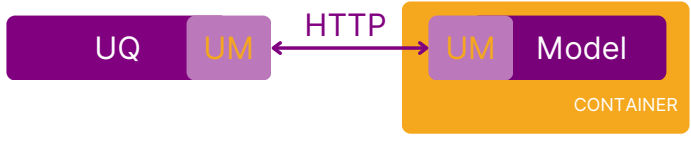
Introduction to UM-Bridge

UM-Bridge Workshop 2025

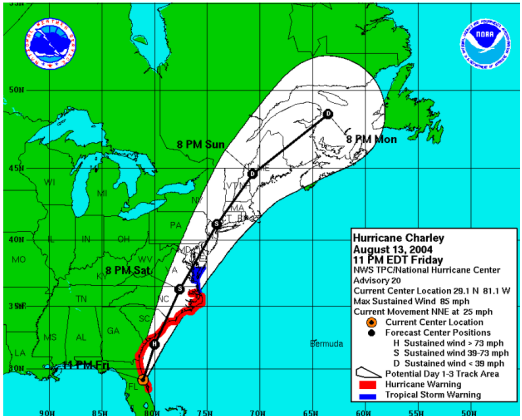
Anne Reinarz[1] (Department of Computer Science, Durham University)
October 17 2025

[1] anne.k.reinarz@durham.ac.uk

Motivation



Why Uncertainty Quantification (UQ)?



- "Don't focus on the skinny black line"
  - US Hurricane Center
- Uncertain data leads to uncertain prediction / inferences.
  $\Rightarrow$ Quantify this!

How to solve UQ problems?

Many methods:

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

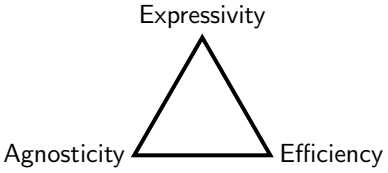


Figure: Main aspects of UQ methods

UQ and Model in Math

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

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

$\to$ Simple, model-agnostic interface!

UQ and Model in Math

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

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

$\to$ Simple, model-agnostic interface!

Model **software** and UQ **software**: Not so easy!
Conflicts in buildsystems, dependencies, languages, parallelization; need experts from both sides, ...

UQ and Model in Math

Figure: Monolithic coupling between model and UQ.

Model **software** and UQ **software**: Not so easy!
Conflicts in buildsystems, dependencies, languages, parallelization; need experts from both sides, ...

UQ and Model in Math

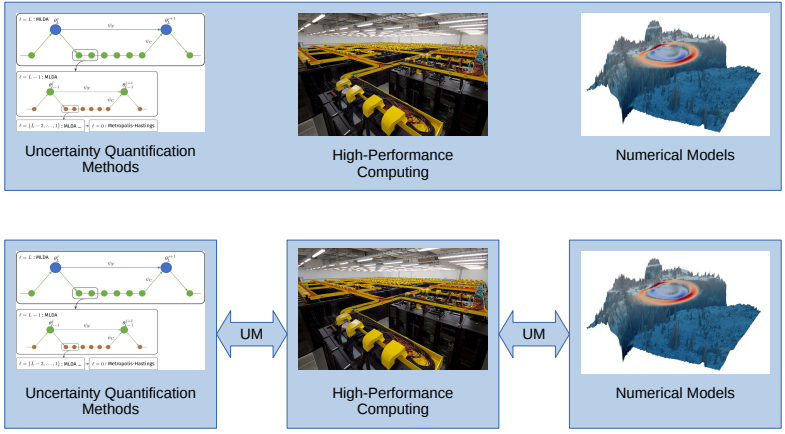


Figure: Lightweight coupling between model and UQ.

UM-Bridge: Model Abstraction in Software

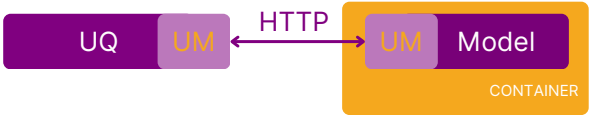


Figure: Network and container based coupling of UQ and model codes that requires minimal extension of software on each side.

