CUQI  Computational Uncertainty Quantification for Inverse Problems

A brief introduction to the project

sites.dtu.dk/cuqi
About the Villum Investigator Project

Focus

Basic science
 Applied mathematics
 Science and engineering
 Society

CUQI

Research question
How to make UQ a general and easy-to-use tool for inverse problems?

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.
What is an **Inverse Problem?** – Example: X-Ray CT

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.
What is an Inverse Problem? – Example: X-Ray CT

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.

Object $f$
what is inside?
What is an Inverse Problem? – Example: X-Ray CT

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.
What is an Inverse Problem? – Example: X-Ray CT

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.

Mathematical model:

\[ \int_{L_{\theta, s}} f(\xi_1, \xi_2) \, d\ell = g(\theta, s) \]
What is an Inverse Problem? – Example: X-Ray CT

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.

Mathematical model:

\[ \int_{L_{\theta,s}} f(\xi_1, \xi_2) d\ell = g(\theta, s) \]

Data is recorded here

Object

CT scanner

3D reconstruction

Projection data \( g \) from CT scanner look like this

Imaging
What is Uncertainty 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.

*Measured blurred data*  
*UQ gives the mean estimate of the image*

*UQ shows the uncertainty (variance) in the pixels*  
*White = high uncertainty*
Computational UQ in the Project

Make UQ *operational* and *accessible* requires:

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

**Philosophy**

- 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.).
Current Activities

Software

CUQIpy

Nonlinear problems, EIT & photonics

Dimensionality

Machine learning

Non-Gaussian priors

Leadership

Convergence theory

Model errors

Goal-oriented UQ

Prior modeling

Tomography of particles in fusion plasma, Villum Synergy

Enhancement

DTU Physics

CUQI Project – Brief Introduction
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.

**Goal Oriented UQ**: mean estimate of boundaries and their uncertainties

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

Build a software package 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 3\textsuperscript{rd} 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
- [github.com/CUQI-DTU/CUQIpy](https://github.com/CUQI-DTU/CUQIpy)

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!
CUQiPy is built as a Bayesian framework

Generic inverse problem

\[
b = A(x)
\]

Bayesian rule

\[
\pi(x \mid b) \propto \pi(b \mid x) \pi(x)
\]

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

Bayesian approach

samples from the posterior: a range of possible estimates!
Handling two very different inverse problems in CUQIPy

1: Linear inverse problems

\[ b = Ax \]

Application: X-ray CT
Model: line integrals

2: PDE-based inverse problems

\[ b = \mathcal{G}(\theta) \]

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**

**X-ray Computed Tomography**

```
# Step 1: Model
A = ParallelBeam2DProblem(im_size=(N,N), ...).model

# Step 2: Prior
x = GMRF(mean=np.zeros(n), std=delta,
          physical_dim=2, bc_type="zero")

# 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()
```
Analysis of posterior samples

Plotting samples

# plot mean
samples.plot_mean()

# uncertainty plot
samples.plot_std()

# scatter plot
samples.plot_pair(marginals=True)
Defect Detection in Bayesian CT Imaging of Subsea Pipes

CT forward problem
\[ d = A(x + \varepsilon) + e \]

Pipe structure  \quad Small defects

Prior distributions
We use a Structural Gaussian Prior \([1]\) to promote the pipe structure in \(x\):
\[ x \sim \mathcal{N}(\mu_{SGP}, (R_{SGP}^T R_{SGP})^{-1}). \]

There are few defects, so we enforce sparsity in \(\varepsilon\) with a hierarchical prior:
\[ \varepsilon_i | \eta_i \sim \mathcal{N}(0, \eta_i), \quad \eta_i \sim \mathcal{IG}(\frac{\nu}{2}, \frac{\nu}{2} \omega^2), \]

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

Christensen, Riis, Pereyra and Jørgensen (in prep)
“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

• Training material
  github.com/CUQI-DTU/CUQIpy-demos

• Expansion plugins
  – X-ray CT
    github.com/CUQI-DTU/CUQIpy-CIL
  – PDE finite element
    github.com/CUQI-DTU/CUQIpy-FEniCS
  – PyTorch autodiff
    github.com/CUQI-DTU/CUQIpy-PyTorch

• Next steps
  – Publication
    Two software articles in preparation
  – Collaboration
    Internal and external projects using CUQIpy
  – Community
    Grow community of users and contributors