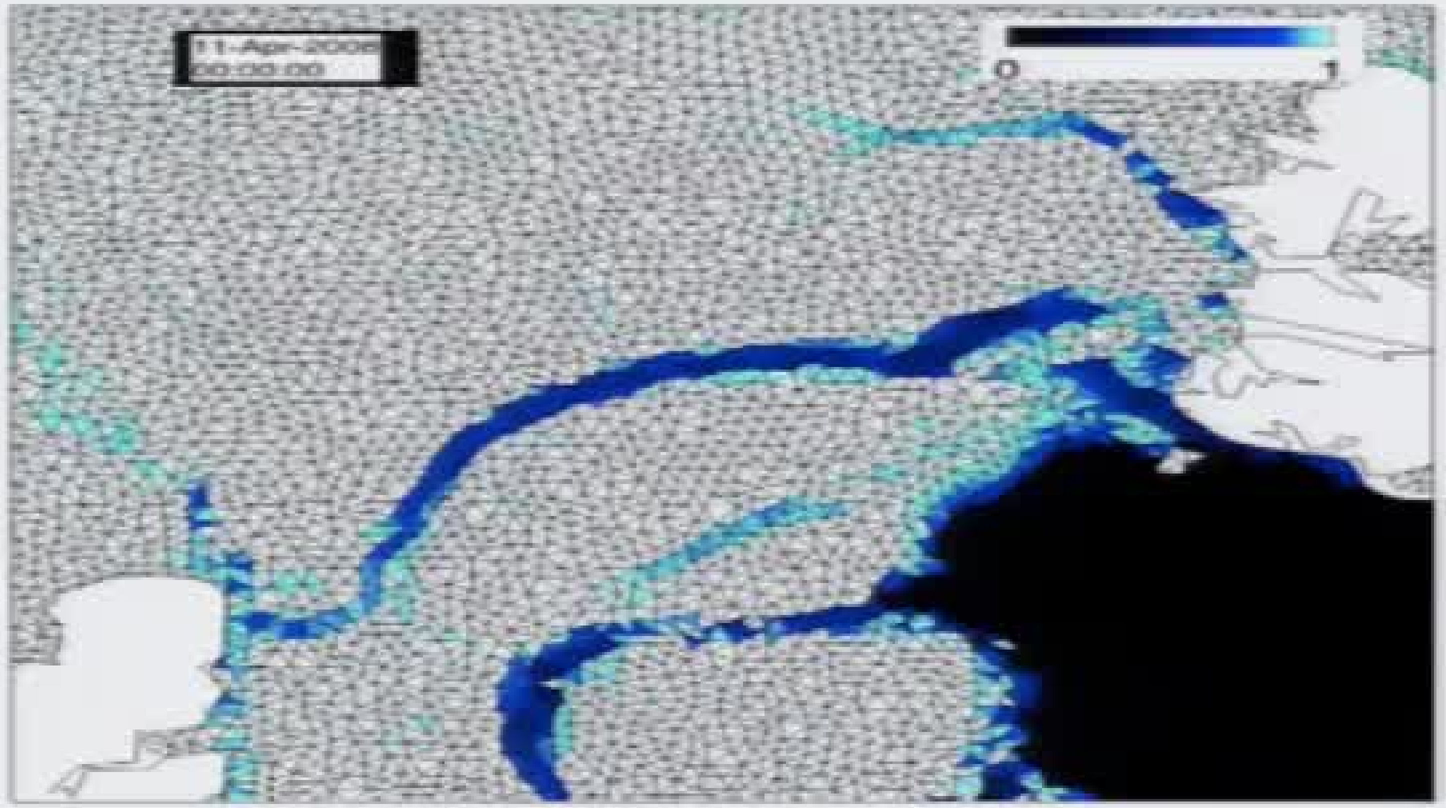
The neXtSIM Model

Variables

- ➢ Center variables
  - Ice thickness h and snow thickness h_s
  - Ice concentration A
  - Ice damage d
  - Internal stress tensor σ
- ➢ Nodal variables
  - Sea ice velocity $\mathbf{u}$

Methodology

- ➢ Lagrangian model running on triangular, unstructured mesh
- ➢ When mesh becomes too distorted, a remeshing process



Data Assimilation with neXtSIM

How do we implement an ensemble-based method (EnKF-like) method for neXtSIM?

