Solving PDE-Based Bayesian Inverse Problems Using

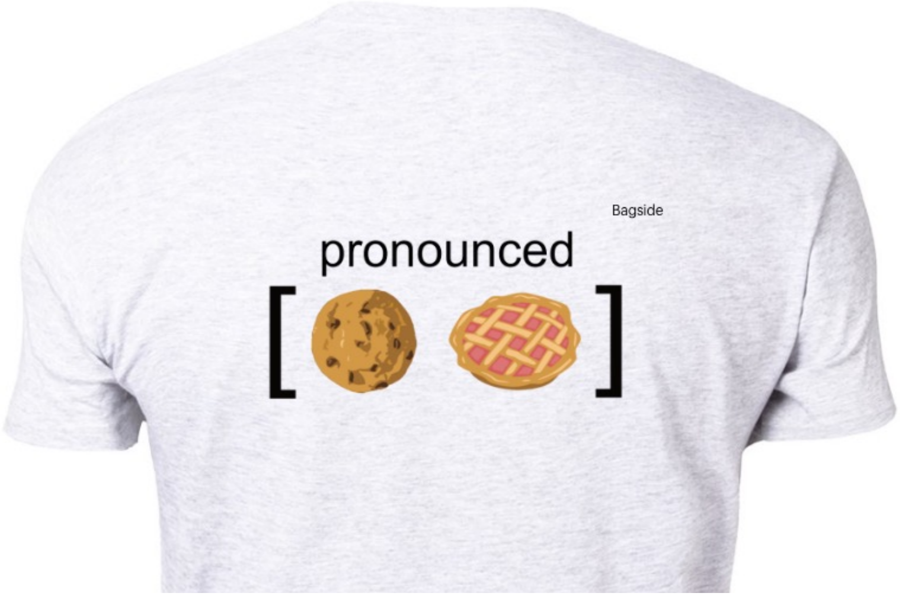
CUQIpy

Joint work with:
Babak Afkham
Silja L. Christensen
Felipe Uribe
Per Christian Hansen

VILLUM FONDEN

Amal Alghamdi
Nicolai Riis
Chao (Charlie) Zhang
Jakob Sauer Jørgensen

28 Feb 2024



VILLUM FONDEN

The CUQI team




Nov. 2021

- Computational Uncertainty Quantification for Inverse problems
- **2019 – 2025**
- PI: Per Christian Hansen
- https://sites.dtu.dk/cuqi

CUQIpy core-developers

Jakob **Nicolai** **Amal** **Charlie**

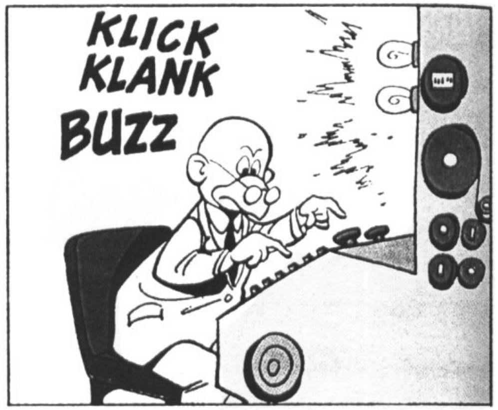
+ CUQI team valuable contribution

CUQIpy in a Nutshell

Vision

Build a <u>software package</u> that uses uncertainty quantification (UQ) to access and quantify uncertainties in solutions to inverse problems.

- **Simplify** the mathematics, statistics and code for the non-expert user.
- Provide **full control** for expert users.
- Allow users to focus on **modeling aspects**.



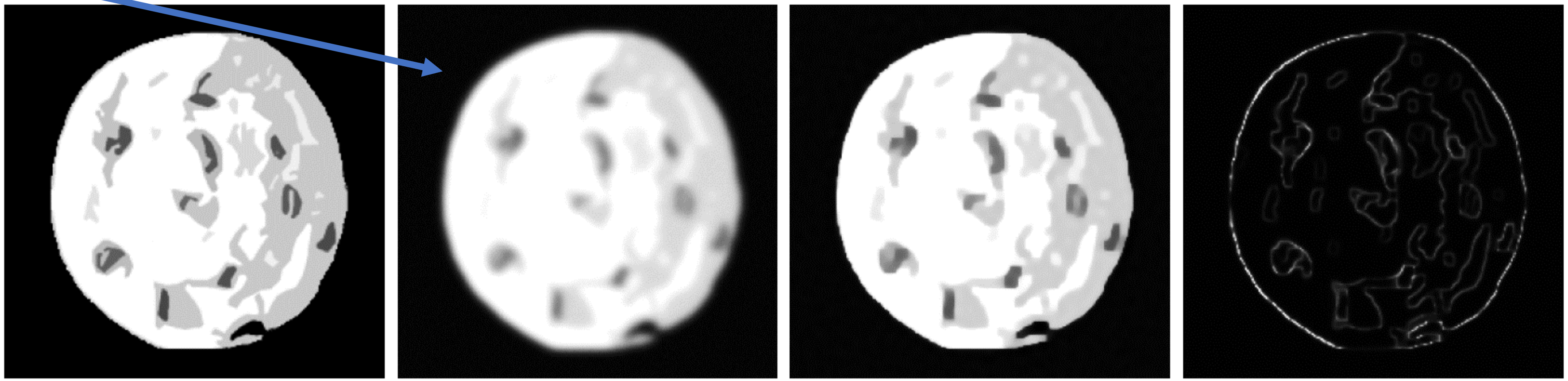
Features

- Easy access to **state-of-the-art** tools in one framework (including 3rd party libraries).
    - Modeling
    - Solving
    - Visualization and statistics
- A suite of **test problems** to allow users to get started.
- Allow users to provide **custom code** for models, distributions, samplers etc.
- Exploit structure to support **large-scale** problems.

