SI2-SSI: Integrating Data with Complex Predictive Models under Uncertainty:

An Extensible Software Framework for Large-Scale Bayesian Inversion

Omar Ghattas[1], **Ki-Tae Kim**[2], **Youssef Marzouk**[3], **Matthew Parno**[4], **Noemi Petra**[2], **Umberto Villa**[5]

[1]Oden Institute for Computational Engineering and Sciences, *The University of Texas at Austin*; [2]Applied Mathematics, *University of California, Merced*; [3]Center for Computational Engineering, *Massachusetts Institute of Technology*; [4]Cold Regions Research and Engineering Lab, *US Army Corps of Engineers*; [5]McKelvey School of Engineering, *Washington University in St. Louis*

Introduction

Needs and motivation

- The **inverse problem** seeks to extract knowledge from data via models, and is a critical precursor to computational prediction with rigorously quantified uncertainties
- Bayesian inference provides a comprehensive and systematic framework for formulating and solving inverse problems under uncertainty
- Bayesian inversion with conventional algorithms and software is prohibitive for complex models and high dimensional parameter spaces
- Intensive research efforts are creating advanced algorithms that exploit the structure of the posterior, resulting in orders of magnitude speedups
- However, these new algorithms have not been made accessible to a broad community of scientists and engineers interested in solving inverse problems

Intellectual merits

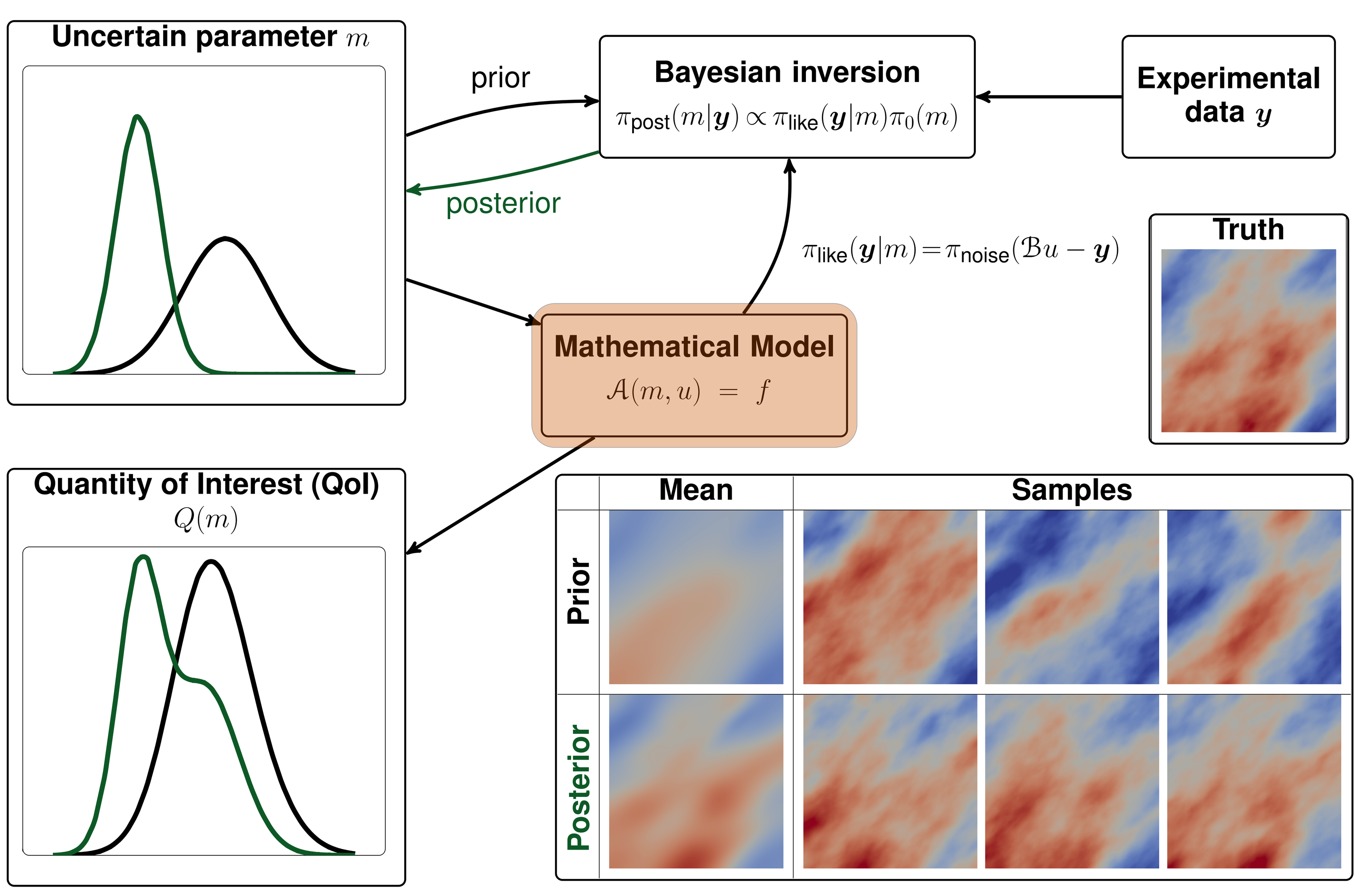
- Develop, deploy, & support robust, scalable, high-performance, open-source software
- Provide reference implementations of advanced Bayesian inversion algorithms
- Enable the solution of Bayesian inverse problems of unprecendent size and realism

