





## **CUQI** <u>Computational Uncertainty Quantification</u> for <u>Inverse Problems</u>

A brief introduction to the project

sites.dtu.dk/cuqi



**Goals:** 1. Build an interdisciplinary and collaborative research team.

- 2. Develop the necessary theory and methodology.
- 3. Create a modeling and computational framework.
- 4. Thus, put us on the world map of computational UQ.





### Towards CUQI – Can We Trust Our Results?

Use X-ray scanning to compute cross-sectional images of oil pipes on the seabed. Detect *defects, cracks, etc.* in the pipe that can lead to loss of oil and contamination. It is expensive and cumbersome to repair the pipe – do it only when necessary.





In an inverse problem we use measured data to infer about an object that is hidden or unavailable.



In an inverse problem we use measured data to infer about an object that is hidden or unavailable.





In an inverse problem we use measured data to infer about an object that is hidden or unavailable.







In an inverse problem we use measured data to infer about an object that is hidden or unavailable.





In an inverse problem we use measured data to infer about an object that is hidden or unavailable.

To do this, we "invert" a mathematical model for the relation between the object and the data.



Oct. 5, 2022



### What is **Uncertanty Quantification**?

### 

Uncertainty Quantification (**UQ**) is the study of the impact of all forms of error and uncertainty in the data and the math models.

### Example: Image Deblurring

Picture of the night sky, blurred due to the rotation of the Earth.





**UQ** gives the mean estimate of the image

**UQ** shows the uncertainty (variance) in the pixels

White = high uncertainty



Make UQ operational and accessible requires:

- a computational modeling platform
- fast, robust, and easy-to-use **software** (written in python)



#### Philosophy

Oct. 5. 2022

- Hide mathematics, statistics and scientific computing from non-experts.
- Give expert users full control of the UQ methods and computations.
- Publish under open-source license as agreed with DTU.

#### **CUQI** is a multidisciplinary project that

- 1. Brings many different topics together,
- 2. let them benefit from each other, and
- 3. aims at a general computational *framework*.

Examples of cross-disciplinary work:

i) new sampling methods for inverse problems based on new ideas in computational statistics;

ii) goal-oriented UQ based on stochastic functions.



### Status for the 4 Goals

- 1. Build an interdisciplinary and collaborative research team
  - > 6 VIPs, 4 postdocs, 8 PhD students, 8 external collaborators.
  - Collaborate efforts: PhD supervision, software development, etc.
- 2. Develop the necessary theory and methodology
  - > **36 publications** as of today, 6 with open access
  - Recurring PhD course at DTU Jan. 2022: 27 participants
- 3. Create a modeling and computational framework
  - > Released first version of the **CUQIPY** software.
  - Gave training courses (Sept. 15: 20 participants online; Sept. 26: 16 participants).
- 4. Put us on the world map of computational UQ
  - > Workshop "Imaging with UQ" last week with 44 participants 50% from abroad.
  - > Invited lectures, e.g., IMAGINE online seminars (Yiqiu, Per Chr.), UC Irvine (Per Chr.).







# 

### Example of Bayesian Inference with Projected Densities



(Everink, Andersen and Dong, preprint 1+2, 2022)



### Example of Boundary Reconstruction in CT

**Goal**: In CT problems, we reconstruct the boundary of the inclusions directly and we quantify the uncertainties of the boundary curves.



(Afkham, Dong and Hansen, SIAM/ASA Journal on UQ, 2022)





Computational Uncertainty Quantification for Inverse problems in python

#### 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**.

# KLICK KLANK BUZZ

#### **Features**

- Easy access to **state-of-the-art** tools in one framework (including 3<sup>rd</sup> party libraries).
- > 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.



### Collaborative development

#### **Developed on GitHub since 2020**

- Sept 2022 first public version released
- <u>github.com/CUQI-DTU/CUQIpy</u>

#### Core team

- Nicolai Riis, Amal Alghamdi and myself
- New postdoc position opening soon!

#### Involvement of CUQI team

- Problem and feature requests
- Code contributions

#### Hackathons

- 2-day collaborative development days
- New users, new features vehicle for collaboration!



Oct. 5, 2022

### **CUQI**py is built as a Bayesian framework

**Generic inverse problem** 



Classic solution: a single estimate



**Bayesian rule** 

 $\mathbf{x}(\mathbf{x} \mid \mathbf{b}) \propto \pi(\mathbf{b} \mid \mathbf{x})\pi(\mathbf{x})$ 

**Posterior:** the probability density function (pdf) of the parameters given the observed data

**Likelihood:** density of the data given the parameters

**Prior:** (subjective) pdf of parameters before observing the data



samples from the **posterior**: a range of possible estimates!