Cookie deblurring with CUQIpy

$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$

True | Blurred, noisy | Posterior mean | Posterior std



$$d \sim \mathrm{Gamma}(1, 10^{-4})$$

$$s \sim \mathrm{Gamma}(1, 10^{-4})$$

$$\boldsymbol{x} \sim \mathrm{LMRF}(d^{-1}),$$

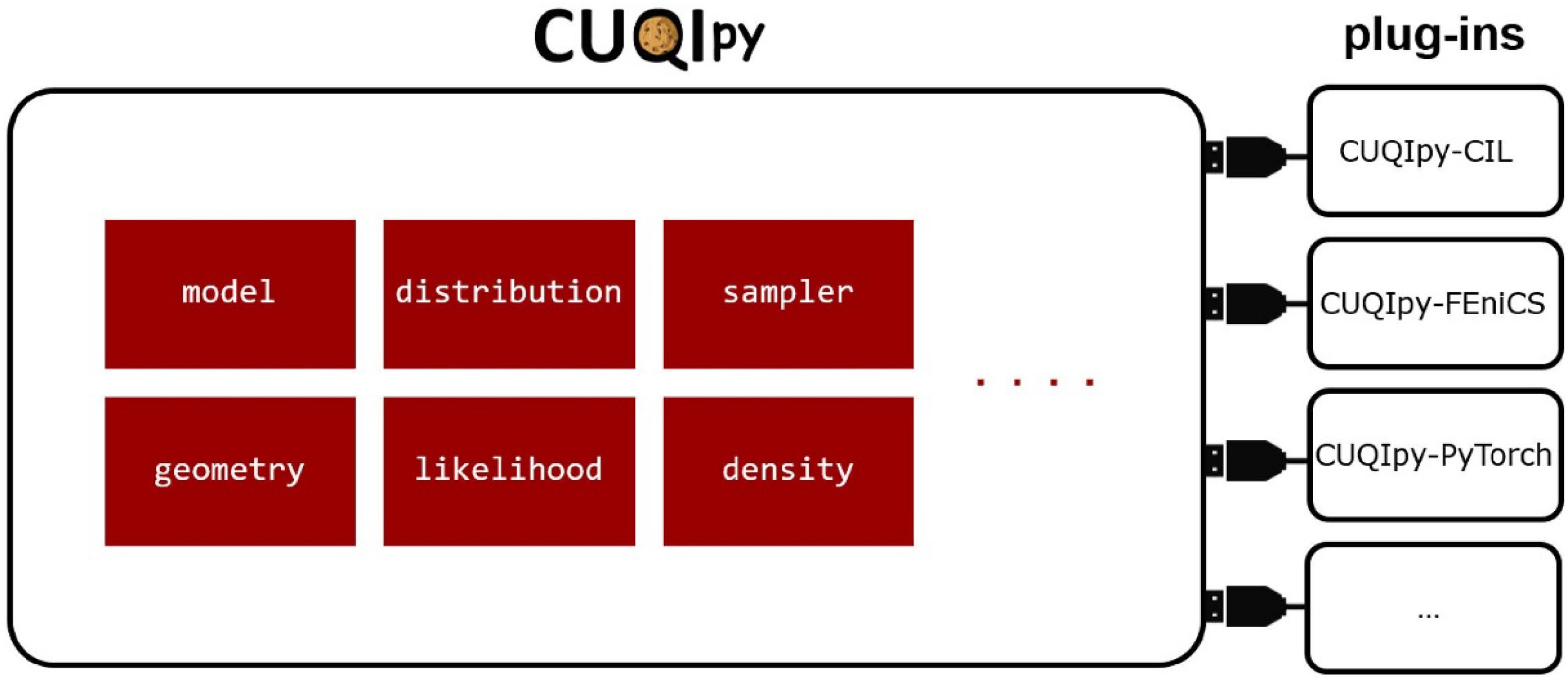
$$\boldsymbol{y} \sim \mathrm{Gaussian}(\boldsymbol{A}\boldsymbol{x}, s^{-1}\boldsymbol{I})$$

```
d = Gamma(1,1e-4), s = Gamma(1,1e-4)
x = LMRF(1/d)
y = Gaussian(A @ x, 1/s)
BP = BayesianProblem(x, y) d, s)
BP.set_data(y=y_obs)
BP.UQ()
```

Automation using BayesianProblem

Riis, Alghamdi, Uribe, et al. “CUQIpy - Part I: computational uncertainty quantification for inverse problems in Python”. (to appear) Inverse Problems
https://github.com/CUQI-DTU/Paper-CUQIpy-1-Core/blob/main/intro/intro.ipynb

CUQIpy modules and plug-ins

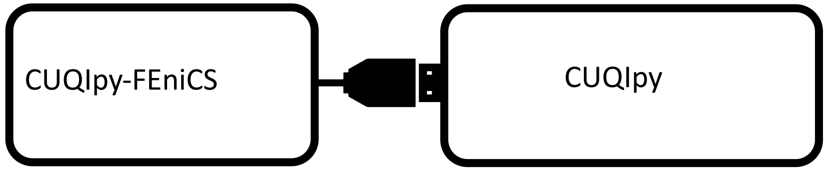


- Plug-in structure modularity / flexible licensing

3 PDE-based Bayesian inverse problems (BIP):

- Case 1: Poisson 2D problem
- Case 2: Electric Impedance Tomography (EIT)
- Case 3: Characterizing ear aqueduct in mice (Diffusion model)