Broader impacts

- Facilitate the wider adoption of Bayesian tools in simulation-driven science
- Organize workshops & summer schools and develop short & long courses
- Any scientist interested in integrating data with models to quantify and reduce uncertainties in model predictions is a potential user

Bayesian Formulation of Inverse Problems

- **Goal:** given (noisy, indirect) data and a deterministic or stochastic forward model, infer model parameters and update model predictions
- *Solving* the inverse problem then amounts to *characterizing* the posterior distribution: drawing samples; estimating the mean, covariance, or higher moments; evaluating the posterior probabilities of particular events or quantities of interest

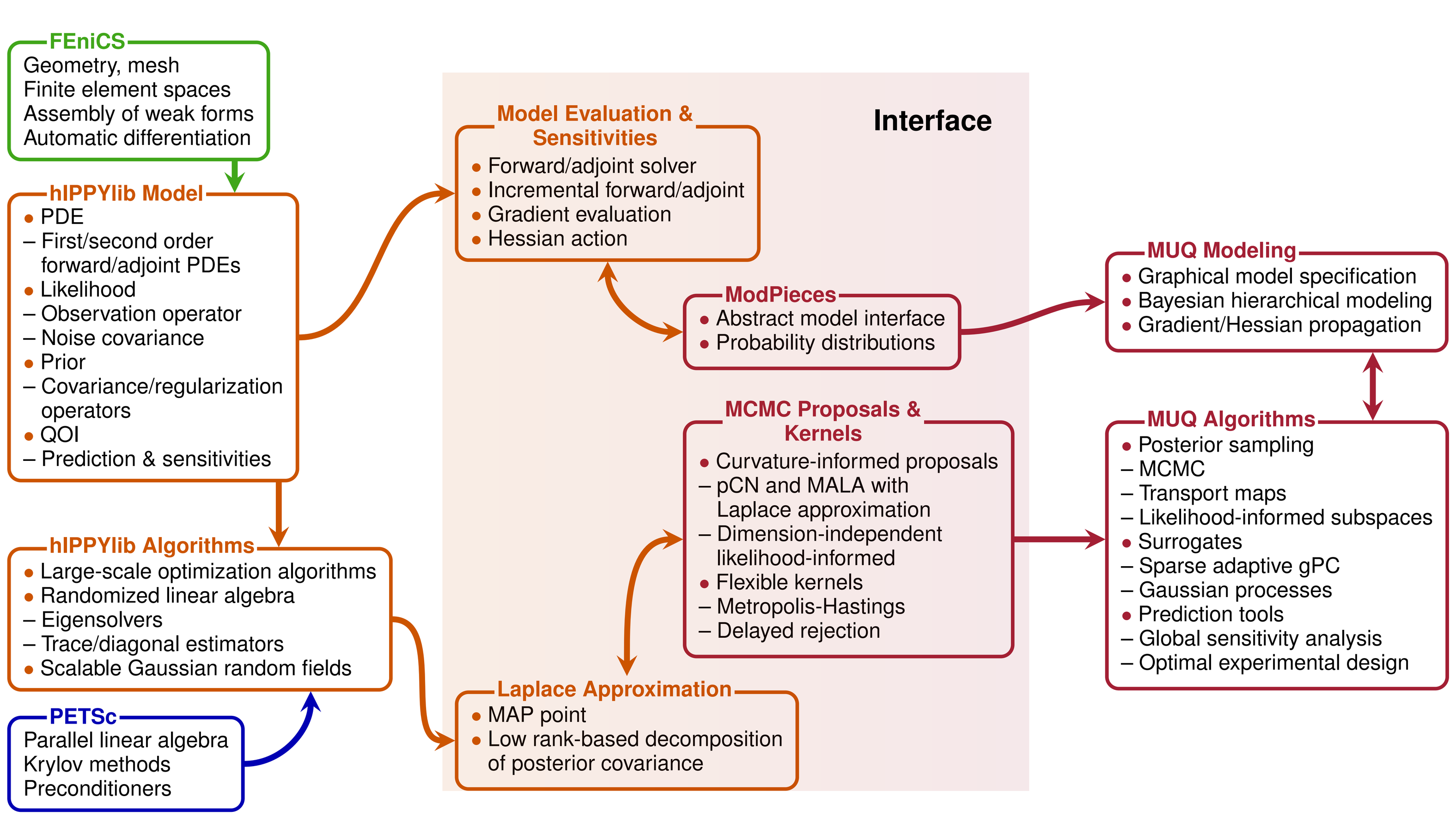


The process of extracting knowledge from data by solving inverse problems

Software Framework

MUQ-hIPPYlib integration

- **Goal:** Make the software *efficient* and *scalable* as well as *easy to use* at all user levels
- Software abstractions to follow mathematical abstraction
- Highly composable modeling framework that encourages code reuse, facilitates construction of complex or hierarchical models, allows for algorithmic innovation