Mesh-based Challenges

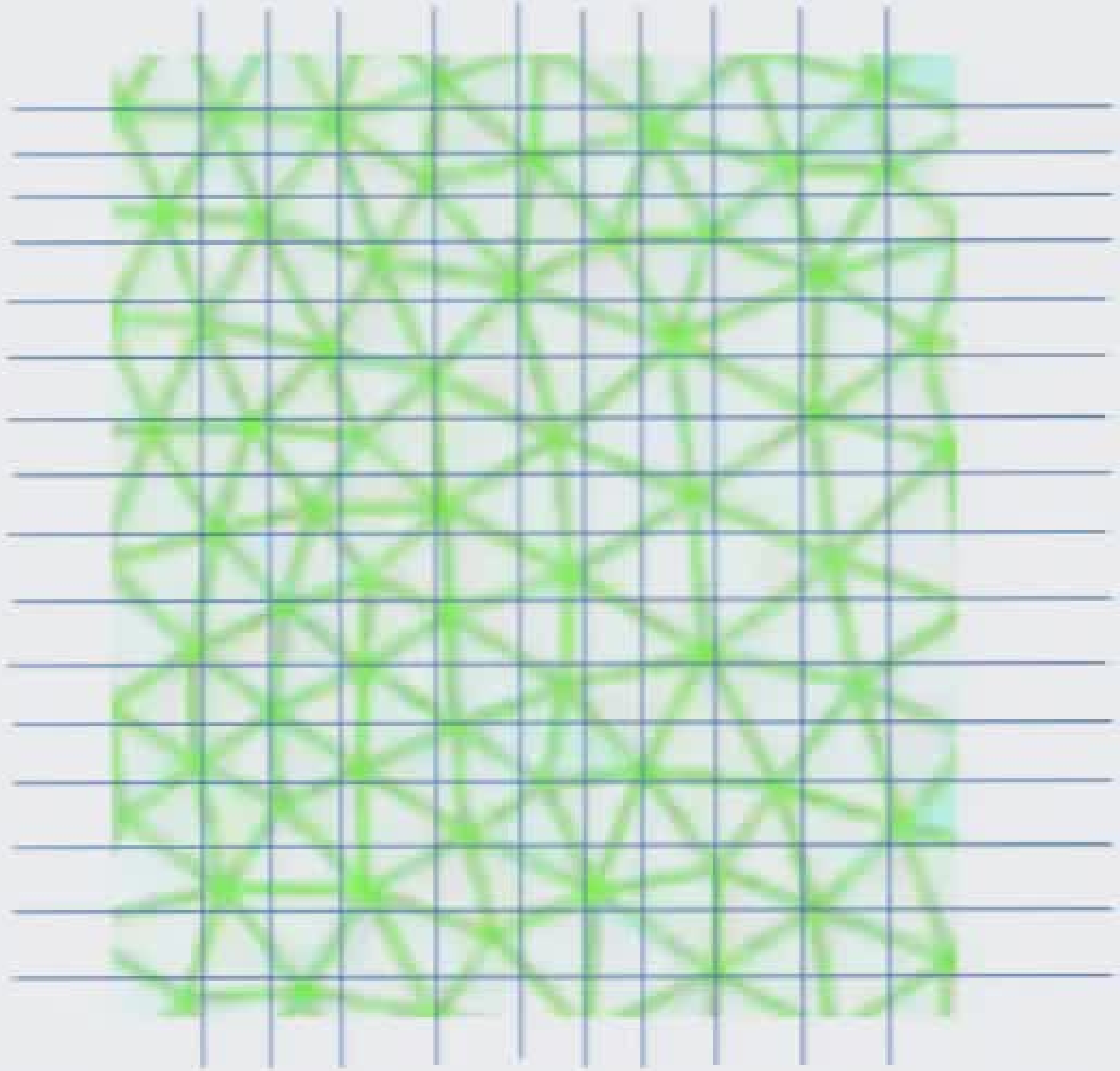
- Each ensemble member will have its own adaptive mesh
- Remeshing occurs independently for each ensemble member, and may occur at different times
- Ensemble meshes will have different sizes – how can we compute ensemble statistics?

Observation-based Challenges

- Combination of satellite observations and in-situ observations
- How do we incorporate Eulerian and Lagrangian observations in same model?
- How will we assimilate different types of observations at different times?

Proposed Solution: Super-meshing

- At analysis time, project each ensemble mesh onto a fixed "supermesh"
- At most one vertex in each mesh box - fill in empty boxes by interpolating
- Perform the analysis on this filled-in supermesh
- The supermesh <u>can be fixed in time</u> or <u>change at each analysis time</u>



