Intro to Uncertainty Quantification

Forward UQ, basic methods, UM-Bridge, containers

Linus Seelinger

Institute for Mathematics, Heidelberg.

Mathematical models

Will this CO2 storage leak in 500 years?
Is this patient's aneurysm critical or not?

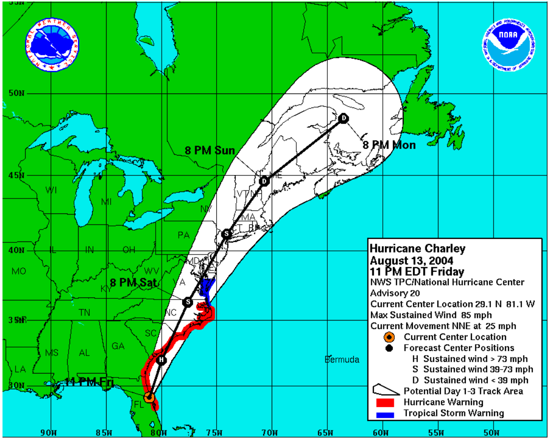
Models represent physical processes $\implies$ simulation experiments, digital twins, ...

Often expressible as map taking parameter vector to model outcome:

$$F : \mathbb{R}^n \to \mathbb{R}^m$$

$\rightarrow$ Space model

Why Uncertainty Quantification (UQ)?



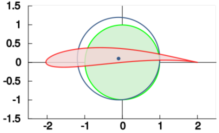
- "Don't focus on the skinny black line"
  - US Hurricane Center
- Uncertain data $\Longrightarrow$ uncertain prediction / inferences.

  UQ: Quantify this!

Forward problems

Given a parameter distribution, how likely is a model outcome? And how likely is a resulting quantity of interest?

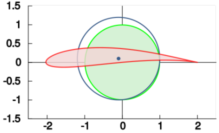
$$\underbrace{\theta}_{\text{Parameter } \theta \sim \pi}$$



Forward problems

Given a parameter distribution, how likely is a model outcome? And how likely is a resulting quantity of interest?

$$\underbrace{\theta}_{\text{Parameter } \theta \sim \pi} \quad \rightarrow \quad \underbrace{F(\theta)}_{\text{Model outcome}}$$

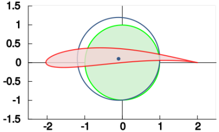


$\rightarrow$

Forward problems

Given a parameter distribution, how likely is a model outcome? And how likely is a resulting quantity of interest?

$$\underbrace{\theta}_{\text{Parameter } \theta\sim\pi} \rightarrow \underbrace{F(\theta)}_{\text{Model outcome}} \rightarrow \underbrace{Q(F(\theta))}_{\text{Quantity of interest}}$$



$\rightarrow$

$\rightarrow$

?

Inverse Problems

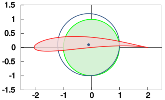
Given observations, how likely is a parameter? And how likely is a resulting quantity of interest?

$$\underbrace{y}_{\text{Observation}}$$

Inverse Problems

Given observations, how likely is a parameter? And how likely is a resulting quantity of interest?

$$\underbrace{F^{-1}(y)}_{\text{Inverse model}} \leftarrow \underbrace{y}_{\text{Observation}}$$



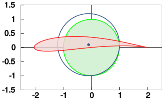
$\leftarrow$

Inverse Problems

Given observations, how likely is a parameter? And how likely is a resulting quantity of interest?

$$\underbrace{Q(F^{-1}(y))}_{\text{Quantity of interest}} \leftarrow \underbrace{F^{-1}(y)}_{\text{Inverse model}} \leftarrow \underbrace{y}_{\text{Observation}}$$

? $\leftarrow$



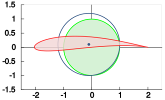
$\leftarrow$

Inverse Problems

Given observations, how likely is a parameter? And how likely is a resulting quantity of interest?

$$\underbrace{Q(F^{-1}(y))}_{\text{Quantity of interest}} \leftarrow \underbrace{F^{-1}(y)}_{\text{Inverse model}} \leftarrow \underbrace{y}_{\text{Observation}}$$

? $\leftarrow$



$\leftarrow$

Cannot invert F, therefore Bayesian problem!

How to solve UQ problems?

Many methods:

- MC / MCMC
- Stochastic Galerkin
- Optimization-based MAP point search
- Multilevel / Multiindex MC / MCMC
- ...

Large-scale problems need HPC!

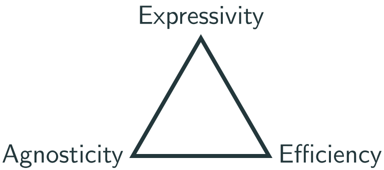


Figure 1: Main aspects of UQ methods

$\rightarrow$ Monte Carlo, QMC

Linking UQ and Model

UQ and Model in Math

Model in UQ: (Often) Just a function $F : \mathbb{R}^n \to \mathbb{R}^m$ with some of the following:

- Model evaluation $F(\theta)$,
- Gradient $v^\top J(\theta)$,
- Jacobian action $J(\theta)v$,
- Hessian action $H(\theta)v$.

$\to$ Simple, model-agnostic interface!

Model **software** and UQ **software**: Not so easy!

Complex software stack, conflicts (buildsystems, dependencies, languages, parallelization), need experts from both sides, ...

UQ and Model in Math

Model in UQ: (Often) Just a function $F : \mathbb{R}^n \to \mathbb{R}^m$ with some of the following:

- Model evaluation $F(\theta)$,
- Gradient $v^\top J(\theta)$,
- Jacobian action $J(\theta)v$,
- Hessian action $H(\theta)v$.