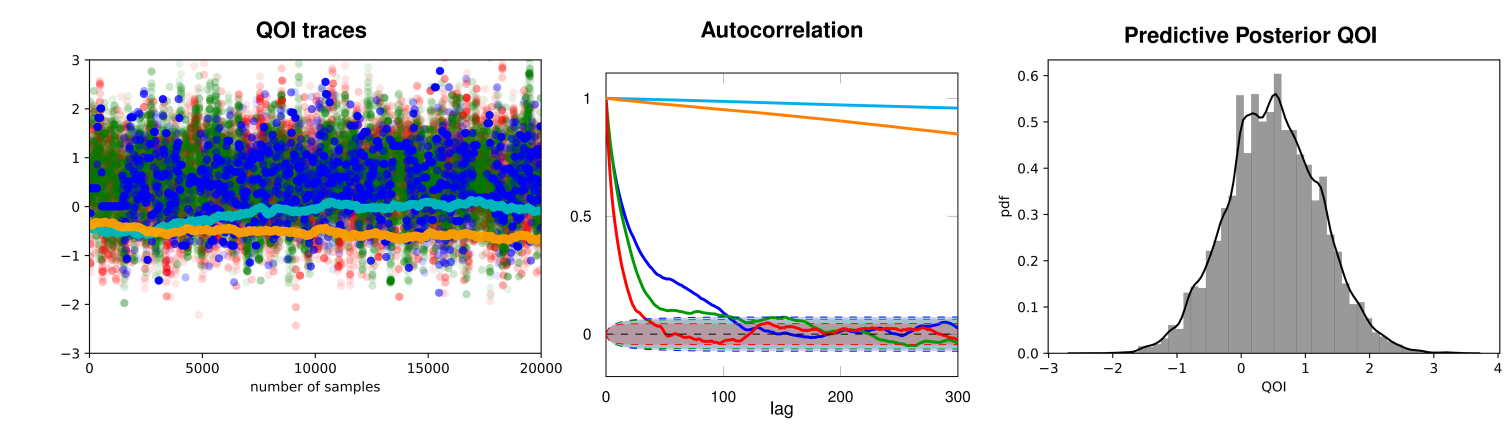


MUQ and hIPPYlib have complementary capabilities that, combined, provide a unique software framework for large-scale Bayesian inversion

Single-phase subsurface flow example

Goal: Infer the log-permeability field from pressure data and compute the predictive posterior distribution of the effective log-permeability of the medium (QOI)

- The parameter, state, and data spaces have dimensions 1,089, 4,225, and 300, respectively
- Each evaluation of the likelihood function requires discretizing and solving a PDE
- Sampling methods that exploit curvature information (h-pCN and h-MALA) show up to 2 orders of magnitude speed up w.r.t. standard methods (pCN and MALA)
- Delayed Rejection (DR) kernel allows us to further improve the mixing of the MCMC chains by combining different proposals



Method	Predicted QOI	AR (%)	IAT	ESS	ES/min
• **pCN** (0.01)	–	7.2	4416.0	4.5	0.2
• **MALA** (1E-5)	–	30.6	2544.8	7.9	0.1
• **h-pCN** (0.9)	0.477	10.6	69.3	288.5	**16.9**
• **h-MALA** (0.1)	0.541	53.5	53.5	373.8	6.0
• **DR h-pCN** (1.0)/ **h-MALA** (0.1)	0.491	**59.5**	**24.8**	**806.8**	10.3