UM-Bridge: Bridging Languages and Frameworks

Language / framework	Client support	Server support
C++	✓	✓
MATLAB	✓	✗
Python	✓	✓
R	✓	planned
julia	✓	✓
emcee	✓	✗
MUQ	✓	✓
PyMC	✓	✗
QMCPy	✓	✗
Sparse Grids MATLAB Kit	✓	✗
tinyDA	✓	✗
TT Toolbox	✓	✗

Brief Demo

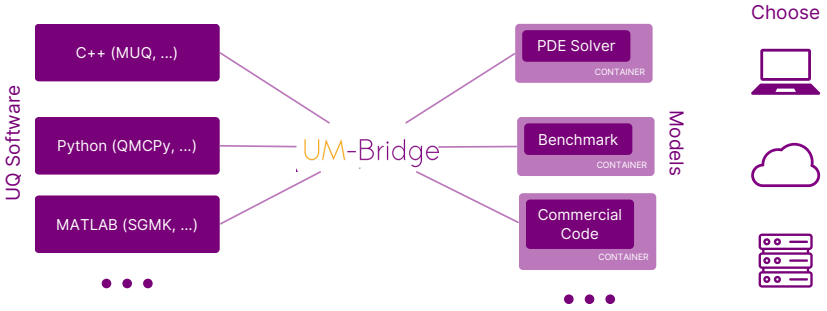
https://um-bridge-benchmarks.readthedocs.io/en/docs/models/exahype-tsunami.html

Brief Demo

`https://um-bridge-benchmarks.readthedocs.io/en/docs/models/exahype-tsunami.html`

- Running a model as easy as
  `docker run -p 4242:4242 linusseelinger/model-exahype-tsunami`
- Details on setting up containerised models to follow

UM-Bridge: Enabling UQ from Prototype to HPC



- Single point of entry for UQ and Models
- Separation of concerns
- Straightforward scalability

UM-Bridge: Client

Connect to model

```
import umbridge
model = umbridge.HTTPModel("http://localhost:4242", "forward")
```

Display input / output dimensions

```
print(model.get_input_sizes( ))
print(model.get_output_sizes( ))
```

Evaluate model

```
print(model([[0.0, 10.0]]))
```

Optionally, pass configuration options

```
print(model([[0.0, 10.0 ]] , {"level": 0}))
```

UM-Bridge: Server

Define model class

```
TestModel(umbridge.Model):
    def get_input_sizes(self):
        # Number and dimensions of input vectors
        return [1]
    def get_output_sizes(self):
        # Number and dimensions of output vectors
        return [1]
    def call(self, parameters, config ={}):
        output = parameters[0][0]*2 # Do something with the input
        return [[output]]
    def supports_evaluate(self):
        return True
```

UM-Bridge: Server

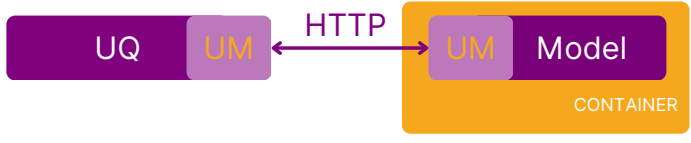
Define model class

```
TestModel(umbridge.Model):
    def get_input_sizes(self):
        # Number and dimensions of input vectors
        return [1]
    def get_output_sizes(self):
        # Number and dimensions of output vectors
        return [1]
    def call(self, parameters, config ={}):
        output = parameters[0][0]*2 # Do something with the input
        return [[output]]
    def supports_evaluate(self):
        return True
```

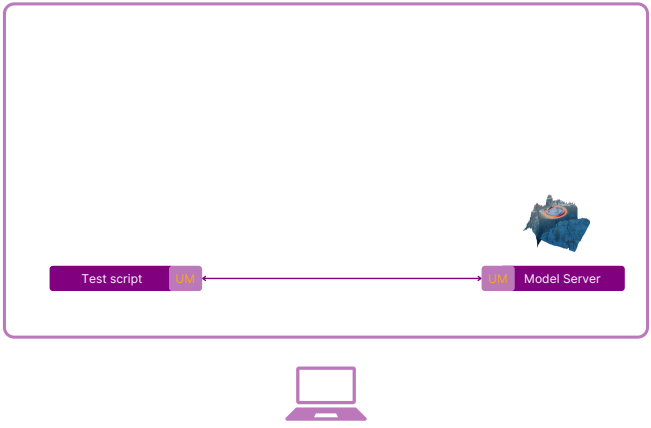
Serve model via HTTP:

```
testmodel = TestModel( )
umbridge.servemodel(testmodel , 4242)
```

