Uncertainty Propagation in (large-scale) Networks via Domain Decomposition

SAND2016-9012PE

S. Guzzetti
K. Carlberg, M. Khalil, K. Sargsyan

EMORY UNIVERSITY

Sandia National Laboratories

September 16, 2016

Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

The team

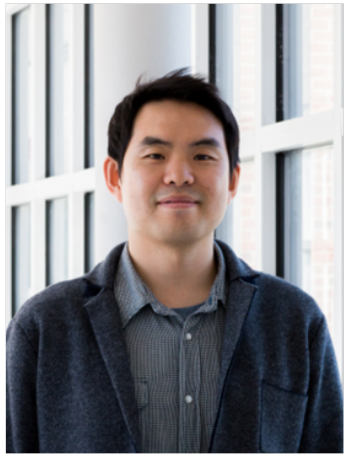
K. Carlberg

M. Khalil

K. Sargsyan

C. H. Ozen

J. Jiang



K. Lee

Acknowledgements

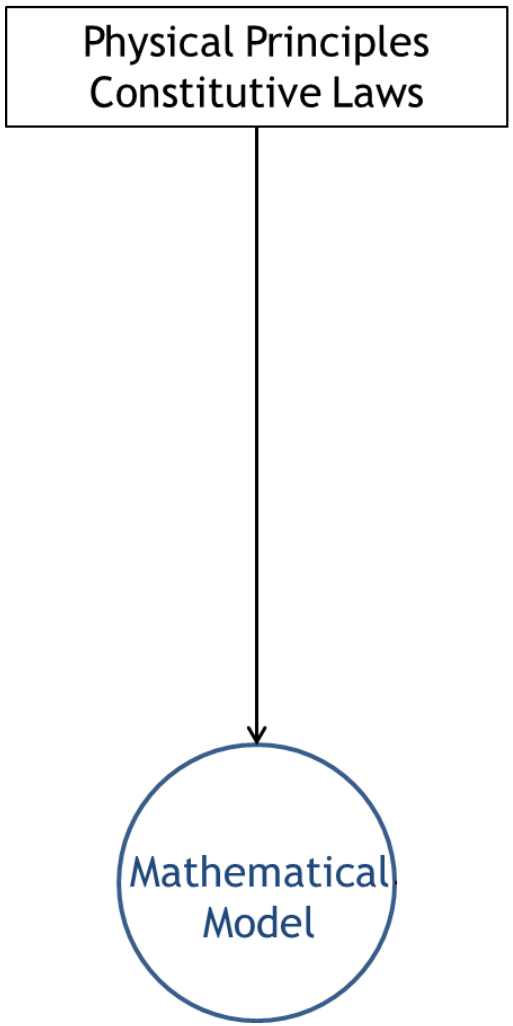
- K. Carlberg, M. Khalil, K. Sargsyan
- C. H. Ozen, J. Jiang, K. Lee
- C. Hoang, Y. Choi
- A. Pinar
- ...