PDE-BIP Case 1: 2D Poisson Problem
Bayesian model



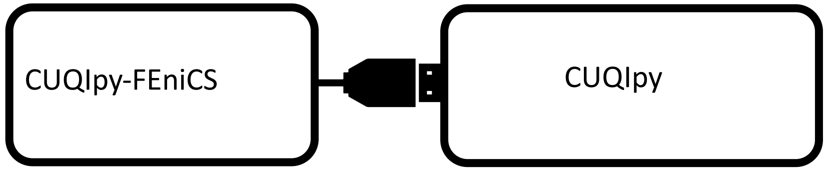
Model

$$\nabla \cdot \left(e^{w(\boldsymbol{\xi})} \nabla u(\boldsymbol{\xi}) \right) = f(\boldsymbol{\xi}) \qquad \text{for} \qquad \boldsymbol{\xi} \in \Gamma = (0,1)^2$$

$$\boldsymbol{w} = \sum_{i=1}^{n_{\mathrm{KL}}} x_i \sqrt{\lambda_i} \boldsymbol{e}_i^{\mathrm{KL}}, \qquad \boldsymbol{x} = [x_1, x_2, \ldots, x_{n_{\mathrm{KL}}}]^{\mathsf{T}}$$

Alghamdi, Riis, Afkham, et al. "CUQIpy - Part II: computational uncertainty quantification for PDE-based inverse problems in Python". (to appear) Inverse Problems
https://github.com/CUQI-DTU/Paper-CUQIpy-2-PDE/blob/main/Poisson/intro_example_Poisson.ipynb

PDE-BIP Case 1: 2D Poisson Problem
Bayesian model



Model

$$\mathbf{y} = \mathbf{A}(\mathbf{x})$$

```
# Step 1: Model
A = FEniCSPoisson2D(dim=(32,32), field_type="KL",
...).model
```

Prior

$$\boldsymbol{x} \sim \text{Gaussian}(\mathbf{0}, \boldsymbol{I})$$

```
# Step 2: Prior
x = Gaussian(np.zeros(n_KL), 1, geometry=G_KL)
```

Data distribution

$$\boldsymbol{y} \sim \text{Gaussian}\left(\boldsymbol{A}(\boldsymbol{x}), s_{\text{noise}}^2 \boldsymbol{I}\right).$$

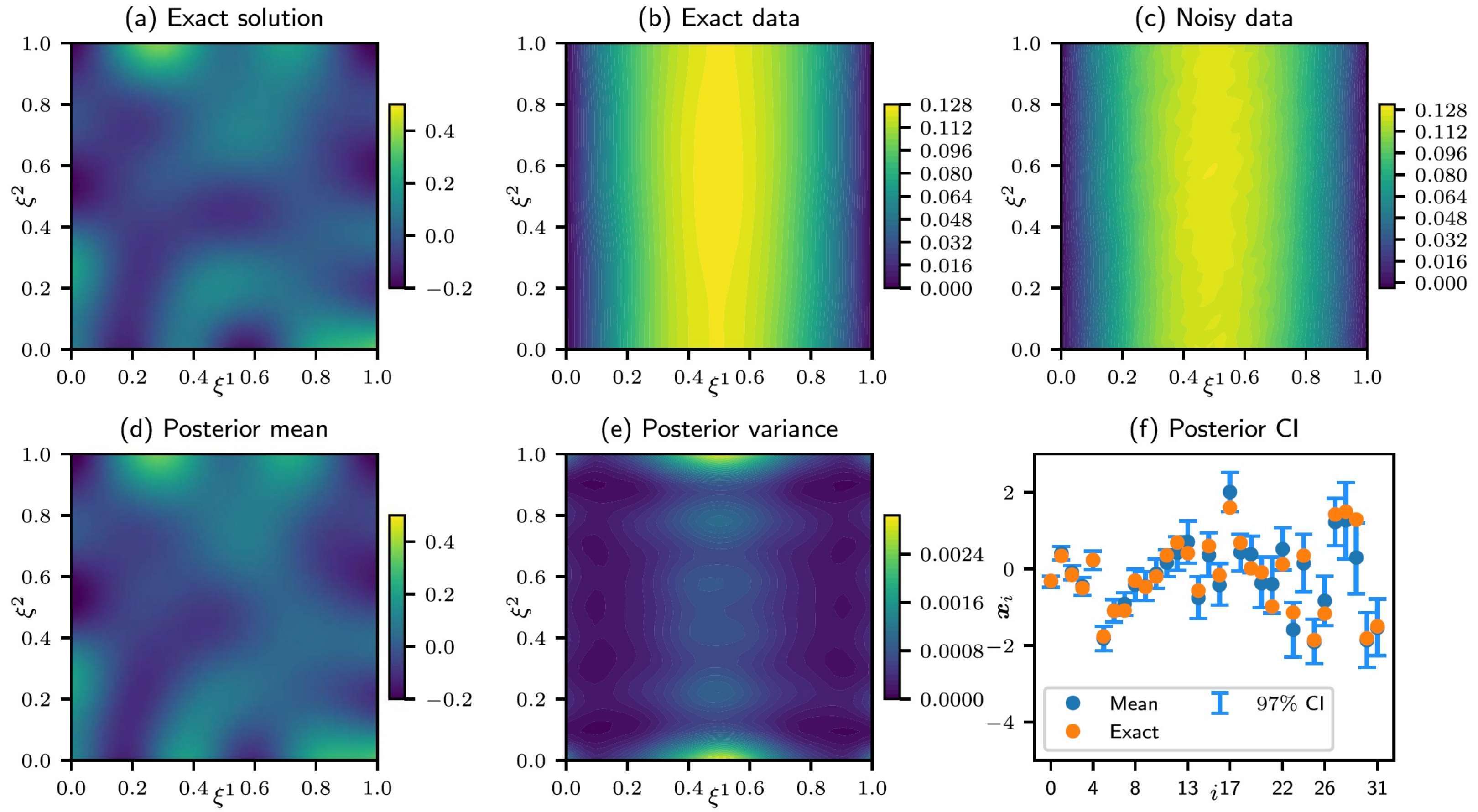
```
# Step 3: Data distribution / Likelihood
y = Gaussian(A(x), s_noise**2, geometry=G_FEM)
```


