PUQ + surmise

Özge Sürer

BAND-Nuclear Data Workshop, December 2024

Collaborators

Stefan Wild (Berkeley Lab)

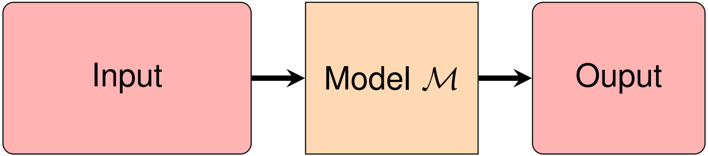
Matt Plumlee (Amazon)

Moses Chan (Northwestern U.)

Simulation Models

A simulation model is usually implemented in computer codes
Used to predict, analyze, and optimize system performance without direct experimentation

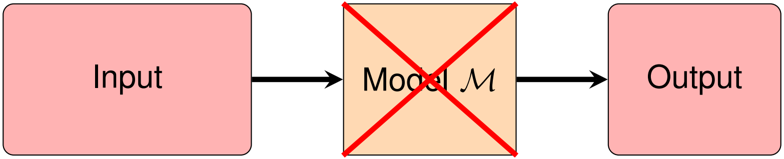
Simulation runs require significant computation time
More complex phenomena demand more time-consuming simulations



Simulation Models

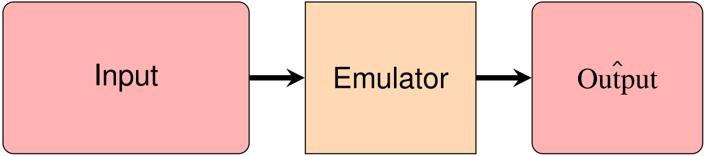
A simulation model is usually implemented in computer codes
Used to predict, analyze, and optimize system performance without direct experimentation

Simulation runs require significant computation time
More complex phenomena demand more time-consuming simulations



Statistical Model Emulation

- We build a cheaper emulator as a proxy to a simulation model $\mathcal{M}$



- Emulators are built with statistical/machine learning techniques (e.g., Gaussian Processes)
  - We first run the model at a few well-chosen input values (e.g., simulation/computer design) and obtain their outputs
  - We build GPs with the simulation data to approximate the output values

BAND Software: `surmise`

`surmise` is a

- interface between emulators, and calibration, uncertainty quantification
- tool to benchmark many methods
- extensible for a variety of application problems

Resources and Links:

- Project package: https://github.com/bandframework/surmise
- Documentation: http://surmise.readthedocs.io
- Example Tutorial using physics code Fresco: https://github.com/bandframework/bandframework/blob/main/software/Bfrescox/Tutorial_I/tutorial1.rst
- BAND SDK compatibility

BAND Software: `surmise`

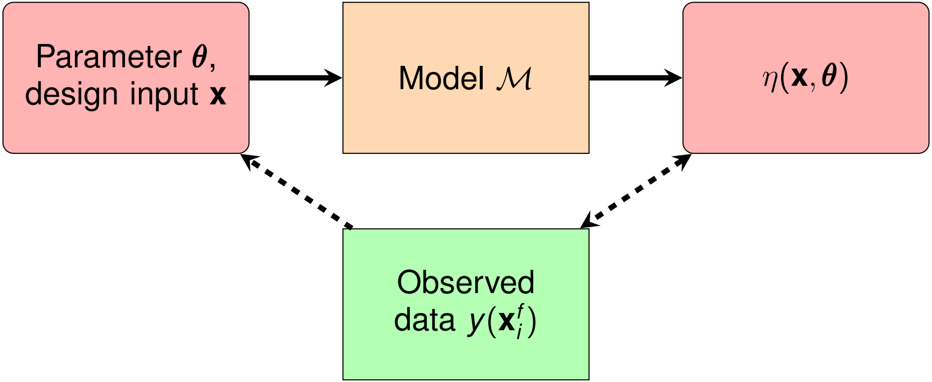
Use an existing method to emulate a computationally expensive code:

```
    emulator(x, theta, f, method="PCGP", args)
```

Use an external package:

```
    emulator(x, theta, f, method="GPy", args)
```

Model Calibration



Observation $y(\mathbf{x}_i^f)$ can be modeled using the simulation $\eta(\mathbf{x}_i^f, \boldsymbol{\theta})$

$$y(\mathbf{x}_i^f) = \eta(\mathbf{x}_i^f, \boldsymbol{\theta}) + \epsilon, \qquad \epsilon \sim \mathrm{N}(0, \sigma^2)$$