A 1-Dimensional Example

We consider the dissipative form of Burgers' equation

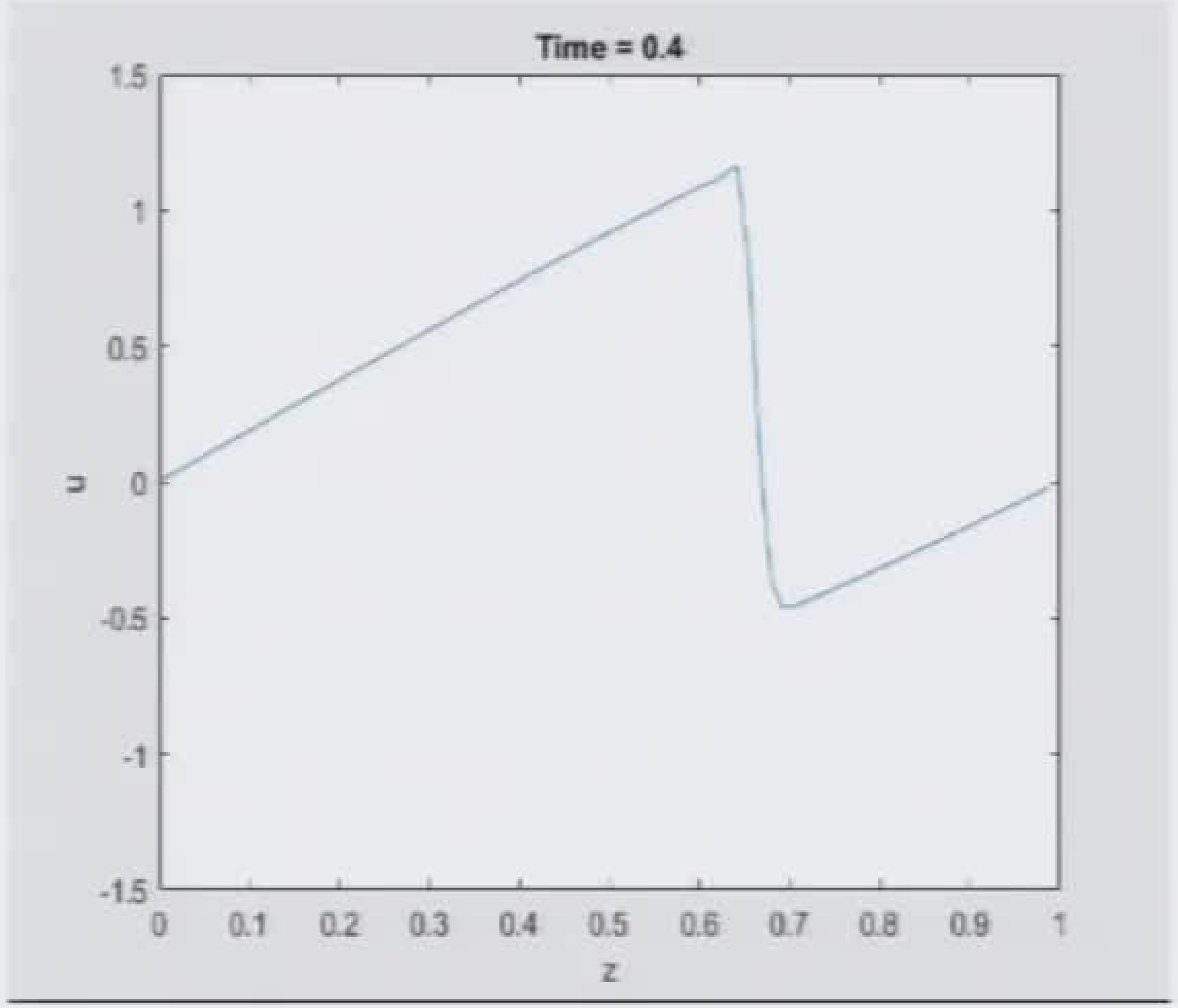
$$u_t = \varepsilon u_{zz} - u u_z$$

for $z \in [0,1]$ with periodic boundary conditions and initial condition

$$u(0,z) = \sin 2\pi z + \frac{1}{2}\sin \pi z.$$

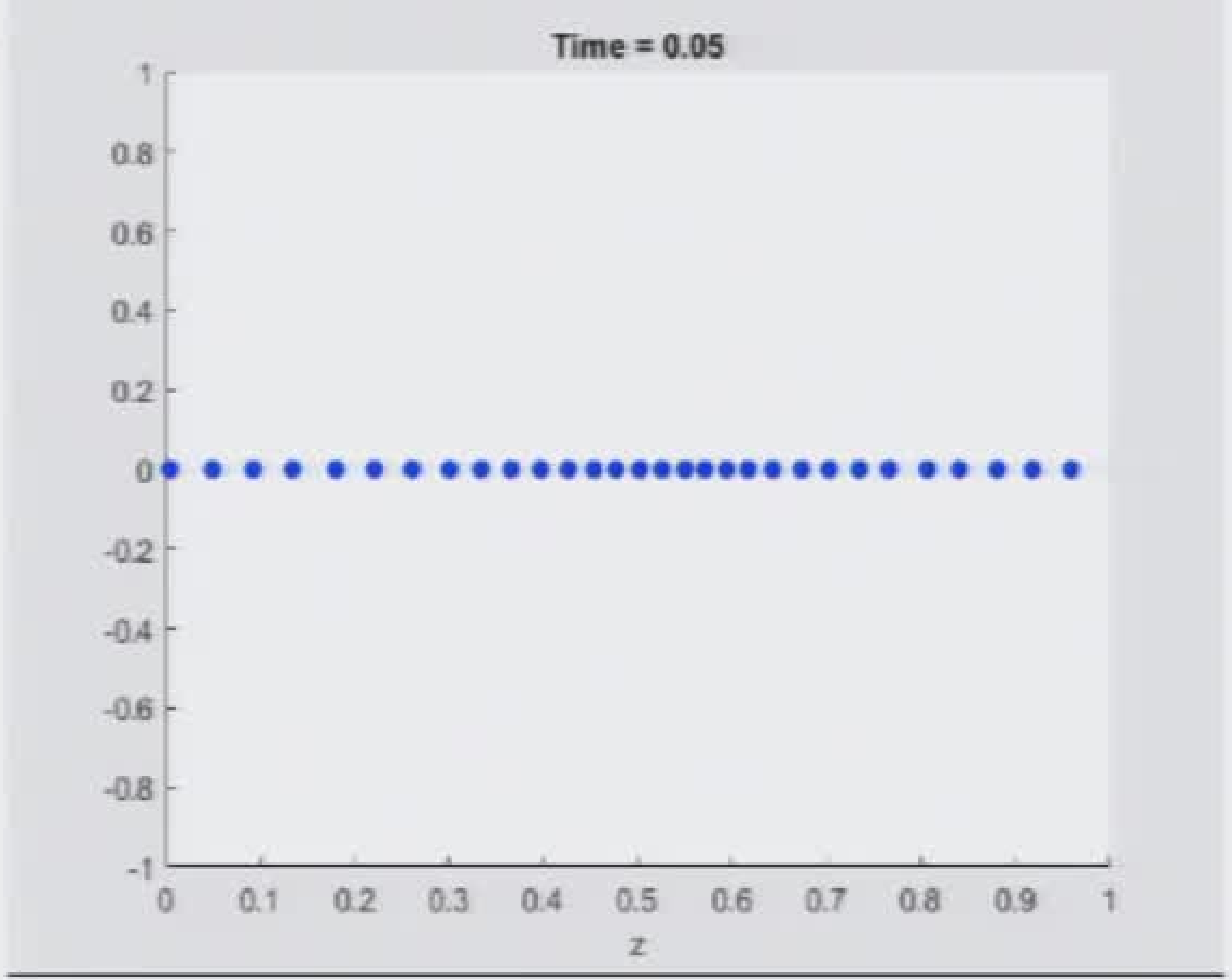
We take the parameter $\epsilon = 0.005$.

Our goal is to estimate u at various points in time and space, given noisy observations of u.