Overview

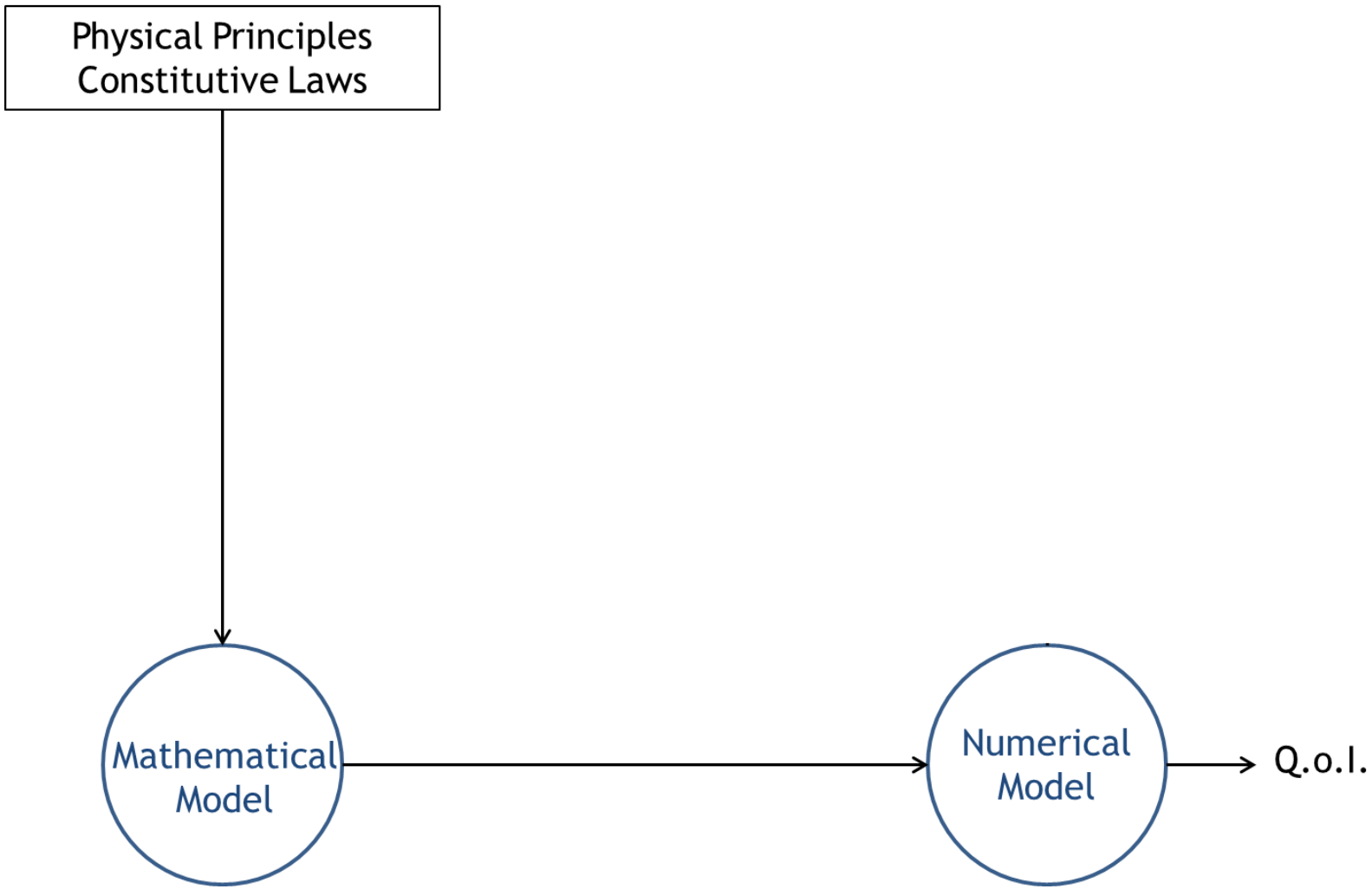
UQ in predictive science

Physical Principles
Constitutive Laws

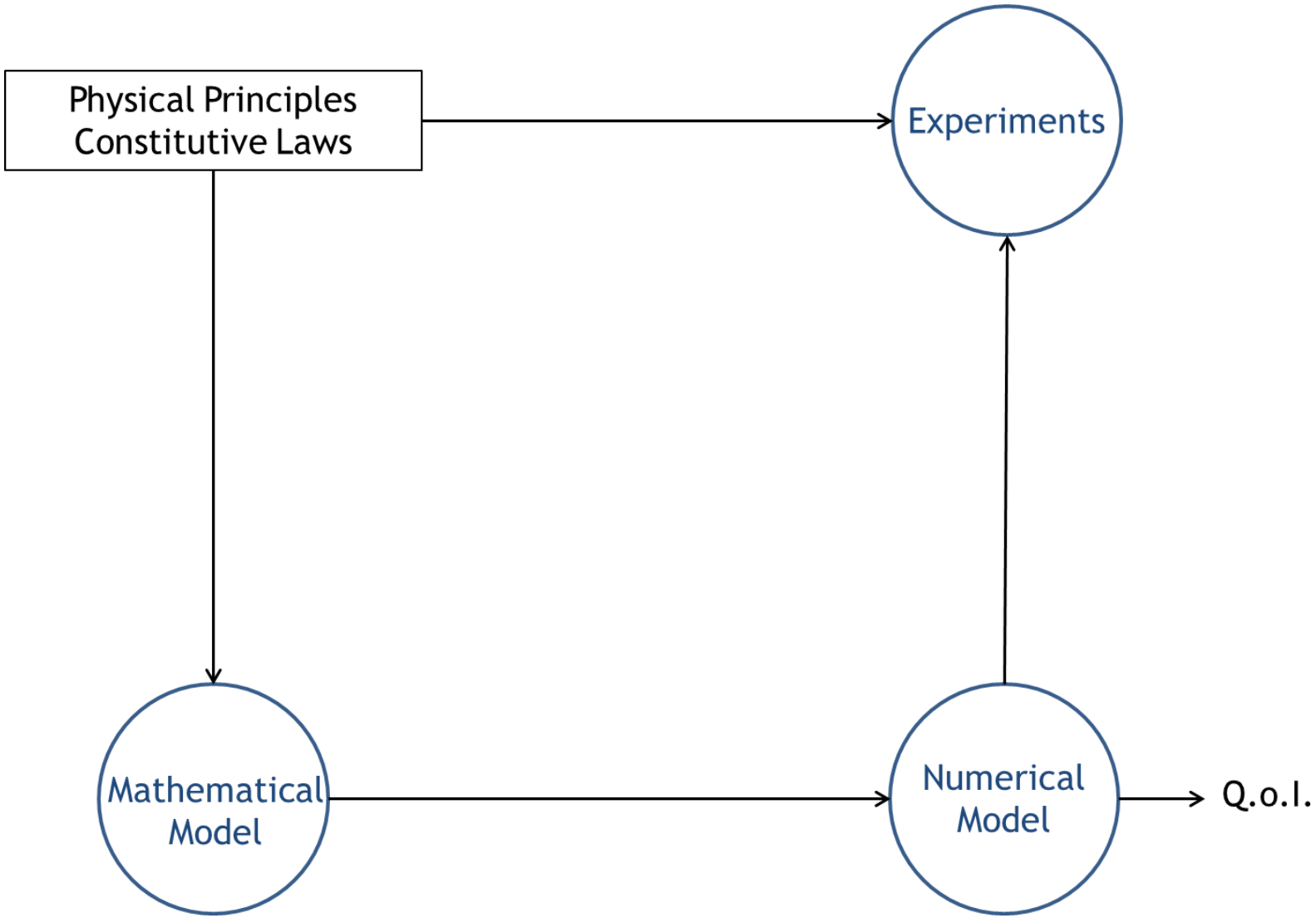
UQ in predictive science



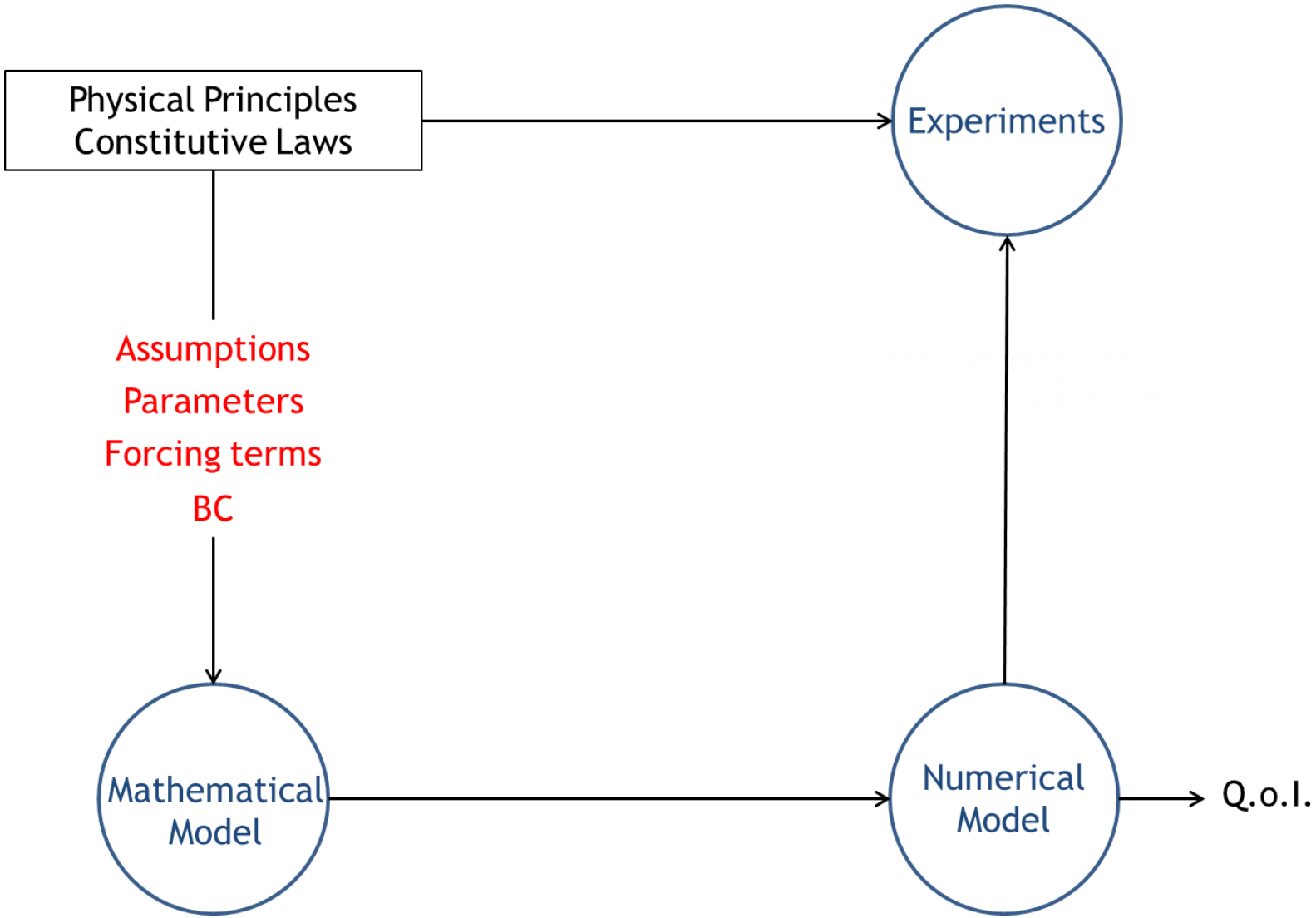
UQ in predictive science



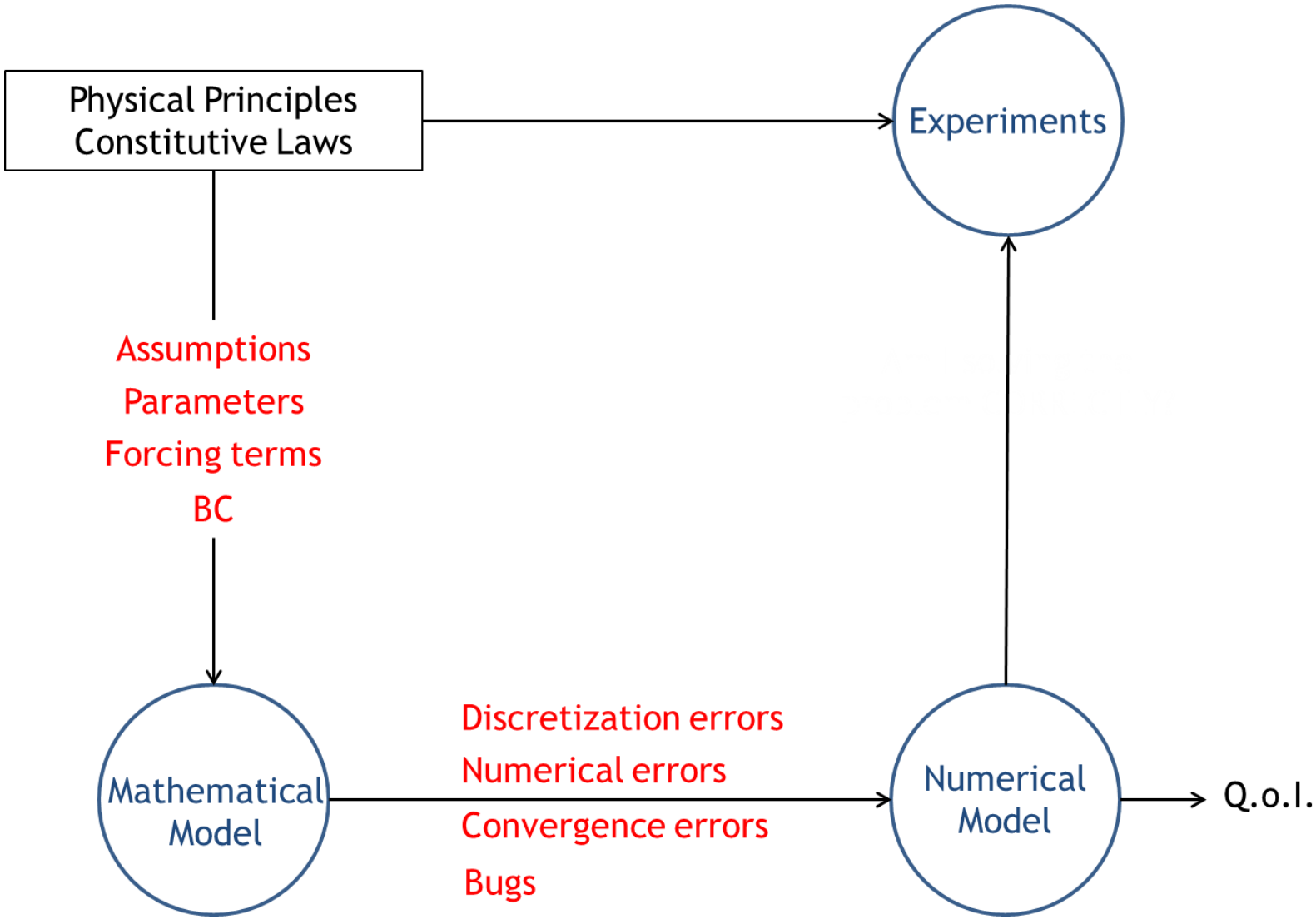
UQ in predictive science



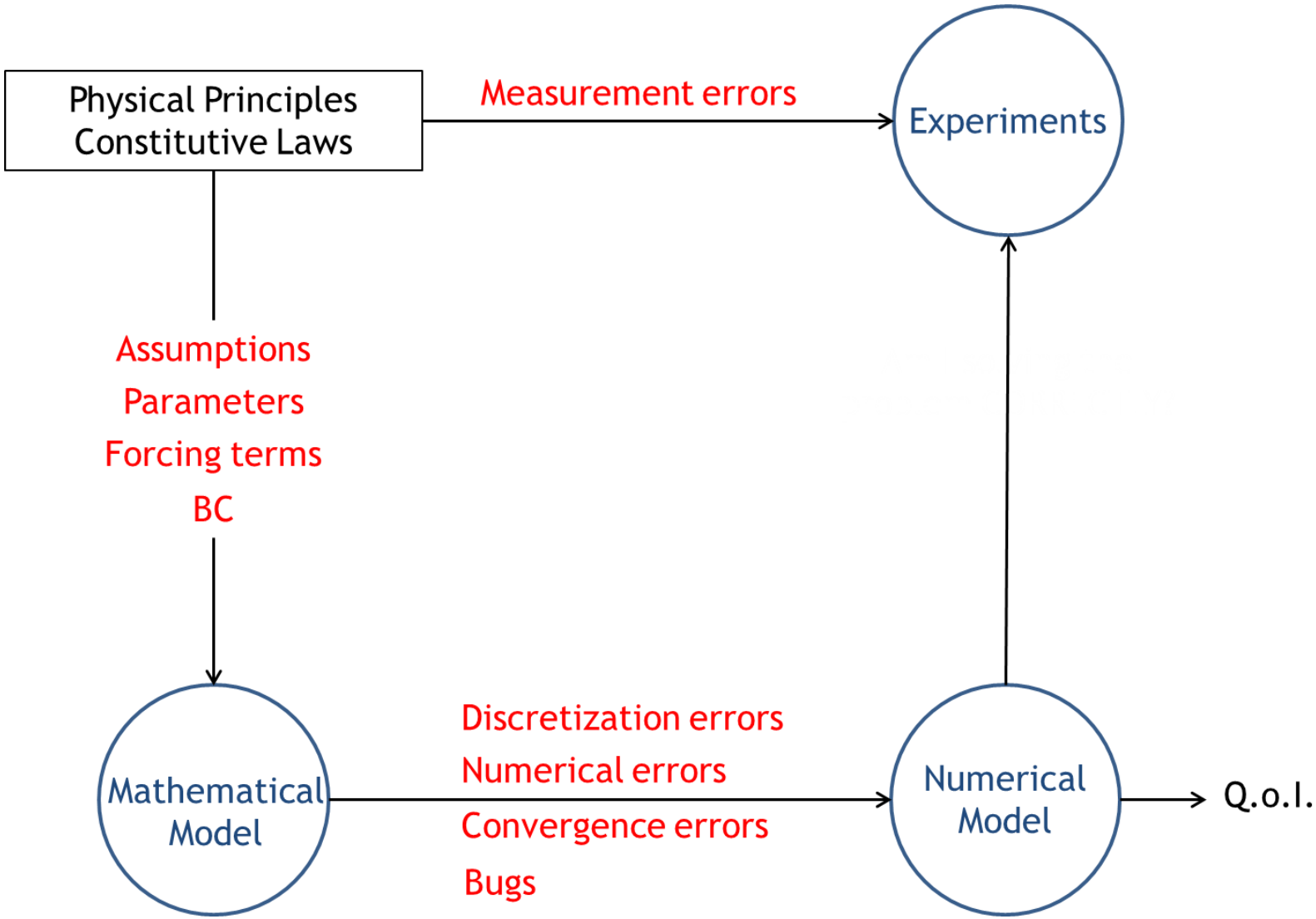
UQ in predictive science



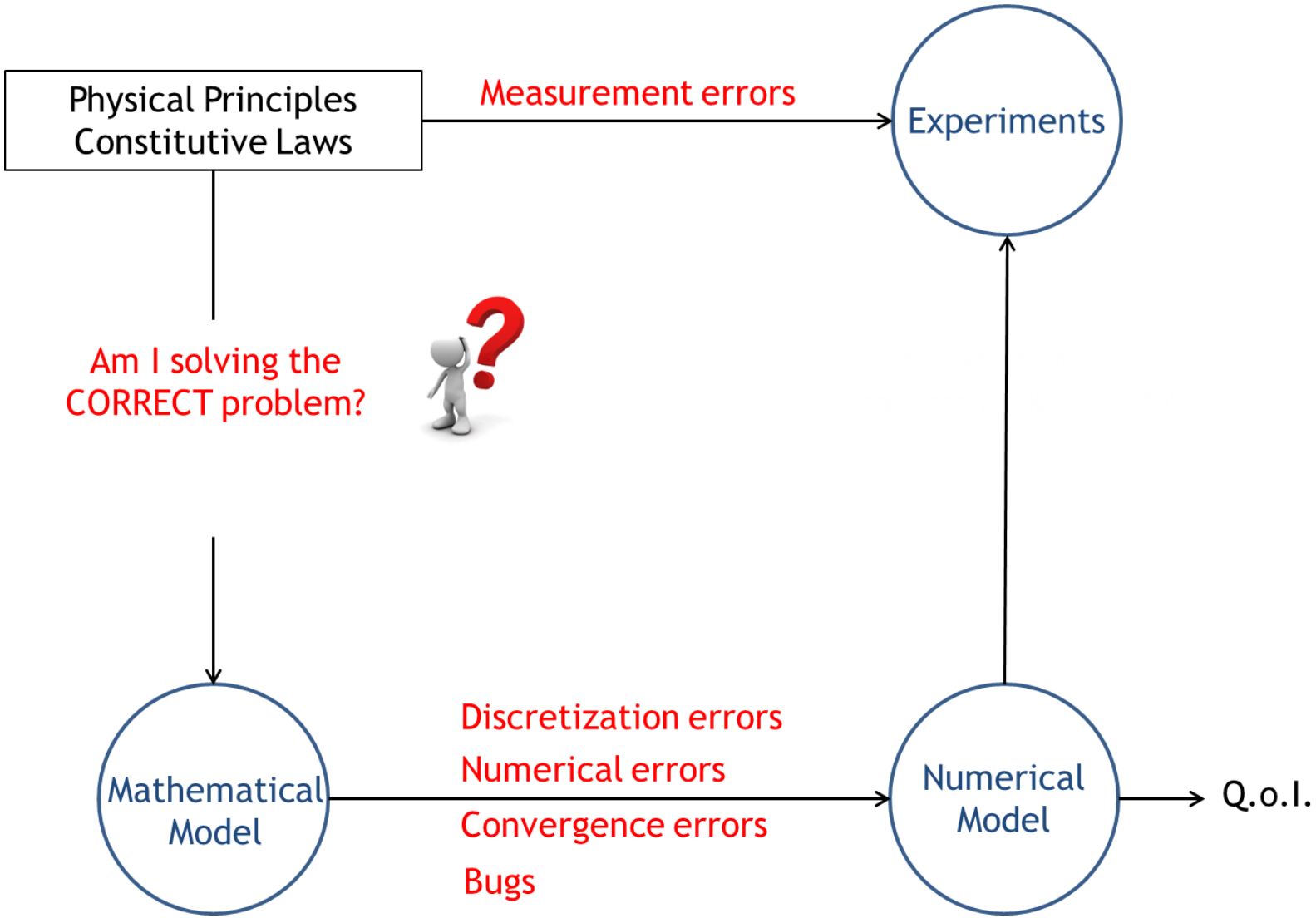
UQ in predictive science



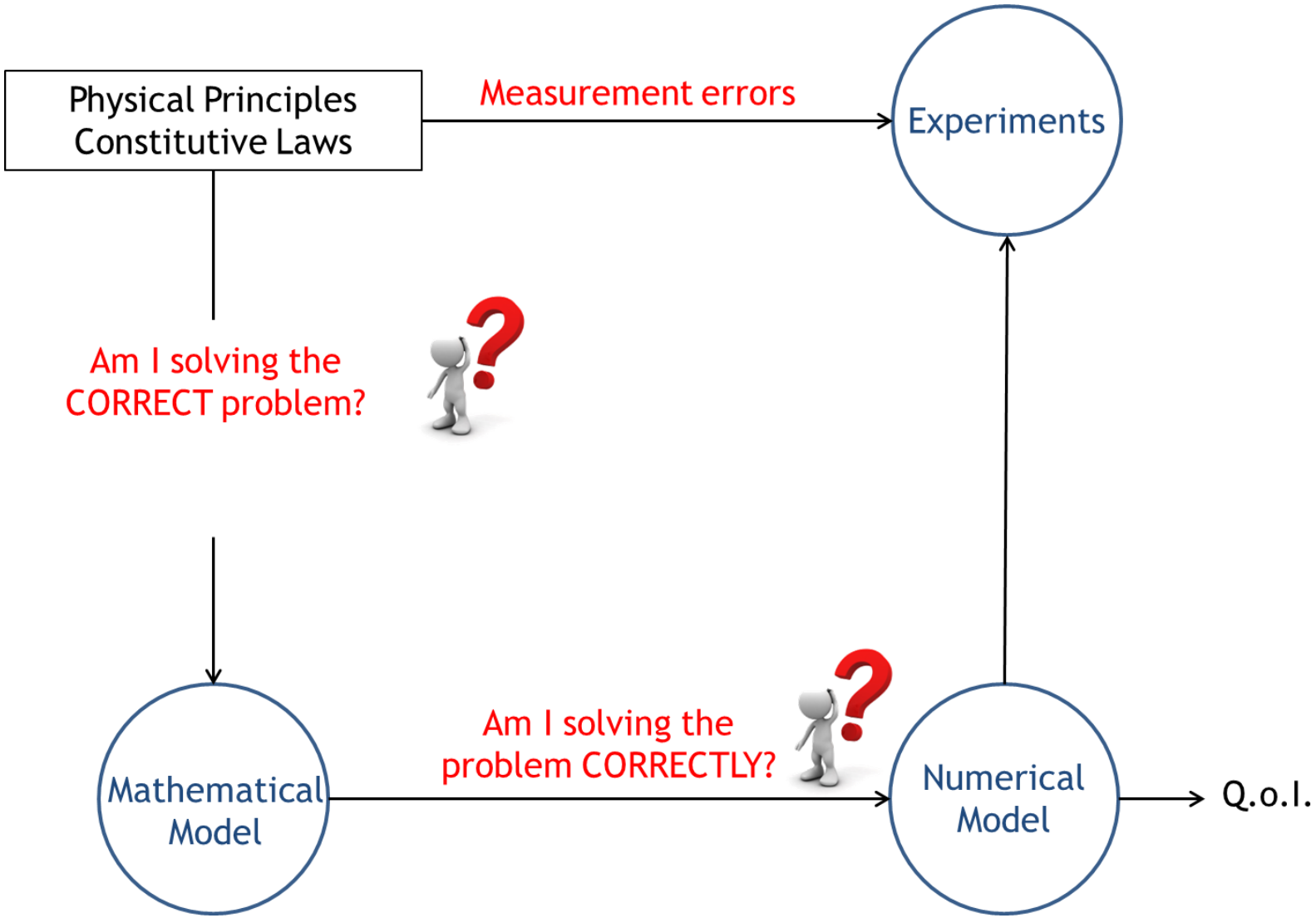
UQ in predictive science



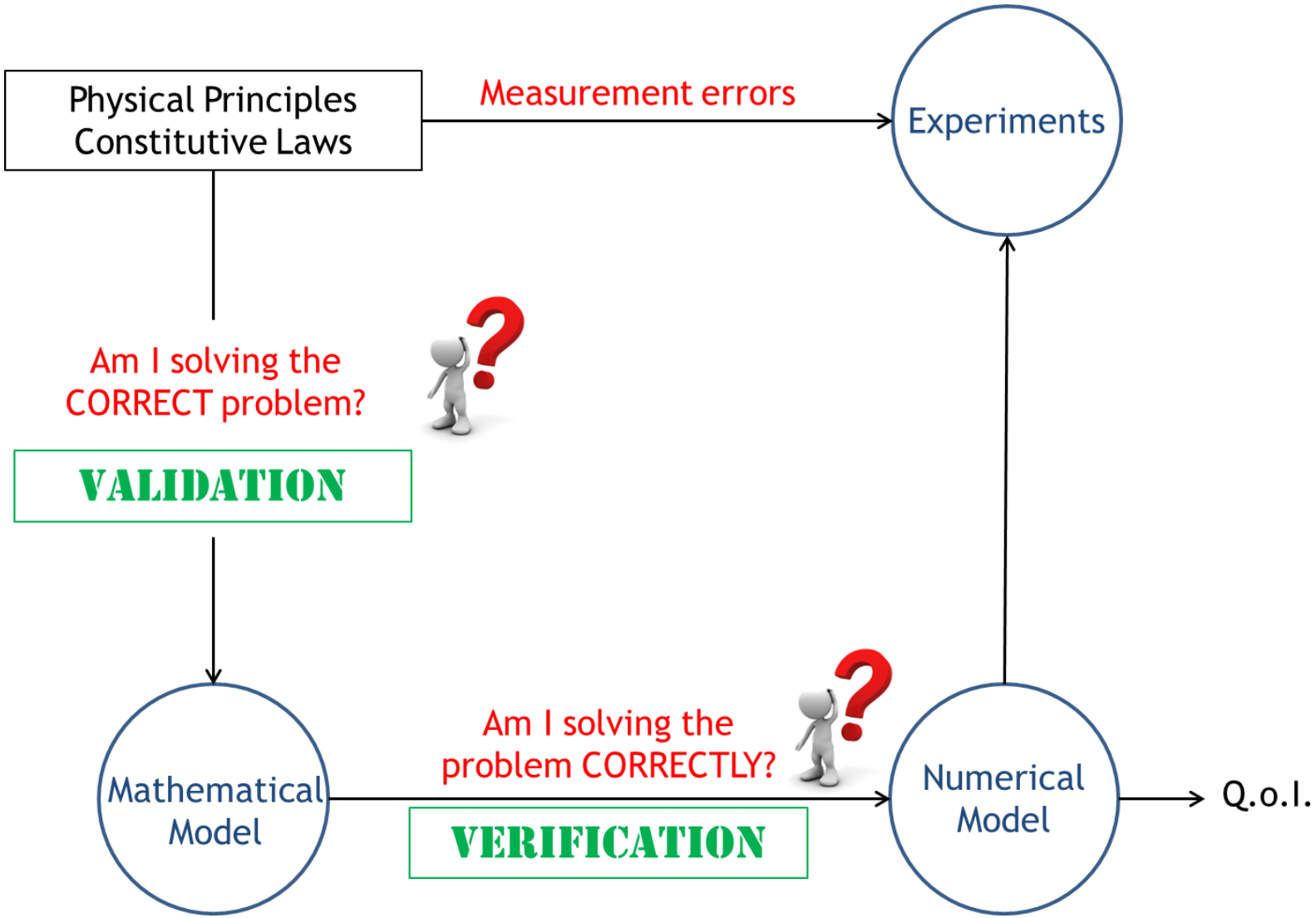
UQ in predictive science



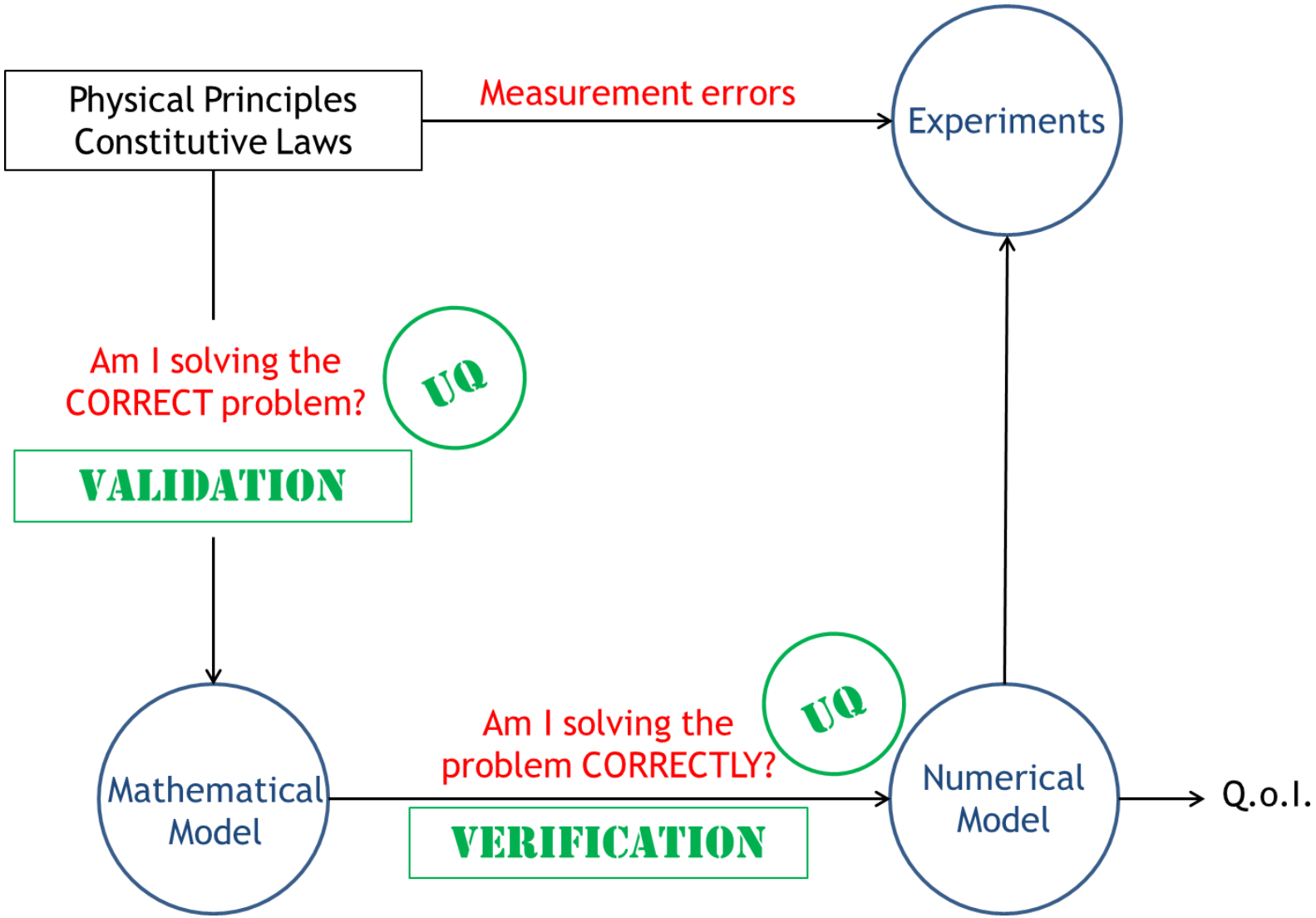