Model Calibration

Posterior density is computed by using Bayes' rule given $\mathbf{y} = \left(y\left(\mathbf{x}_1^f\right), \ldots, y\left(\mathbf{x}_d^f\right)\right)^\top$

$$\underbrace{p(\boldsymbol{\theta}|\mathbf{y})}_{\text{posterior}} \propto \tilde{p}(\boldsymbol{\theta}|\mathbf{y}) = \underbrace{p(\mathbf{y}|\boldsymbol{\theta})}_{\text{likelihood}} \underbrace{p(\boldsymbol{\theta})}_{\text{prior}}$$

where

$$p(\mathbf{y}|\boldsymbol{\theta}) = (2\pi)^{-d/2}|\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\eta}(\boldsymbol{\theta}))^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\eta}(\boldsymbol{\theta}))\right),$$

and $\boldsymbol{\eta}(\boldsymbol{\theta}) = \left(\eta\left(\mathbf{x}_1^f, \boldsymbol{\theta}\right), \ldots, \eta\left(\mathbf{x}_d^f, \boldsymbol{\theta}\right)\right)^\top$.

⚠ **Posterior depends on potentially expensive simulation $\eta(\mathbf{x}, \theta)$** ⚠

Use an existing method to calibrate:

```
calibrator(emu, y, x, thetaprior, method="directbayes",
    args)
```

Use an existing method to calibrate with another sampler:

```
calibrator(emu, y, x, thetaprior, method="directbayes",
    args={"sampler": LMC})
```

Use an existing method to benchmark:

```
calibrator(emu, y, x, thetaprior,
    method="directbayeswoodbury", args)
```

Designs for Simulation Experiments

- Simulation models are expensive to run and we can only afford a small number of evaluations
- How do we choose the input values at which to evaluate our simulation model?
- We need to choose the "best possible" input values to evaluate the simulation model

One drawback of space-filling designs is that the (unknown) input region of interest may not be adequately explored especially in high-dimensional spaces

BAND Software: PUQ

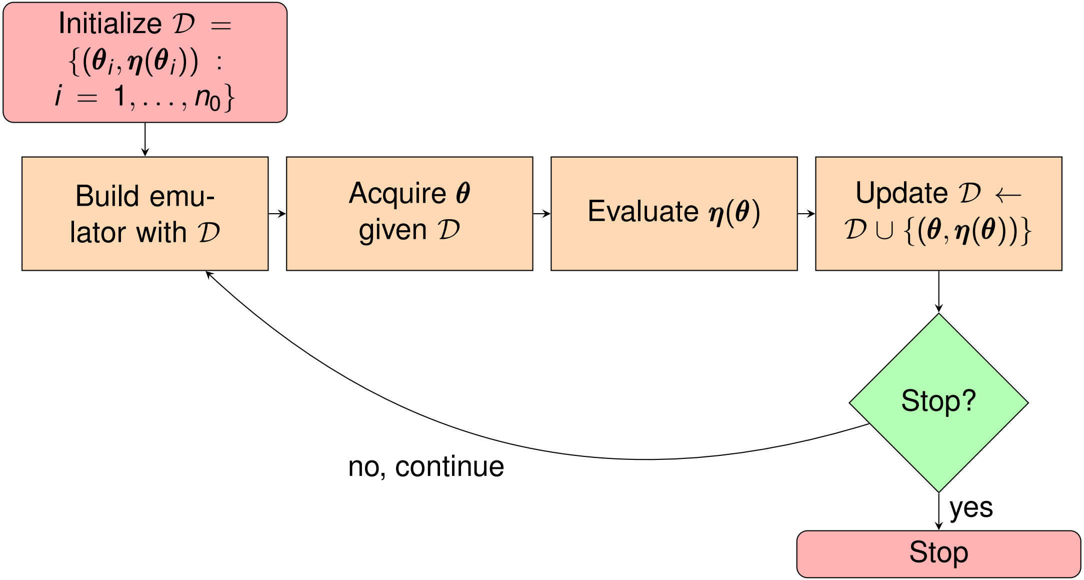
PUQ offers

- novel experimental design techniques with intelligent selection criteria
- tools for data collection to enhance the efficiency and effectiveness of uncertainty quantification
- features parallel implementations

Resources and Links:

- Project package: https://github.com/parallelUQ/PUQ
- Documentation: https://puq.readthedocs.io/en/latest/
- Example Tutorial using physics code Fresco: https://github.com/parallelUQ/PUQ/tree/main/examples/fresco_example
- BAND SDK compatibility