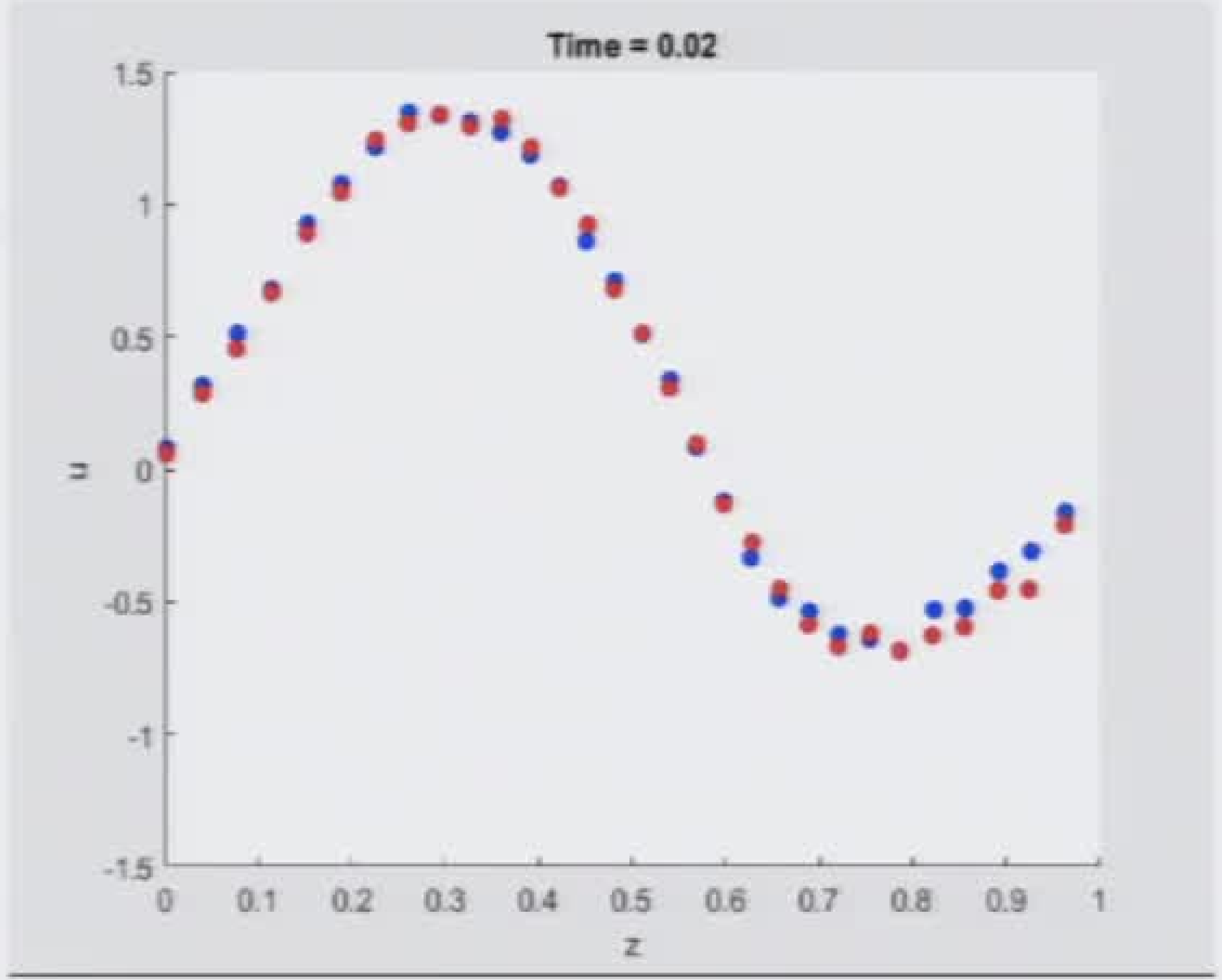
Super-meshing in the 1-D Case

- Each ensemble member will have its own adaptive mesh
- The adaptive mesh, $z(t)$, evolves in time
- We consider the mesh valid if $\delta_1 \leq z_{i+1} - z_i < \delta_2$ for all i
- Otherwise, a remeshing process occurs
- To left is example with $\delta_1 = 0.02$ and $\delta_2 = 0.05$



Evolution of Ensemble Members

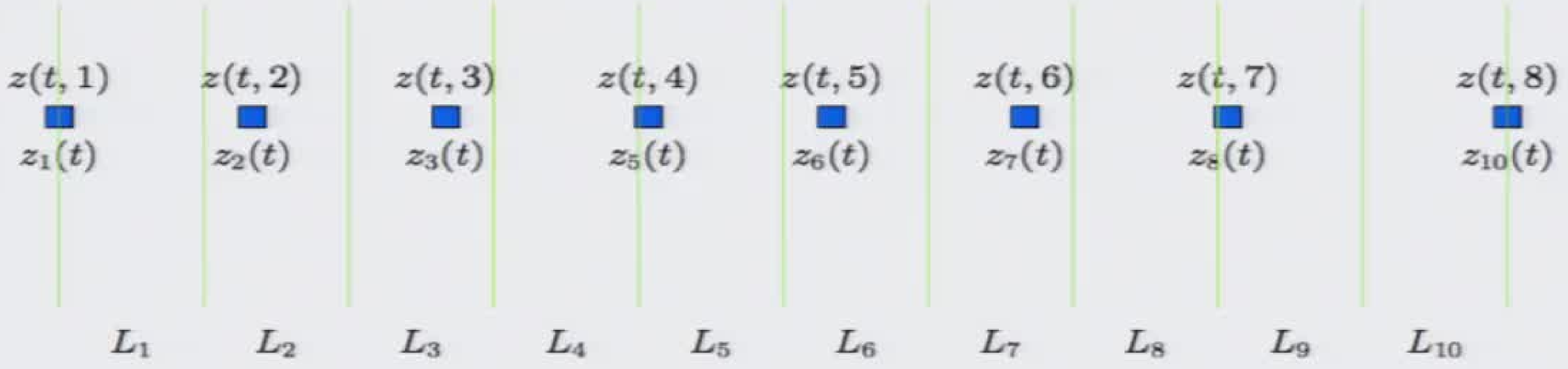
- Each ensemble member has its own adaptive mesh
- Values of u are defined at adaptive mesh points
- Ensemble members evolve according to adaptive moving mesh equations
- Evolution of u and evolution of mesh are coupled
- Remeshing occurs when mesh points become too close together, or too far apart