UQ in predictive science



UQ in predictive science



UQ in predictive science



UQ in predictive science

Uncertainty Quantification

"The science of **identifying**, **quantifying**, and **reducing** uncertainties associated with models, numerical algorithms, experiments, and quantities of interest."

[R.C. Smith, "Uncertainty Quantification: Theory, Implementation, and Applications", SIAM, 2013]

UQ in predictive science

Uncertainty Quantification

"The science of **identifying**, **quantifying**, and **reducing** uncertainties associated with models, numerical algorithms, experiments, and quantities of interest."

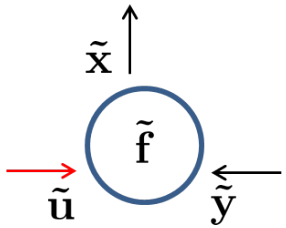
[R.C. Smith, "Uncertainty Quantification: Theory, Implementation, and Applications", SIAM, 2013]

Identification

- **Aleatory** uncertainties - Non-reducible
  Due to **intrinsic** variation or randomness
  E.g.: Manufacturing variability
- **Epistemic** uncertainties - Reducible
  Due to **lack of knowledge**
  E.g.: Coarse meshes, finite time-step sizes, low-fidelity models, BC data

Quantification of uncertainties

Deterministic

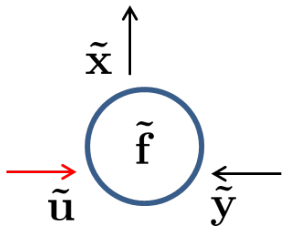


- Inputs: $\tilde{\mathbf{y}} \in \mathbb{R}^{\tilde{y}}$, $\tilde{\mathbf{u}} \in \mathbb{R}^{\tilde{u}}$
- Outputs: $\tilde{\mathbf{x}} \in \mathbb{R}^{\tilde{x}}$