Alghamdi, Riis, Afkham, et al. “CUQIpy - Part II: computational uncertainty quantification for PDE-based inverse problems in Python”. (to appear) Inverse Problems
https://github.com/CUQI-DTU/Paper-CUQIpy-2-PDE/blob/main/Poisson/intro_example_Poisson.ipynb

PDE-BIP Case 1: 2D Poisson Problem
Bayesian model

```
# Sample the prior
x_true = x.sample()
X_true.plot()
```

```
# Sample the data distribution
y_obs = y(x=x_true).sample()
y_obs.plot()
```

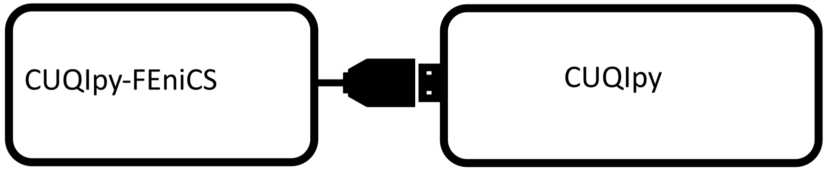


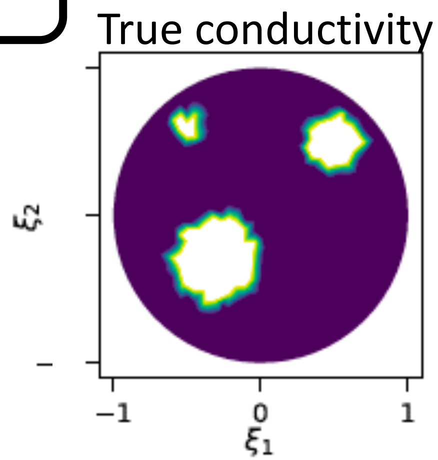

```
# Setting up the Bayesian inverse problem
BP = BayesianProblem(y, x).set_data(y=y_obs)
BP.UQ(..)
```

Automation using BayesianProblem

Alghamdi, Riis, Afkham, et al. “CUQIpy - Part II: computational uncertainty quantification for PDE-based inverse problems in Python”. (to appear) Inverse Problems
https://github.com/CUQI-DTU/Paper-CUQIpy-2-PDE/blob/main/Poisson/intro_example_Poisson.ipynb

PDE-BIP Case 2: EIT
Forward model





EIT problem (medical imaging)

- Infer conductivity
- Measure electric current w on the boundaries
- We impose electrical potential on the boundaries (multiple frequencies $k = 1, 2, 3, 4$)

Forward model

$$-\nabla \cdot (\sigma(\boldsymbol{\xi}) \nabla u_k(\boldsymbol{\xi})) = 0,$$

$$u_k(\boldsymbol{\xi}) = g_k(\boldsymbol{\xi}) = \sin\left(k \arctan\left(\frac{\xi^2}{\xi^1}\right)\right)$$

Observations

$$y_k(\boldsymbol{\xi}) := \frac{\sigma(\boldsymbol{\xi}) \partial u_k(\boldsymbol{\xi})}{\partial \mathbf{n}}$$

```
PDE1 =SteadyStateLinearFEniCSPDE(
    (lhs_form, rhs_form_k1), …,
    observation_operator=obs_map1)

PDE2 = PDE1.with_updated_rhs(rhs_form_k2)
PDE2.observation_operator = obs_map2
```

Similarly for k= 3, 4

EIT Test case by Babak Afkham, with CUQIpy core developer team support

Alghamdi, Riis, Afkham, et al. “CUQIpy - Part II: computational uncertainty quantification for PDE-based inverse problems in Python”. (to appear) Inverse Problems

https://github.com/CUQI-DTU/Paper-CUQIpy-2-PDE/blob/main/EIT/EIT.ipynb

PDE-BIP Case 2: EIT
Prior

Parametrize the conductivity: $\boldsymbol{G}_{\text{Heavi}} \circ \boldsymbol{G}_{\text{KL}}(\boldsymbol{x})$

Finite element function space of

$\sigma_{\text{FEM}}(\boldsymbol{\xi})$

Karhunen-Loève (KL) expansion of Matérn covariance

$$\boldsymbol{r} = \sum_{i=1}^{n_{\text{KL}}} x_i \sqrt{\lambda_i} \boldsymbol{e}_i^{\text{KL}}.$$

Level-Set mapping

$$\boldsymbol{\sigma} = \boldsymbol{G}_{\text{Heavi}}(\boldsymbol{r}) := \frac{1}{2}\left(\sigma^+ (1 - \operatorname{sign}(\boldsymbol{r})) + \sigma^- (1 + \operatorname{sign}(\boldsymbol{r}))\right).$$

```
G_FEM = FEniCSContinuous(Vh)

G_KL = MaternKLExpansion(G_FEM, …)

G_Haevi = FEniCSMappedGeometry(G_KL,
    map=heaviside, …)
```