Projecting onto the supermesh

The smaller mesh parameter, δ_1, is what allows us to define a constant-dimensional state space for our DA algorithm. Specifically, letting $N = \frac{1}{\delta_1}$, we write

$$\begin{aligned}[0,1) &= [0,\delta_1) \cup [\delta_1, 2\delta_1) \cup \dots \\ &\cup [(N-2)\delta_1, (N-1)\delta_1) \cup [(N-1)\delta_1, 1) \\ &= L_1 \cup L_2 \cup \cdots \cup L_N.\end{aligned}$$



Defining the State Space

We then define

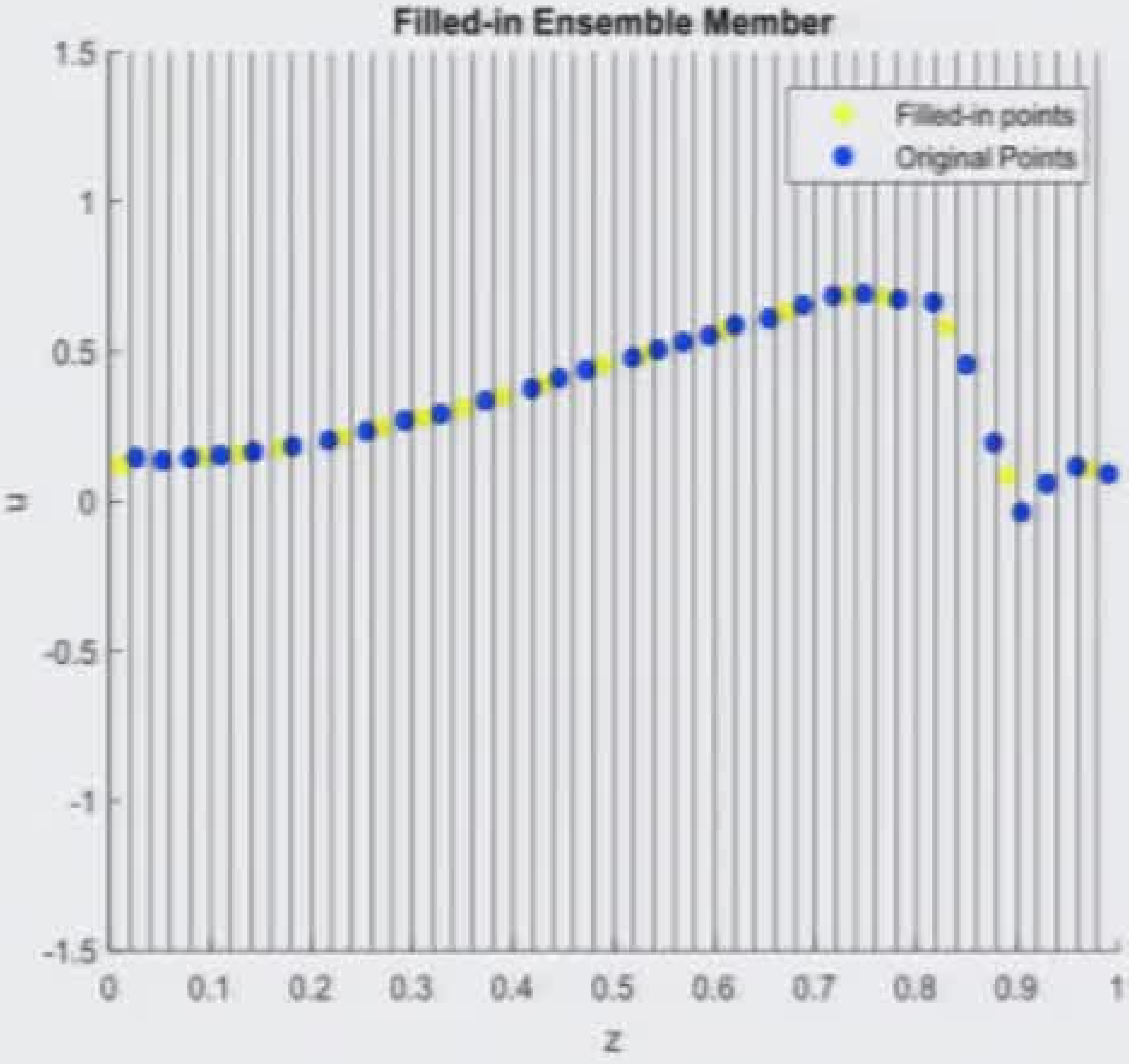
$$z_i(t) = z(t, j) \qquad \text{(mesh point)}$$
$$u_i(t) = u(t, z(t, j)) \qquad \text{(physical variable)}$$

if $z(t, j) \in L_i$. With this definition, we can define our state vector

$$\mathbf{x}(t) = \begin{pmatrix} \mathbf{u}(t) \\ \mathbf{z}(t) \end{pmatrix} = \begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_N(t) \\ z_1(t) \\ z_2(t) \\ \vdots \\ z_N(t) \end{pmatrix}$$

Filling in the Ensemble Members

- For now, we only perform data assimilation on the physical variables, and not the mesh points themselves
- Ensemble members will generally have different meshes, so they will have different "active" cells
- We address this by "filling in" each ensemble mesh
- This allows us to carry out DA in the standard way on a state space of full dimension