- Forward function: $\tilde{\mathbf{f}} : \mathbb{R}^{\tilde{y}} \times \mathbb{R}^{\tilde{u}} \to \mathbb{R}^{\tilde{x}}$ s.t.

$$\tilde{\mathbf{x}} = \tilde{\mathbf{f}}(\tilde{\mathbf{y}}, \tilde{\mathbf{u}}).$$

Quantification of uncertainties

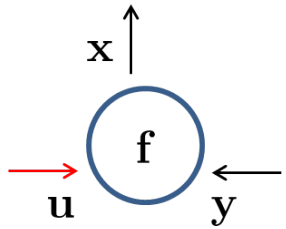
Deterministic



- Inputs: $\tilde{\mathbf{y}} \in \mathbb{R}^{\tilde{y}}$, $\tilde{\mathbf{u}} \in \mathbb{R}^{\tilde{u}}$
- Outputs: $\tilde{\mathbf{x}} \in \mathbb{R}^{\tilde{x}}$
- Forward function: $\tilde{\mathbf{f}} : \mathbb{R}^{\tilde{y}} \times \mathbb{R}^{\tilde{u}} \to \mathbb{R}^{\tilde{x}}$ s.t.

$$\tilde{\mathbf{x}} = \tilde{\mathbf{f}}(\tilde{\mathbf{y}}, \tilde{\mathbf{u}}).$$

Stochastic



- Inputs: random variables described by deterministic parameters $\mathbf{y} \in \mathbb{R}^{y}$, $\mathbf{u} \in \mathbb{R}^{u}$
- Outputs: random variables described by deterministic parameters $\mathbf{x} \in \mathbb{R}^{x}$
- Forward uncertainty propagator: $\mathbf{f} : \mathbb{R}^{y} \times \mathbb{R}^{u} \to \mathbb{R}^{x}$ s.t.

$$\mathbf{x} = \mathbf{f}(\mathbf{y}, \mathbf{u}).$$

Stochastic representations

- **Polynomial Chaos Expansion (PCE, spectral UQ)** [Wiener, 1938]
  Truncated **spectral expansion** in terms of orthogonal functions.