Sequential Bayesian Experimental Design



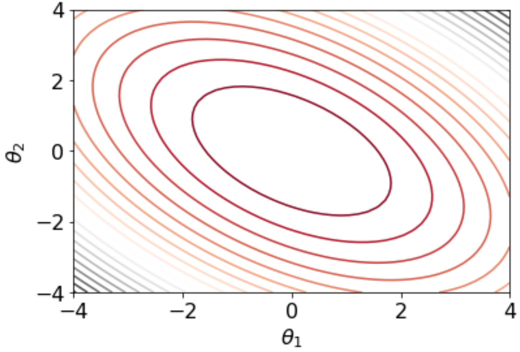
Acquisition function

Next input is chosen to minimize (approximately) the acquisition function $\mathcal{A}_t$ such that

$$\boldsymbol{\theta}^{\text{new}} \in \underset{\boldsymbol{\theta}^* \in \mathcal{L}_t}{\arg\min}\, \mathcal{A}_t(\boldsymbol{\theta}^*)$$

where

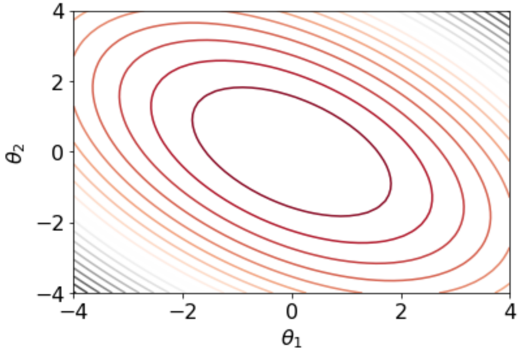
$$\mathcal{A}_t(\boldsymbol{\theta}^*) = \int_{\Theta} \mathbb{E}_{\boldsymbol{\eta}^*|\mathcal{D}_t} \left(\mathbb{V}[p(\boldsymbol{\theta}|\mathbf{y}) \,|(\boldsymbol{\theta}^*, \boldsymbol{\eta}^*) \cup \mathcal{D}_t] \right) d\boldsymbol{\theta}$$