Ensemble Kalman Filter (EnKF) with super-meshing (1D-case)

Define

$$\mathbf{A} \in \mathbb{R}^{N \times N_e}$$

to be the matrix of forecast ensemble members. We subtract the matrix whose columns contain the ensemble means to form the matrix of anomalies:

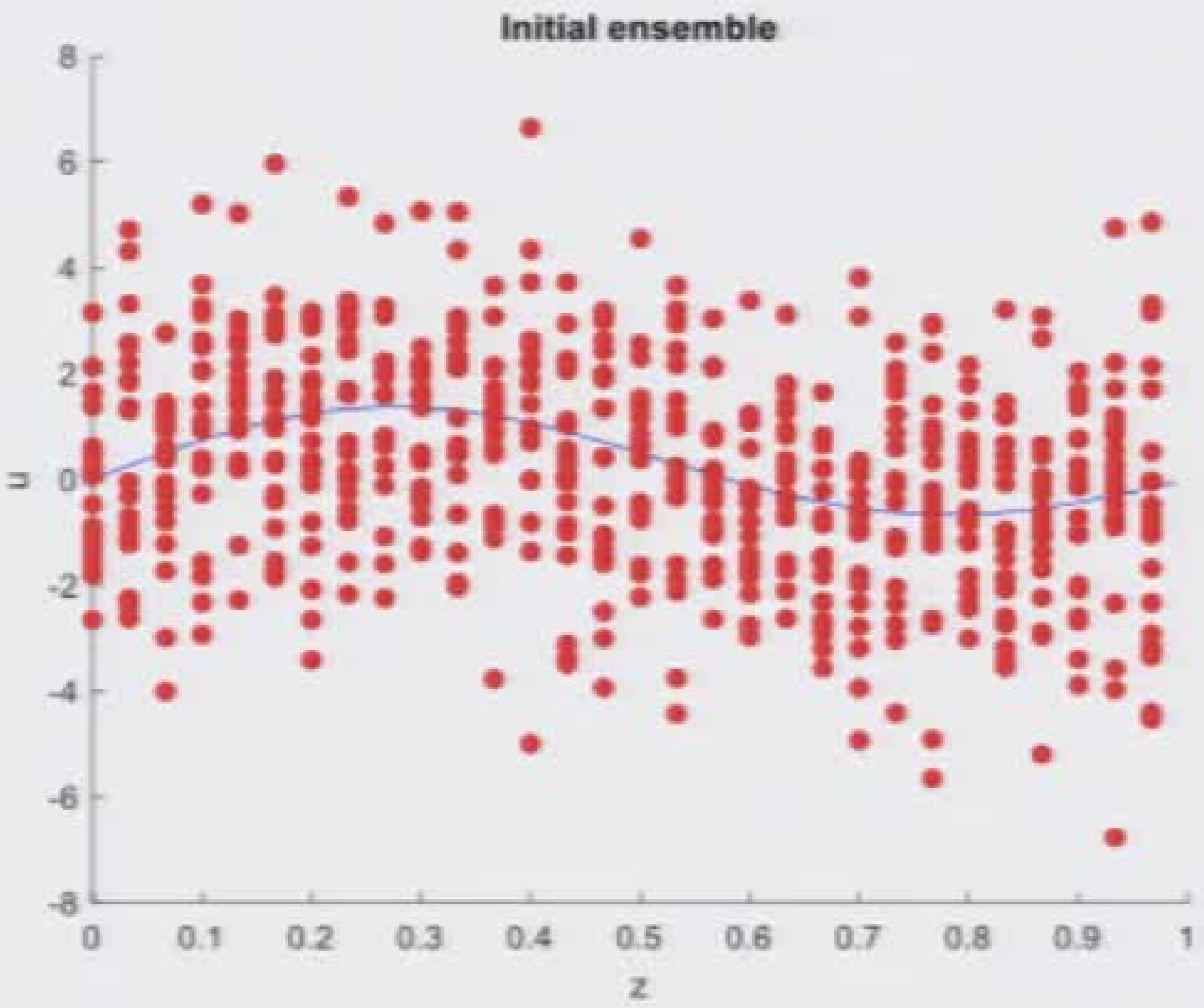
$$\mathbf{A}' = \mathbf{A} - \bar{\mathbf{A}} \in \mathbb{R}^{N \times N_e}$$

We assume we have p observations of u at certain points in the interval $[0,1)$, and that we have an observation operator that maps the state space to the observation space. We can then define the matrix

$$\mathbf{S} = \mathcal{H}(\mathbf{A}) - \mathcal{H}(\bar{\mathbf{A}}) \in \mathbb{R}^{N \times p}$$