Stochastic representations

- **Polynomial Chaos Expansion (PCE, spectral UQ)** [Wiener, 1938]
  Truncated **spectral expansion** in terms of orthogonal functions.

  Prior work:
  - Elasticity
    - ★ Ghanem, R., Spanos, P.: A spectral stochastic finite element formulation for reliability analysis. J. Eng. Mech. ASCE 117, 2351-2372 (1991)
    - ★ Ghanem, R., Spanos, P.: Stochastic Finite Elements: A Spectral Approach, 2nd edn. Dover, New York (2002)
  - Heat transfer
    - ★ Hien, T., Kleiber, M.: Stochastic finite element modeling in linear transient heat transfer. Comput. Methods Appl. Mech. Eng. 144, 111-124 (1997)
    - ★ Babuska, I., Chatzipantelidis, P.: On solving elliptic stochastic partial differential equations. Comput. Methods Appl. Mech. Eng. 191, 4093-4122 (2002)
  - Fluid flow
    - ★ Ghanem, R.: Probabilistic characterization of transport in heterogeneous media. Comput. Methods Appl. Mech. Eng. 158, 199?220 (1998)
    - ★ Le Maitre, O., Knio, O., Najm, H., Ghanem, R.: A stochastic projection method for fluid flow. I. Basic formulation. J. Comput. Phys. 173, 481-511 (2001)
  - FSI, electro-chemical microchannel flows, ...

Stochastic representations

- **Polynomial Chaos Expansion (PCE, spectral UQ)** [Wiener, 1938]
  Truncated **spectral expansion** in terms of orthogonal functions.

  Let
  - $\tilde{\mathbf{W}}(t, x, \omega)$: random process
  - $\boldsymbol{\xi}(\omega) = (\xi_1(\omega), \ldots, \xi_n(\omega))$: random vector
  - (Ω, σ, π): probability space

Stochastic representations

- **Polynomial Chaos Expansion (PCE, spectral UQ)** [Wiener, 1938]
  Truncated **spectral expansion** in terms of orthogonal functions.

  Let
  - $\tilde{\mathbf{W}}(t, x, \omega)$: random process
  - $\boldsymbol{\xi}(\omega) = (\xi_1(\omega), \ldots, \xi_n(\omega))$: random vector
  - (Ω, σ, π): probability space

  Then,

  PCE

  $$\tilde{\mathbf{W}}_K(t, x, \boldsymbol{\xi}) = \sum_{k=0}^{K} w_k(t, x) \boldsymbol{\Psi}_k(\boldsymbol{\xi})$$

  - $w_k(t, x)$: deterministic coefficients
  - $\{\boldsymbol{\Psi}_k\}$: π-orthogonal basis functions s.t. $(n = 1)$

  $$\mathbb{E}[\psi_i(\xi)\psi_j(\xi)] = \int_{\Omega} \psi_i(\xi)\psi_j(\xi)\pi(\xi)d\xi = \langle \psi_i, \psi_i \rangle_\pi \delta_{ij}$$

Stochastic representations

- **Polynomial Chaos Expansion (PCE, spectral UQ)** [Wiener, 1938]

$$\tilde{\mathbf{W}}_K(t,x,\boldsymbol{\xi}) = \sum_{k=0}^{K} w_k(t,x)\boldsymbol{\Psi}_k(\boldsymbol{\xi})$$

Properties:

- Smoothness
- "Easy" sampling
- "Easy" statistical and sensitivity analysis, e.g. $(n = 1)$

$$\mathbb{E}\left[\tilde{W}_K(t,x,\xi)\right] = w_0(t,x)$$

$$\text{var}\left[\tilde{W}_K(t,x,\xi)\right] = \sum_{k=1}^{K} w_k^2(t,x)\langle\psi_k,\psi_k\rangle_\pi$$

$\ldots$

Stochastic representations

- **Polynomial Chaos Expansion (PCE, spectral UQ)** [Wiener, 1938]

$$\tilde{\mathbf{W}}_K(t, x, \boldsymbol{\xi}) = \sum_{k=0}^{K} w_k(t, x) \boldsymbol{\Psi}_k(\boldsymbol{\xi})$$

Basis functions

Stochastic representations

- **Polynomial Chaos Expansion (PCE, spectral UQ)** [Wiener, 1938]

$$\tilde{\mathbf{W}}_K(t, x, \boldsymbol{\xi}) = \sum_{k=0}^{K} w_k(t, x) \boldsymbol{\Psi}_k(\boldsymbol{\xi})$$

Basis functions

Q: For a fixed number of terms of the PCE, what family of orthogonal polynomials maximizes the convergence rate?

Stochastic representations

- **Polynomial Chaos Expansion (PCE, spectral UQ)** [Wiener, 1938]

$$\tilde{\mathbf{W}}_K(t, x, \boldsymbol{\xi}) = \sum_{k=0}^{K} w_k(t, x) \boldsymbol{\Psi}_k(\boldsymbol{\xi})$$

Basis functions

Q: For a fixed number of terms of the PCE, what family of orthogonal polynomials maximizes the convergence rate?

A: Askey Scheme [Askey, R., Wilson, J., 1985; Xiu, D., Karniadakis, G., 2002]

Distribution	Basis	Support
Gaussian	Hermite	$(-\infty, \infty)$
Uniform	Legendre	$[a, b]$
γ	Laguerre	$[0, \infty)$
β	Jacobi	$[a, b]$

Table: A selection from the Askey scheme.

Stochastic representations

- **Deterministic**
  "Degenerate" case of PCE.

Stochastic representations

- **Deterministic**
  "Degenerate" case of PCE.

- **Kernel Density Estimation (KDE)**
  Non-parametric representation of the **p.d.f.** of a random variable based on samples $\mathbf{s}_i$.

KDE

$$\pi(\mathbf{w}) = \frac{1}{(2\pi)^{K/2} det(\mathbf{B})} \sum_{k=1}^{K} exp\left(-\frac{1}{2}(\mathbf{w}-\mathbf{s}_k)^T \mathbf{B}^{-1}(\mathbf{w}-\mathbf{s}_k)\right)$$

for some bandwidth matrix (usually diagonal) $\mathbf{B}$.

Stochastic representations

- **Moment-generating function (MGF)**
  Encapsulate all the **moments** of a distribution.

Moment-generating function

The moment-generating function of an n-dimensional random vector $\tilde{\mathbf{W}}$ is a function $M_{\tilde{\mathbf{W}}} : \mathbb{R}^n \to [0, \infty]^n$ defined by

$$M_{\tilde{\mathbf{W}}}(\mathbf{t}) = \mathbb{E}\left[e^{\mathbf{t}^T \tilde{\mathbf{W}}}\right].$$

Stochastic representations

- **Moment-generating function (MGF)**
  Encapsulate all the **moments** of a distribution.