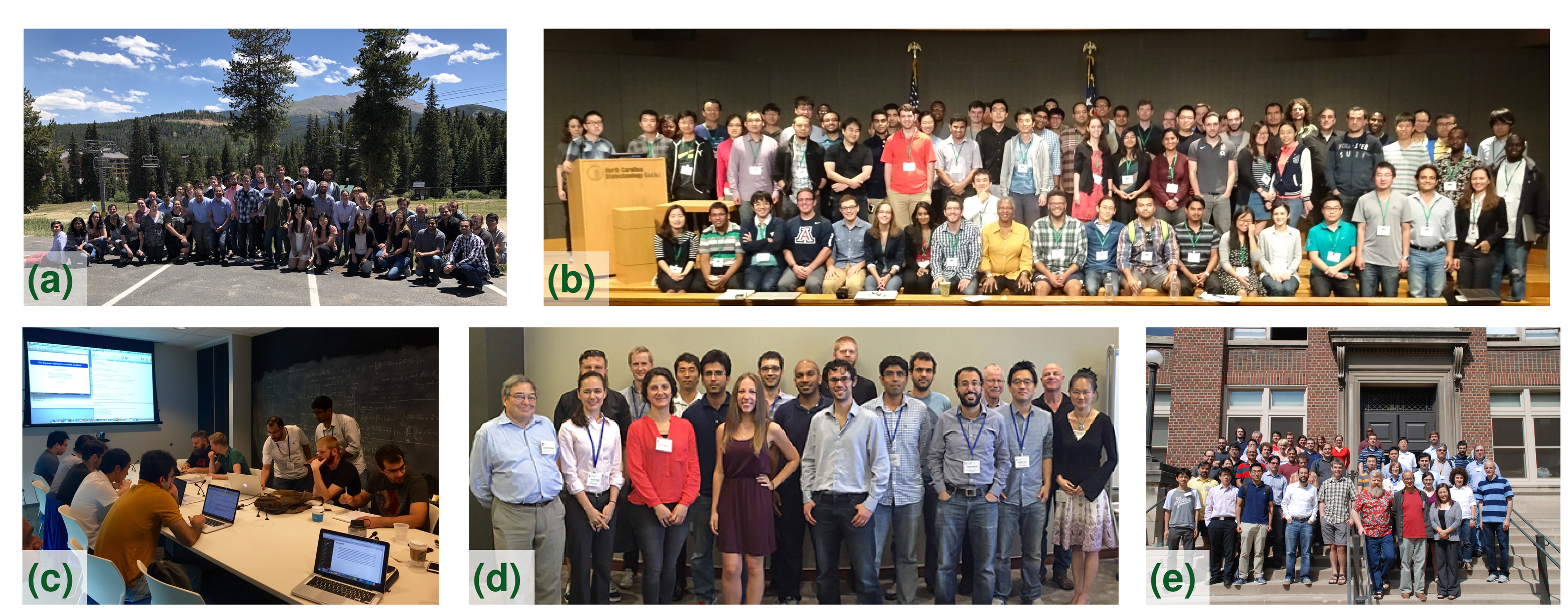
Research, Education and Outreach

Research

Several projects at Army Corps of Engineers, MIT, NYU, NC State, University of Heidelberg (Germany), UC Merced, UT Austin, and WashU are using MUQ-hIPPYlib, including:

- Inversion for coupled flow–geomechanics to understand induced seismicity (MIT)
- Design of experiments, optimization, and learning for complex systems under uncertainty (MIT, UT)
- Inversion and prediction of ice sheet dynamics (UC Merced, WashU, MIT)
- Accounting for model errors in inverse problems (UC Merced)
- Statistical treatment of inverse problems constrained by stochastic models (UC Merced)
- Inversion and control for CO_2 sequestration with poroelastic models (UT)
- Goal-oriented inference for reservoir models (UT)
- Inferring functional tissue properties from photoacoustic tomography data (WashU)

Workshops, summer schools, and short & long courses



- *Short course on Inverse Problems and Uncertainty Quantification*, Trimester on Mathematics of Climate and the Environment, Institut Henri Poincaré, Paris, France, November 4–8, 2019