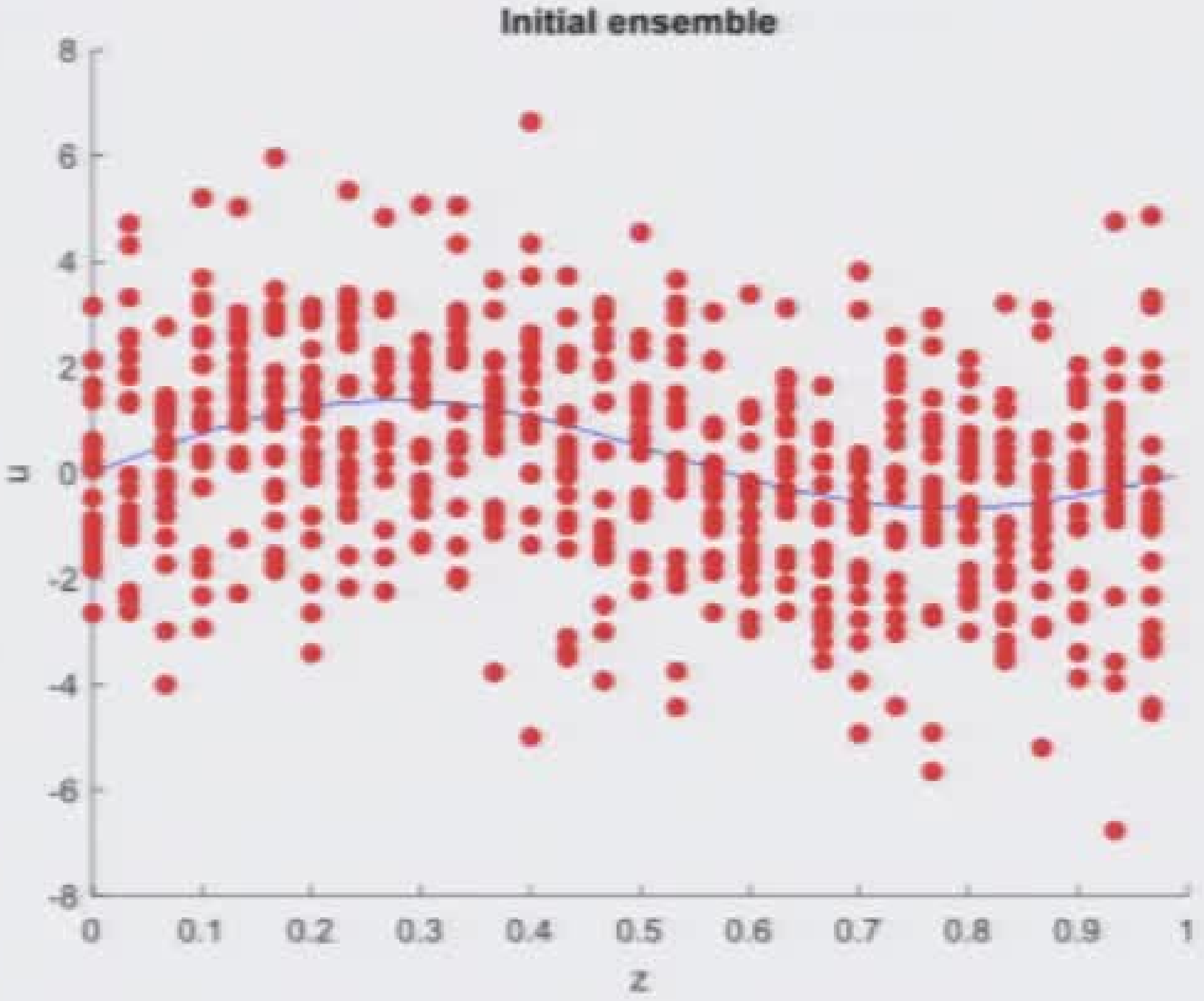
Numerical Experiment

- We run the experiment from time $t = 0$ to $t = 1$, performing data assimilation time step $t_{obs} = 0.2$
- We use an ensemble of size $N_e = 20$. Initially, all ensemble members have the same mesh. Initial ensemble values of u centered at true initial value of u with standard deviation $\sigma_{ens} = 2$