PDE-BIP Case 2: EIT

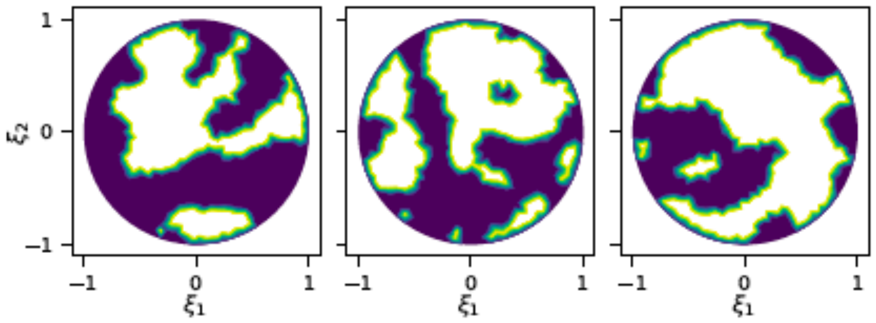
Prior

Parametrize the conductivity: $G_{\text{Heavi}} \circ G_{\text{KL}}(\boldsymbol{x})$

$$\boldsymbol{x} \sim \text{Gaussian}(\boldsymbol{0}, \boldsymbol{I}_{n_{\text{KL}}})$$

```
x = Gaussian(0, 1, geometry=G_Haevi)
```

```
prior_sample = x.sample(5)
prior_sample.plot()
```



PDE-BIP Case 2: EIT
Multiple likelihood posterior

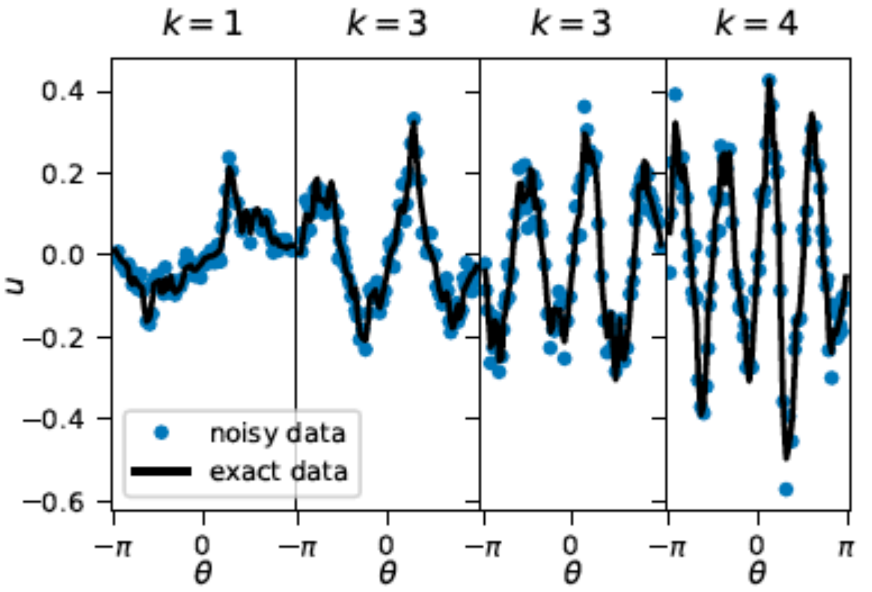
Data distribution

$$\boldsymbol{y}_k \sim \text{Gaussian}\left(\boldsymbol{A}_k\left(\boldsymbol{x}\right), s_{\text{noise}}^2 \boldsymbol{I}_m\right)$$

$$k = 1, 2, 3, 4.$$

```
y1 = Gaussian(A1(x),
     s_noise**2,
     geometry=G_cont)
```

Similarly for k= 2, 3, 4



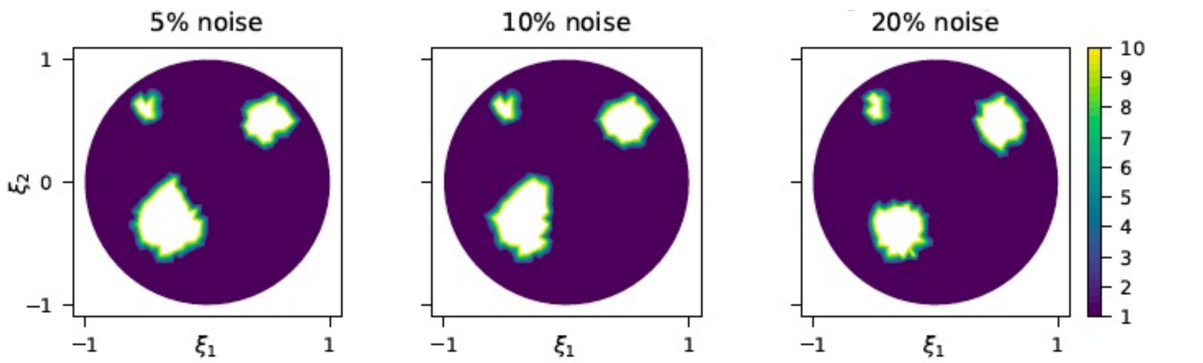
Posterior

$$p(\mathbf{x}|\mathbf{y}_1, \ldots, \mathbf{y}_k) \propto p(\mathbf{y}_1, \ldots, \mathbf{y}_k|\mathbf{x})p(\mathbf{x}) = p(\mathbf{y}_1|\mathbf{x}) \ldots p(\mathbf{y}_k|\mathbf{x})p(\mathbf{x})$$