$\rightarrow$ Simple, model-agnostic interface!

Model **software** and UQ **software**: Not so easy!

Complex software stack, conflicts (buildsystems, dependencies, languages, parallelization), need experts from both sides, ...

UM-Bridge: Model Abstraction in Software

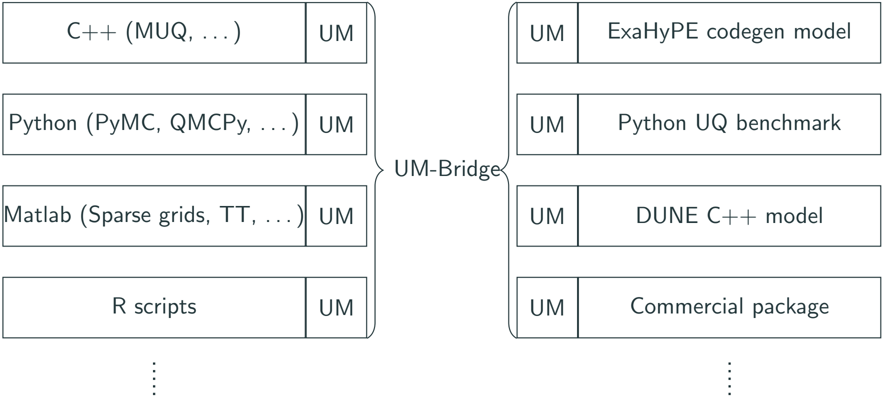
Interface mimics math (Pointwise eval. of F, derivatives optional)

Inspired by microservices (established in industry)

Achieves:

- Level 3: Turn-key HPC setup in the cloud
- Level 2: Portable, reproducible models via containers; Separation of concerns
- Level 1: Coupling across languages

UM-Bridge: Bridging Languages and Frameworks



Requires only HTTP and JSON support → almost every language

Existing integrations for various languages and frameworks

$\rightarrow$ Quick demo

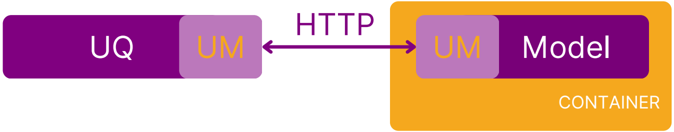
Exercise

UM-Bridge docs, Tutorial 1: First steps

`https://um-bridge-benchmarks.readthedocs.io`

Containerized Models

UM-Bridge: Containerization - Portable Models



- Run tsunami model as easy as

  `docker run -p 4242:4242 linusseelinger/model-exahype-tsunami`
- Evaluate model in python:

```
model = umbridge.HTTPModel('localhost:4242', 'forward')
model([[0.1,0.4]])
```

→ Separation of concerns!

UQ Benchmarks

UQ Benchmarks

Navigation

Quickstart Guide
Analytic-Gaussian-Mixture Benchmark
ExaHyPE-Tsunami Benchmark
Inferring material properties of a cantilevered beam
Analytic-Banana Benchmark
Analytic-Donut Benchmark
Analytic-Funnel Benchmark
ExaHyPE-Tsunami Model
Euler-Bernoulli Beam

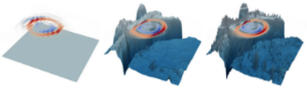
Quick search

Go

ExaHyPE-Tsunami Model

Overview

In this benchmark we model the propagation of the 2011 Tohoku tsunami by solving the shallow water equations. For the numerical solution of the PDE, we apply an ADER-DG method implemented in the ExaHyPE framework. The aim is to obtain the parameters describing the initial displacements from the data of two available buoys located near the Japanese coast



Authors

- Anne Reinarz

Run

```
docker run -it -p 4243:4243 linusseelinger/model-exahype-tsunami
```

Properties

Mapping	Dimensions	Description
inputSizes	[2]	x and y coordinates of a proposed tsunami origin
outputSizes	[1]	Arrival time and maximum water height at two buoy points

Feature	Supported
Evaluate	True
Gradient	False
ApplyJacobian	False
ApplyHessian	False

Config	Type	Default	Description
level	int	0	chooses the model level to run (see below for fur-

- Community project:
  - > 20 models and benchmarks,
  - > 15 contributors from
  - > 10 institutions
- Ready-to-run containers
- Automated builds, testing etc.

Exercise

UM-Bridge docs, Tutorial 2: Model containers

`https://um-bridge-benchmarks.readthedocs.io`

Exercise

Run space model:

`docker run -it -p 4242:4242 linusseelinger/kcds-space`

Look at its output in `explore_model.ipynb` from
`https://github.com/linusseelinger/kcds`.

Exercise

Forward UQ question: With uniformly dist. launch parameter

$$\theta \sim \pi = \mathcal{U}([3.4, 1.0] \times [3.5, 1.2]),$$

what is the probability of hitting moon (i.e. $\mathbb{E}_\pi[F_3]$ / model output index 2)?

- Adapt QMCPy example to this problem
- Write custom Monte Carlo code, any language! (draw random $\theta_i \sim \pi$, compute mean of $F_3(\theta_i)$).
- Change to a Normal distribution $\pi = \mathcal{N}((3.45, 1.1)^\top, 0.1I)$
- Replace model by custom / modified one

$\rightarrow$ Demo: building a container

Exercise

Things to try:

- Containerize and run given space model
- Containerize and run custom model from before
- Upload your image to dockerhub, share with colleagues
  Note: `docker login` to authenticate, `docker push` to upload, image name must be `dockerhubusername/arbitrary`.