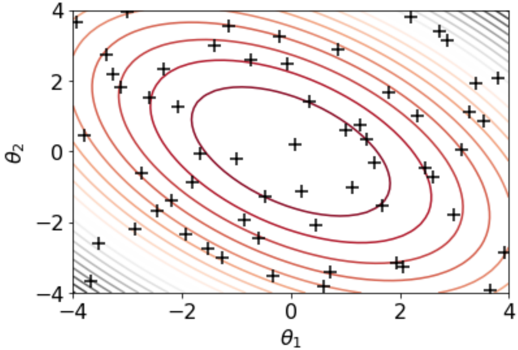
Simulation model $\mathcal{M}$

$$\eta(\boldsymbol{\theta}) = \theta_1^2 + \theta_1\theta_2 + \theta_2^2$$

Motivation

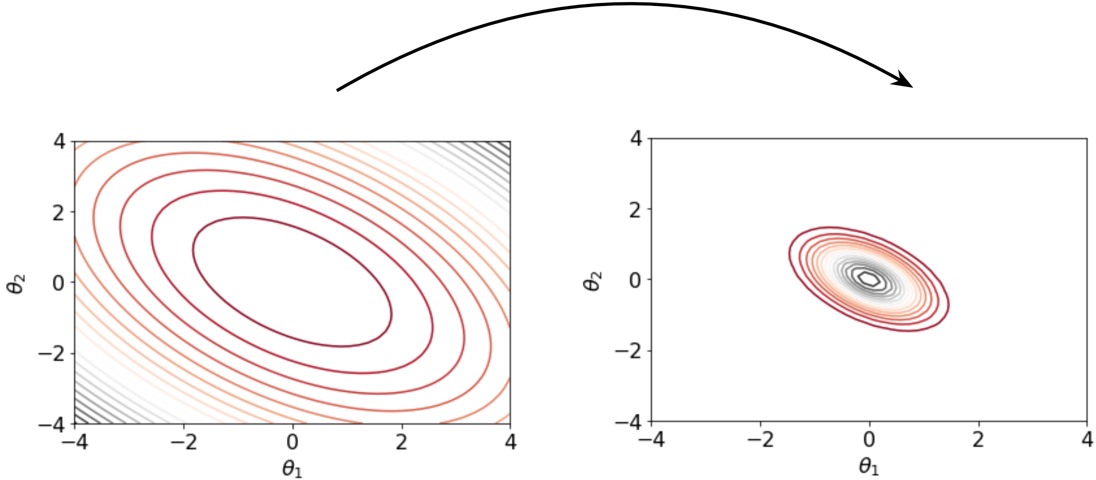


Simulation model $\mathcal{M}$

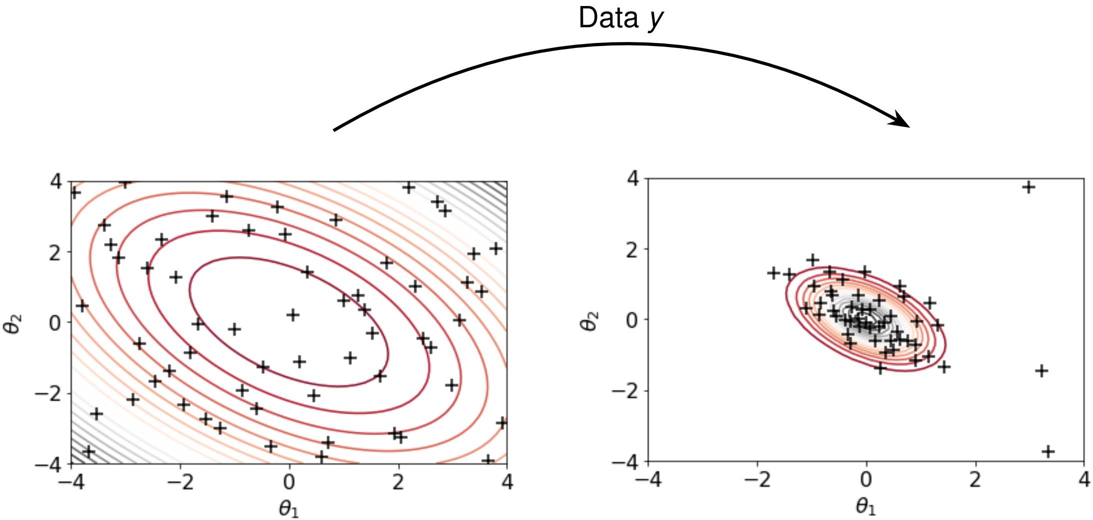


Space-filling design

Data y



Simulation model $\mathcal{M}$ Posterior density



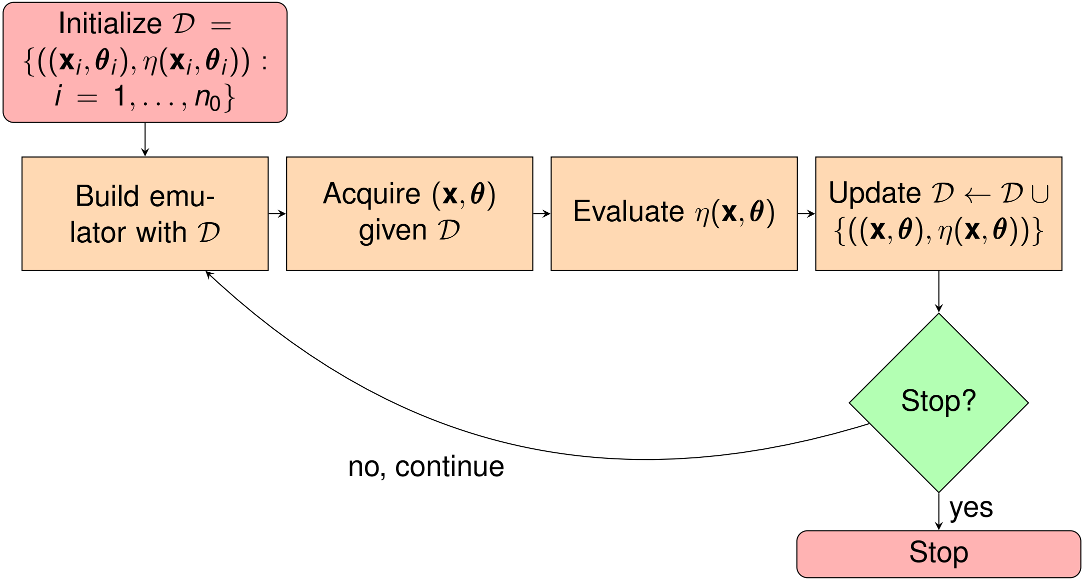
Simulation model $\mathcal{M}$ Posterior density

This work builds on research by Sürer, Plumlee, and Wild (2024).

```
        designer(
            data_cls,
            method="SEQCAL",
            args={
                "mini_batch": 1,
                "n_init": 10,
                "nworkers": 2,
                "AL": "eivar",
                "prior": prior_func,
                "max_evals": 210
                },
            )
```

Sürer, Plumlee, and Wild, Sequential Bayesian Experimental Design for Calibration of Expensive Simulation Models, *Technometrics* (2024),
https://doi.org/10.1080/00401706.2023.2246157