### Handling two very different inverse problems in CUQIPY

1: Linear inverse problems

 $\mathbf{b} = \mathbf{A}\mathbf{x}$ 

Application: X-ray CT Model: line integrals



 $\mathbf{b} = \mathcal{G}(\boldsymbol{\theta})$ 

Application: Electrical impedance tomography Model: Steady-state diffusion PDE





Hauptman et. al. Open 2D Electrical Impedance Tomography data archive. fips.fi



UQ in 5 steps with CUQIPY

from cuqipy\_cil.testproblem
import ParallelBeam2DProblem
from cuqipy\_fenics.testproblem
import Poisson2D\_FEniCS

from cuqi.distribution
import Gaussian, GMRF

from cuqi.problem
import BayesianProblem

#### X-ray Computed Tomography

# Step 1: Model

A = ParallelBeam2DProblem(im\_size=(N,N), ...).model

# Step 2: Prior

```
# Step 3: Data distribution / Likelihood
```

y = Gaussian(mean=A@x, std=sigma)

```
# Step 4: Posterior sampling
BP = BayesianProblem(y, x).set_data(y=data)
samples = BP.UQ()
```

# Step 5: Analysis
samples.plot\_trace()

#### Electrical impedance tomography

# Step 1: Model

A = Poisson2D\_FEniCS(domain="circle", field="KL", ...).model

# Step 2: Prior

x = Gaussian(mean=np.zeros(n), std=delta)

# Step 3: Data distribution / Likelihood
y = Gaussian(mean=A(x), std=sigma)

```
# Step 4: Posterior sampling
BP = BayesianProblem(y, x).set_data(y=data)
samples = BP.UQ()
```

# Step 5: Analysis
samples.plot\_trace()



#### Defect Detection in Bayesian CT Imaging of Subsea Pipes



# CUQIPY

CT forward problem

DTU

$$\mathbf{d} = \mathbf{A}(\mathbf{x} + \boldsymbol{\varepsilon}) + \mathbf{e}$$

**Pipe structure** 

**Small defects** 

#### **Prior distributions**

We use a Structural Gaussian Prior [1] to promote the pipe structure in **x**:

$$\mathbf{x} \sim \mathcal{N}\left(\boldsymbol{\mu}_{SGP}, \left(\mathbf{R}_{SGP}^{T} \mathbf{R}_{SGP}\right)^{-1}\right).$$

There are few defects, so we enforce sparsity in  $\varepsilon$  with a hierarchical prior:

$$\varepsilon_i | \eta_i \sim \mathcal{N}(0, \eta_i), \qquad \eta_i \sim \mathcal{I}\mathcal{G}\left(\frac{\nu}{2}, \frac{\nu}{2}\omega^2\right)$$

[1] Christensen, Uribe, Riis and Jørgensen (arxiv.org/abs/2203.01030)



Figure 1: Phantom







Christensen, Riis, Pereyra and Jørgensen (in prep)

0.2

0.1

0.0

-0.1

# CUQIPY training at IUQ workshop





#### CUQIpy's Documentation

This software package is part of the CUQI project funded by the Villum Foundation.

#### Quick Links: Installation | Tutorials | How-To Guides | Source Repository



tailed descriptions of CUQIpy library components (modules, classes, methods, etc.). Here you find information on how to contribute to CUQIpy. All contributions are welcome, small and big!

"Well documented and easy to use."

"I think the whole user-experience was very smooth [...]"

"It's obvious that it is aimed towards non-experts, but it's also great that experts can really take advantage of the package and do more complex stuff." Install



pip install cuqipy

• Website

cuqi-dtu.github.io/CUQIpy

PyPI Stats

Downloads last day: 7 Downloads last week: 192 Downloads last month: 814

- Training material
- Expansion plugins
  - X-ray CT
  - PDE finite element
  - PyTorch autodiff

#### Next steps

Oct. 5. 2022

- Publication
- Collaboration
- Community

github.com/CUQI-DTU/CUQIpy-CIL github.com/CUQI-DTU/CUQIpy-FEniCS github.com/CUQI-DTU/CUQIpy-PyTorch

github.com/CUQI-DTU/CUQIpy-demos

Two software articles in preparation Internal and external projects using CUQIpy Grow community of users and contributors