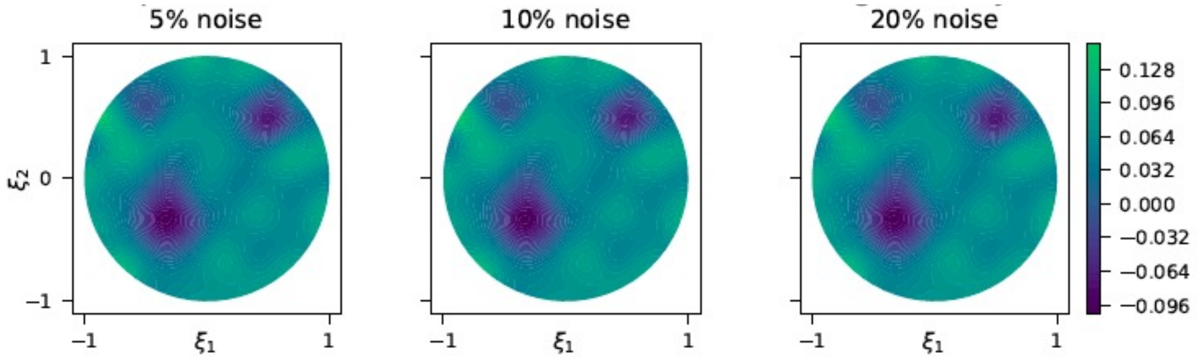
```
post = JointDistribution(x, y1, y2, y3, y4)(y1=y1_data, y2=y2_data, y3=y3_data, y4=y4_data)
```

PDE-BIP Case 2: EIT Results

```python
# Plot mean
posterior_samples.plot_mean()
```

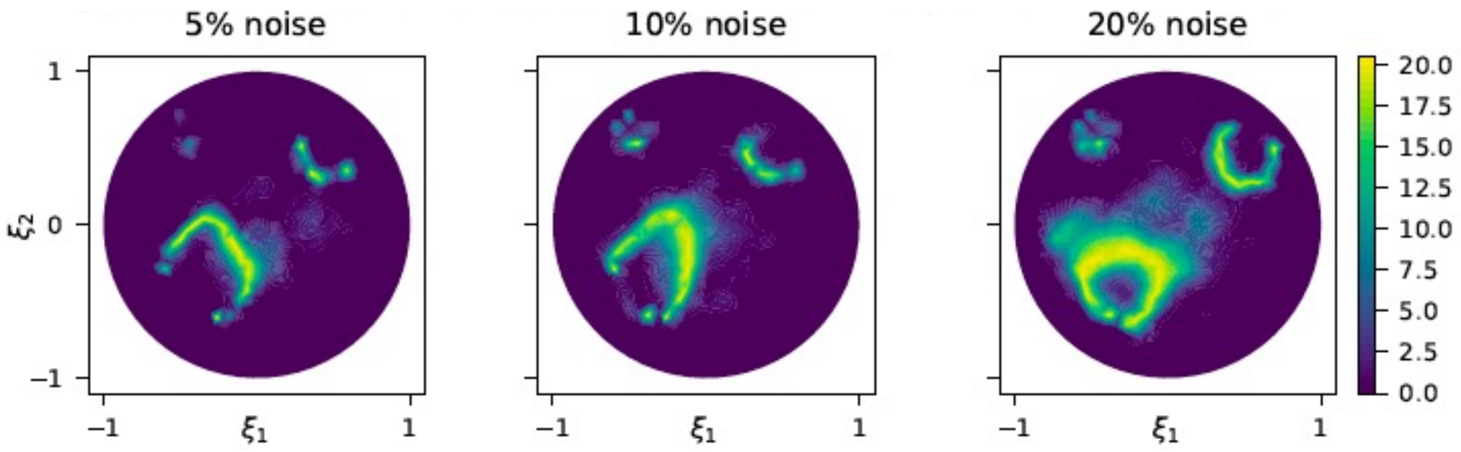



```python
# Plot mean
posterior_samples.geometry = G_KL
posterior_samples.plot_mean()
```

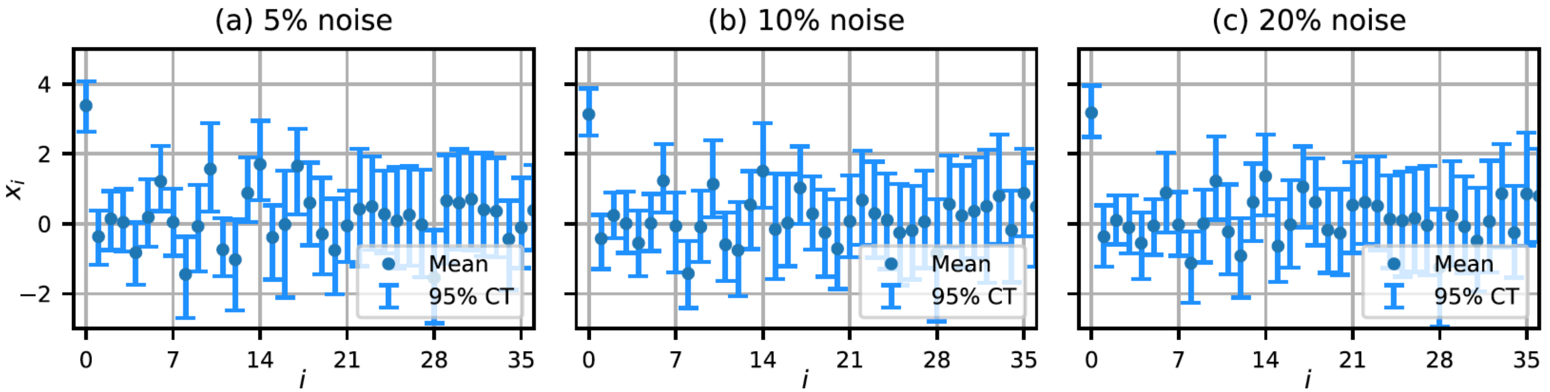


PDE-BIP Case 2: EIT
Results

```python
# Plot variance
posterior_samples.funvals.vector.plot_variance()
```




```python
# Plot credibility intervals
posterior_samples.plot_ci(95, plot_par=True)
```