Sequential Bayesian Experimental Design

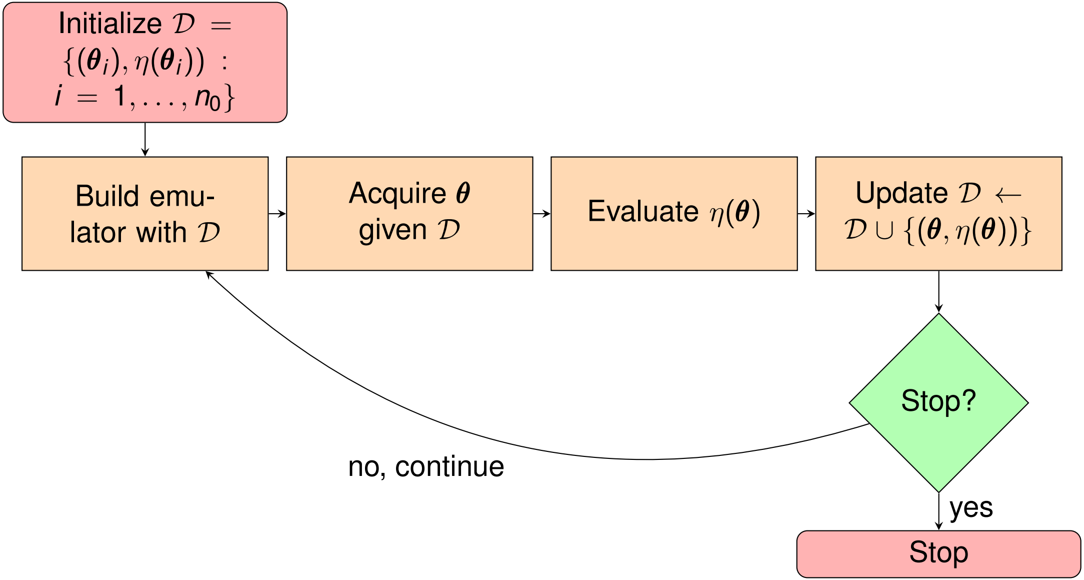


This work builds on research by Sürer (2024).

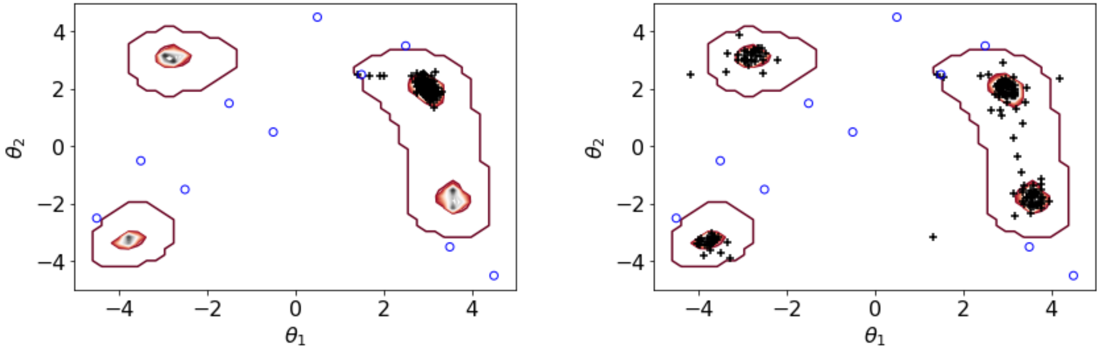
```
        designer(data_cls,
            method="SEQDES",
            args={
                "mini_batch": 1,
                "nworkers": 2,
                "n_init": 10,
                "AL": "ceivar",
                "prior": priors,
                "max_evals": 30,
                },
            )
```

Sürer, Simulation Experiment Design for Calibration via Active Learning, *Journal of Quality Technology* (2024), https://doi.org/10.1080/00224065.2024.2391780

Sequential Bayesian Experimental Design



Illustration



This work builds on research by Sürer and Wild (2024).

```
designer(data_cls,
    method="SEQCALOPT",
    args={
        "mini_batch": 1,
        "nworkers": 2,
        "AL": "hybrid_ei",
        "prior": prior_func,
        "max_evals": 100,
    },
)
```

Sürer and Wild, An Active Learning Performance Model for Parallel Bayesian Calibration of Expensive Simulations, *NeurIPS 2024 Workshop on Bayesian Decision-making and Uncertainty*.

Sürer and Wild, Performance Analysis of Sequential Experimental Design for Calibration in Parallel Computing Environments, (2024+), https://arxiv.org/pdf/2412.00654

Questions?