- *Bayesian Inverse Problems for Structural Health Monitoring of Civil Infrastructure*, Dartmouth College, Hanover, NH, May 21–24, 2019
- *Inverse Problems: Systematic Integration of Data with Models under Uncertainty*, 2018 Gene Golub SIAM Summer School, Breckenridge, CO, June 17–30, 2018 (a)
- *hIPPYlib: An Extensible Software Framework for Large-Scale Deterministic and Linearized Bayesian Inverse Problems*, SAMSI Summer School, SAMSI, Research Triangle Park, NC, 8–12 August, 2016 (b)
- *QUEST Uncertainty Quantification Summer School*, USC, 19–21 August, 2015
- *Inverse Problems and Uncertainty Quantification*, ICERM IdeaLab, Brown University, 6–10 July, 2015 (c,d)
- *Introduction to Uncertainty Quantification*, IMA Short Course, University of Minnesota, 15–26 June, 2015 (e)
- Graduate-level courses on inverse problems and uncertainty quantification at Dartmouth College, MIT, NC State, NYU, UC Merced, UT Austin, and WashU

Code repositories & infrastructure

- `http://muq.mit.edu`
- `http://hippylib.github.io`

Python, C++, Jupyter, Scipy, Eigen, FEniCS, PETSc, Docker, Github, Bitbucket

Acknowledgement

This work was partially supported by National Science Foundation grants ACI-1550487, ACI-1550547, and ACI-1550593.