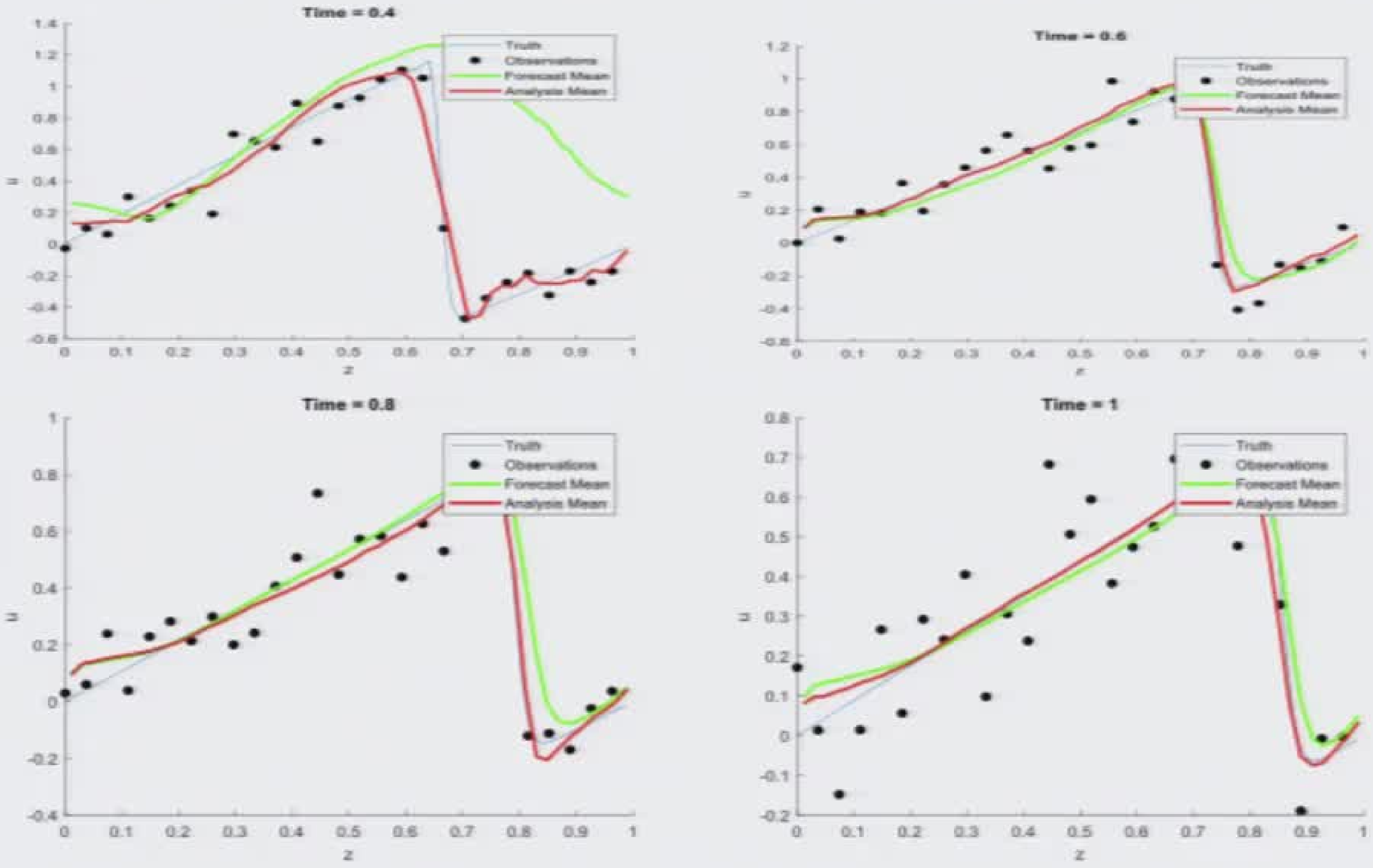


Numerical Experiment

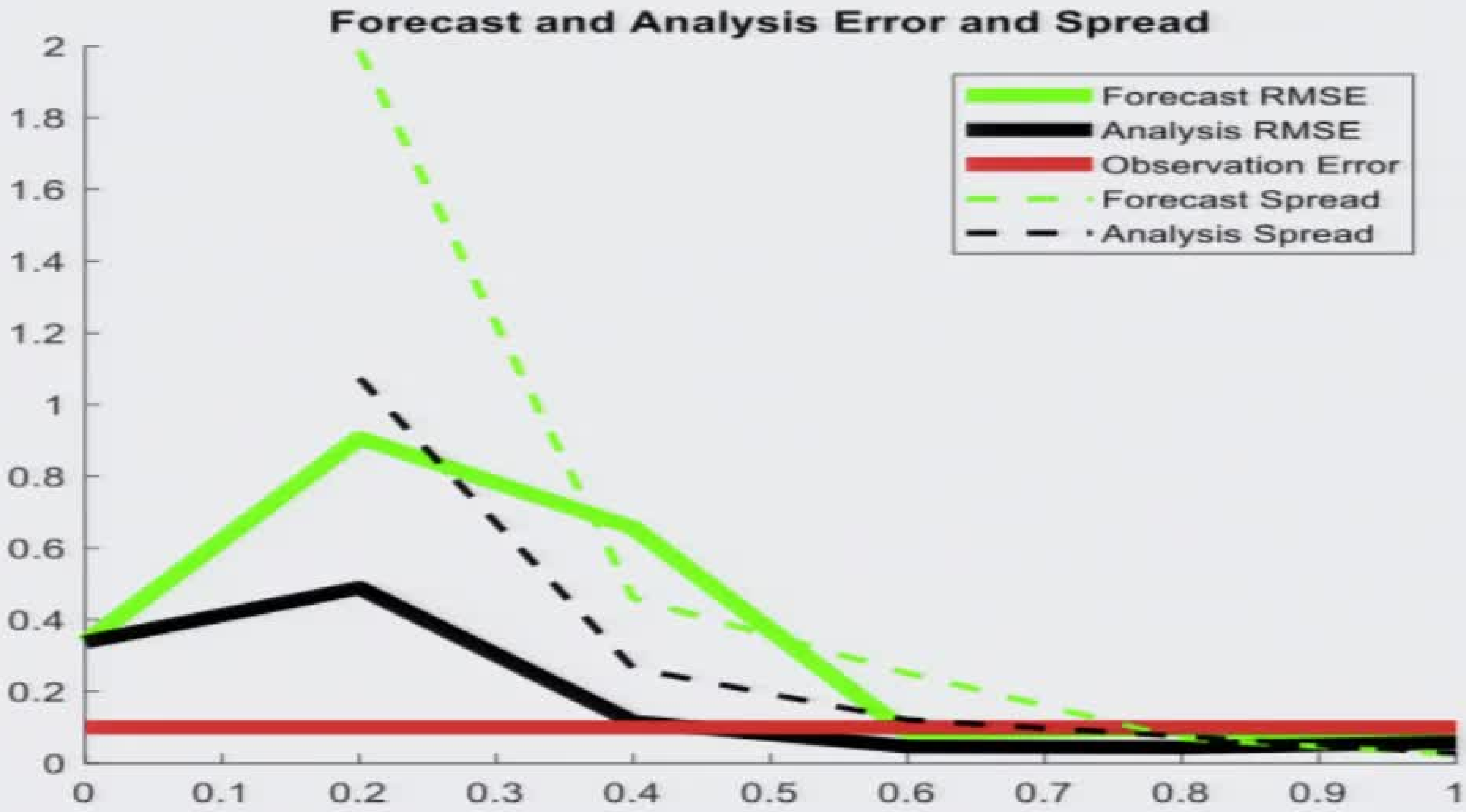
- We run the experiment from time $t = 0$ to $t = 1$, performing data assimilation time step $t_{obs} = 0.2$
- We use an ensemble of size $N_e = 20$. Initially, all ensemble members have the same mesh. Initial ensemble values of u centered at true initial value of u with standard deviation $\sigma_{ens} = 2$



Forecast and Analysis Means



RMSE and Spread



Future Directions

- Run experiment with Lagrangian observations
- Test a more interesting one-dimensional model (Kuramoto-Sivashinsky experiment in progress)
- Further develop a "low-resolution" model, in which the state space is defined by larger remeshing parameter
- Eventually work up to two-dimensional models, with neXtSIM the ultimate goal