PDE-BIP Case 3: Characterize flow of brain-injected tracer in mice inner ear

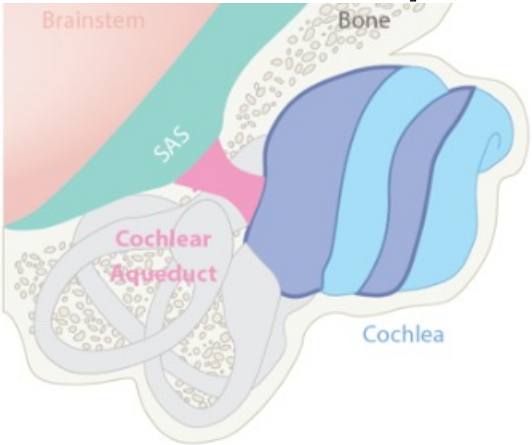
CUQIpy

- Joint work with Barbara K. Mathiesen and Peter A. R. Bork, Center for Translational Neuromedicine, University of Copenhagen

Motivation and implications:

- Understand tracer/particle/therapeutics flow from brain to inner ear.
- Can we identify existence of a membrane in the inner ear?
- Implications on e.g., delivery of therapeutics to inner ear (to rescue hearing)

Figures credit [Mathiesen, Barbara K., et al., 2023]



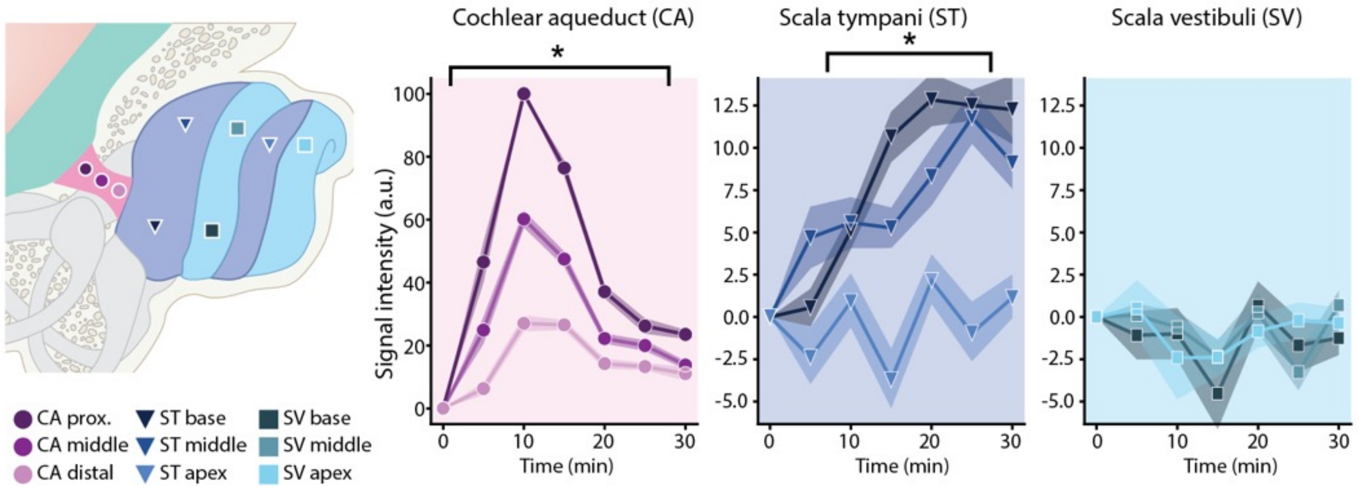
Mathiesen et al. (2023). Delivery of gene therapy through a cerebrospinal fluid conduit to rescue hearing in adult mice. *Science Translational Medicine*, *15*(702), eabq3916

PDE-BIP Case 3: Characterize flow of brain-injected tracer in mice inner ear

CUQIpy

Approach

- Bayesian inversion framework in CUQIpy
  - Advection-diffusion model
  - Gaussian Markov Random Field
- CT/X-ray tracer data



Mathiesen et al. (2023). Delivery of gene therapy through a cerebrospinal fluid conduit to rescue hearing in adult mice. *Science Translational Medicine*, *15*(702), eabq3916