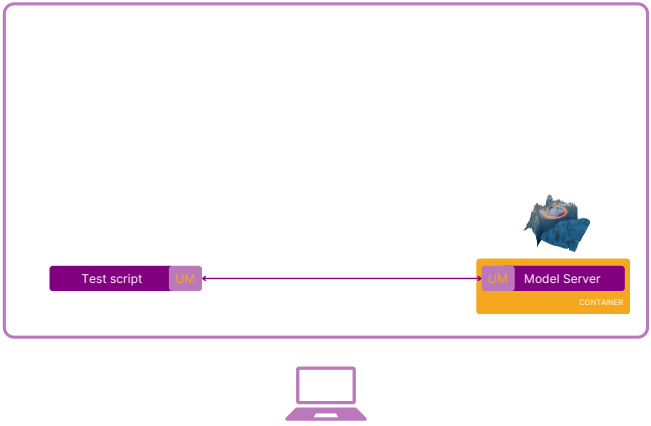
Workflows



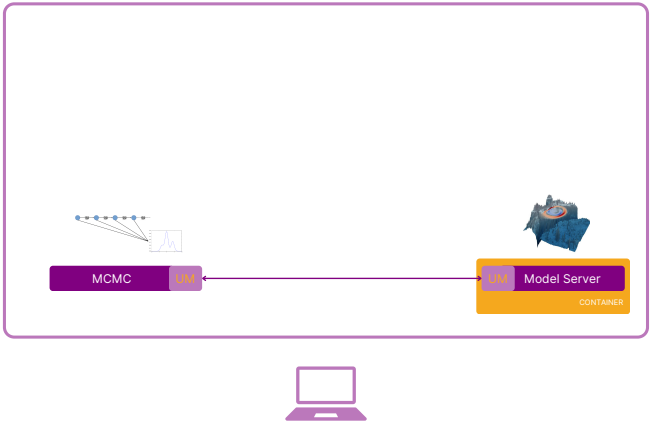
Workflow



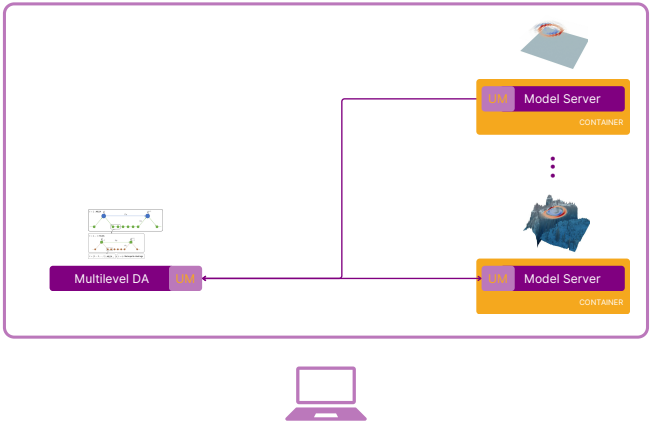
Workflow



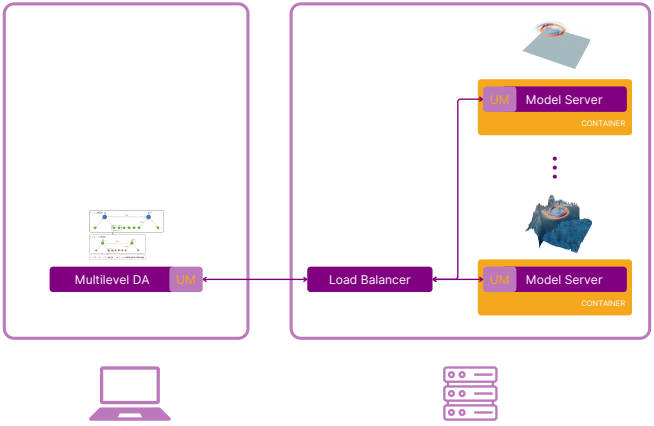
Workflow



Workflow


UM
Model Server
CONTAINER
UM
Model Server
CONTAINER
Multilevel DA
UM

Workflow


UM
Model Server
CONTAINER
Multilevel DA
UM
Load Balancer
UM
Model Server
CONTAINER

Over to you

Tutorial section 1-3
`https://um-bridge-benchmarks.readthedocs.io/en/docs/tutorial.html`

References

[1] L. Seelinger, A. Reinarz, L. Rannabauer, M. Bader, P. Bastian, R. Scheichl.
High performance uncertainty quantification with parallelized multilevel Markov chain Monte Carlo.
Proc. of the Int. Conf. for High Performance Computing, 2021.

[2] L. Seelinger, V. Cheng-Seelinger, A. Davis, M. Parno, A. Reinarz.
UM-Bridge: Uncertainty quantification and modeling bridge.
Journal of Open Source Software, 8(83), 2023.

[3] L. Seelinger, A. Reinarz et al.
Democratizing uncertainty quantification.
Journal of Computational Physics, 521:113542, 2024.

[4] C.M. Loi, A. Reinarz, L. Seelinger, W. Hornsby, J. Buchanan, M. Lykkegaard.
A Performance Analysis of Task Scheduling for UQ Workflows on HPC Systems.
ISC High Performance 2025 Research Paper Proceedings.