PDE-BIP Case 3: Characterize flow of brain-injected tracer in mice inner ear

The forward model

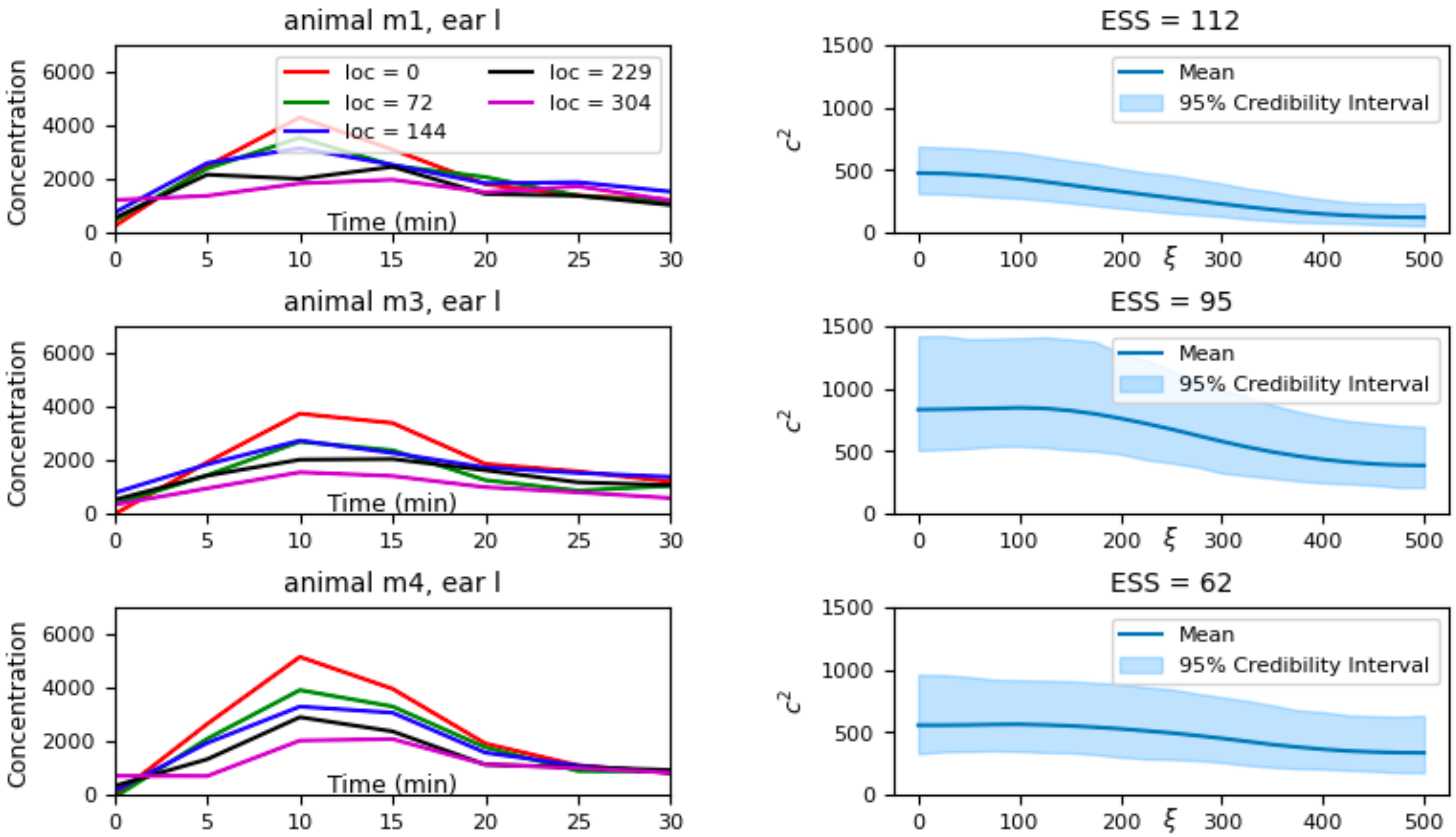
$$y^{\text{obs}} = A(x) + \epsilon$$

- x is the unknown parameters representing $c^2(\xi)$, the varying in space diffusivity
- ϵ is the measurement noise
- y^{obs} is the noisy concentrations at the given times and locations
- A is the forward map from x to the non-noisy measurement y (involves solving the time-dependent diffusion equation)

Preliminary results show identification of low diffusivity area towards the ear

We are interested in advection-diffusion model characterization estimation

Results from different animals



Future directions

- Increate parameterization/prior support
- Increase automatic/symbolic differentiation support
  - FEniCS derivatives [hIPPYlib]
  - FEniCS-adjoint?
  - Sundials?
  - pyTorch?
- Large-scale
- Other applications and methods
  - e.g. delayed acceptance
  - Photoacoustic tomography

Upcoming event

Workshop: UQ for Inverse Problems and Imaging (UQIPI24)

- September 16-20, 2024 @ ICMS, Bayes Centre, Edinburgh
- Organizers: Per Christian Hansen, Marcelo Pereyra, and Yiqiu Dong.
- Workshop homepage: https://www.icms.org.uk/UQIPI24

Programme

	Monday	Tuesday	Wednesday	Thursday	Friday
Morning	UQ tutorial	CUQIpy course	Workshop	Workshop	Workshop
After-noon	CUQIpy course	Workshop	Workshop	Workshop	Social event
Evening	CUQIpy course for the nerds	Reception		Guided tour & workshop dinner	

Plenary Speakers

- Yoann Altmann, Heriot-Watt University
- Tatiana Bubba, University of Bath
- Per Christian Hansen, Technical University of Denmark
- Aku Seppänen, University of Eastern Finland
- Julián Tachella, CNRS and ENS de Lyon
- Faouzi Triki, Grenoble-Alpes University

CUQIpy resources

- **Main repository:** https://github.com/CUQI-DTU/CUQIpy
- **Training notebooks:** https://github.com/CUQI-DTU/CUQIpy-demos
- **Documentation:** https://cuqi-dtu.github.io/CUQIpy/
- **Plugins:**
    - CUQIpy_CIL: https://github.com/CUQI-DTU/CUQIpy-CIL
    - CUQIpy_FEniCS: https://github.com/CUQI-DTU/CUQIpy-FEniCS
    - CUQIpy_PyTorch: https://github.com/CUQI-DTU/CUQIpy-PyTorch
    - CUQIpy_Umbridge: https://github.com/CUQI-DTU/CUQIpy-UMBridge
- **User showcase** https://github.com/CUQI-DTU/CUQIpy-User-Showcase
- **Publications:** Part I: https://iopscience.iop.org/article/10.1088/1361-6420/ad22e7
  https://github.com/CUQI-DTU/Paper-CUQIpy-1-Core
  Part II: https://iopscience.iop.org/article/10.1088/1361-6420/ad22e8
  https://github.com/CUQI-DTU/Paper-CUQIpy-2-PDE