Moment-generating function

The moment-generating function of an n-dimensional random vector $\tilde{\mathbf{W}}$ is a function $M_{\tilde{\mathbf{W}}} : \mathbb{R}^n \to [0, \infty]^n$ defined by

$$M_{\tilde{\mathbf{W}}}(\mathbf{t}) = \mathbb{E}\left[e^{\mathbf{t}^T \tilde{\mathbf{W}}}\right].$$

Series expansion ($n = 1$):

$$M_{\tilde{W}}(t) = \mathbb{E}\left[e^{t\tilde{W}}\right] = \mathbb{E}\left[1 + t\tilde{W} + \frac{t^2\tilde{W}^2}{2!} + \frac{t^3\tilde{W}^3}{3!} + \ldots + \frac{t^n\tilde{W}^n}{n!} + \ldots\right]$$

$$= 1 + t\mathbb{E}\left[\tilde{W}\right] + \frac{t^2\mathbb{E}\left[\tilde{W}^2\right]}{2!} + \frac{t^3\mathbb{E}\left[\tilde{W}^3\right]}{3!} + \ldots + \frac{t^n\mathbb{E}\left[\tilde{W}^n\right]}{n!} + \ldots$$

Stochastic representations

- **Moment-generating function (MGF)**
  Encapsulate all the **moments** of a distribution.

Moment-generating function

The moment-generating function of an n-dimensional random vector $\tilde{\mathbf{W}}$ is a function $M_{\tilde{\mathbf{W}}} : \mathbb{R}^n \to [0, \infty]^n$ defined by

$$M_{\tilde{\mathbf{W}}}(\mathbf{t}) = \mathbb{E}\left[e^{\mathbf{t}^T \tilde{\mathbf{W}}}\right].$$

Series expansion ($n = 1$):

$$M_{\tilde{W}}(t) = \mathbb{E}\left[e^{t\tilde{W}}\right] = \mathbb{E}\left[1 + t\tilde{W} + \frac{t^2\tilde{W}^2}{2!} + \frac{t^3\tilde{W}^3}{3!} + \ldots + \frac{t^n\tilde{W}^n}{n!} + \ldots\right]$$

$$= 1 + t\mathbb{E}\left[\tilde{W}\right] + \frac{t^2\mathbb{E}\left[\tilde{W}^2\right]}{2!} + \frac{t^3\mathbb{E}\left[\tilde{W}^3\right]}{3!} + \ldots + \frac{t^n\mathbb{E}\left[\tilde{W}^n\right]}{n!} + \ldots$$

$$\mathbb{E}\left[\tilde{W}^n\right] = \frac{d^n}{dt^n} M_{\tilde{W}}(t)$$

Stochastic representations

- **Moment-generating function (MGF)**

Sigma Points

Set of points that encode information about **mean** and **variance**.

- S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new approach for filtering nonlinear systems. In Proceedings of the American Control Conference, pages 1628-1632, 1995
- S. J. Julier. A Skewed Approach to Filtering. In SPIE Conference on Signal and Data Processing of Small Targets, volume 3373, pages 271?282, Orlando, Florida, April 1998. SPIE
- S. J. Julier and J. K. Uhlmann. A New Extension of the Kalman Filter to Nonlinear Systems. In Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls., 1997

Stochastic representations

- **Moment-generating function**

Sigma Points [S. Julier, J. Uhlmann, 1997]

Set of points that encode information about **mean** and **variance**.

Let

- $\tilde{\mathbf{W}}$: n-dimensional random variable with mean $\overline{\mathbf{w}}$ and covariance $\mathbf{P}$

Stochastic representations

- **Moment-generating function**

Sigma Points [S. Julier, J. Uhlmann, 1997]

Set of points that encode information about **mean** and **variance**.

Let

- $\tilde{\mathbf{W}}$: n-dimensional random variable with mean $\overline{\mathbf{w}}$ and covariance $\mathbf{P}$
- $(\boldsymbol{\mathcal{X}}_i(\kappa), \omega_i)$, $i = 0, \ldots, 2n$, $\kappa \in \mathbb{R}$: sigma points and weights

Stochastic representations

- **Moment-generating function**

Sigma Points [S. Julier, J. Uhlmann, 1997]

Set of points that encode information about **mean** and **variance**.

Let

- $\tilde{\mathbf{W}}$: n-dimensional random variable with mean $\overline{\mathbf{w}}$ and covariance $\mathbf{P}$
- $(\boldsymbol{\mathcal{X}}_i(\kappa), \omega_i)$, $i = 0, \ldots, 2n$, $\kappa \in \mathbb{R}$: sigma points and weights
- $\boldsymbol{\mathcal{W}}_i = \tilde{\mathbf{W}}(\boldsymbol{\mathcal{X}}_i)$, $i = 0, \ldots, 2n$.

Stochastic representations

- **Moment-generating function**

Sigma Points [S. Julier, J. Uhlmann, 1997]

Set of points that encode information about **mean** and **variance**.

Let

- $\tilde{\mathbf{W}}$: n-dimensional random variable with mean $\overline{\mathbf{w}}$ and covariance $\mathbf{P}$
- $(\boldsymbol{\mathcal{X}}_i(\kappa), \omega_i)$, $i = 0, \ldots, 2n$, $\kappa \in \mathbb{R}$: sigma points and weights
- $\boldsymbol{\mathcal{W}}_i = \tilde{\mathbf{W}}(\boldsymbol{\mathcal{X}}_i)$, $i = 0, \ldots, 2n$.

Then

$$\overline{\mathbf{w}} = \sum_{i=0}^{2n} \omega_i \boldsymbol{\mathcal{W}}_i, \quad \mathbf{P} = \sum_{i=0}^{2n} \omega_i \left[\boldsymbol{\mathcal{W}}_i - \overline{\mathbf{w}}\right] \left[\boldsymbol{\mathcal{W}}_i - \overline{\mathbf{w}}\right]^T.$$

Stochastic representations

- **Moment-generating function**

Sigma Points [S. Julier, J. Uhlmann, 1997]

Set of points that encode information about **mean** and **variance**.

Let

- $\tilde{\mathbf{W}}$: n-dimensional random variable with mean $\overline{\mathbf{w}}$ and covariance $\mathbf{P}$
- $(\boldsymbol{\mathcal{X}}_i(\kappa), \omega_i)$, $i = 0, \ldots, 2n$, $\kappa \in \mathbb{R}$: sigma points and weights
- $\boldsymbol{\mathcal{W}}_i = \tilde{\mathbf{W}}(\boldsymbol{\mathcal{X}}_i)$, $i = 0, \ldots, 2n$.

Then

$$\overline{\mathbf{w}} = \sum_{i=0}^{2n} \omega_i \boldsymbol{\mathcal{W}}_i, \quad \mathbf{P} = \sum_{i=0}^{2n} \omega_i \left[\boldsymbol{\mathcal{W}}_i - \overline{\mathbf{w}}\right] \left[\boldsymbol{\mathcal{W}}_i - \overline{\mathbf{w}}\right]^T.$$

Properties

- κ can be chosen to reduce the error of the higher order moments (e.g., $\kappa = 3 - n$ is **optimal** for Normal distributions)
- Mean and covariance are correct to the **second order**.
  Higher order moments can be calculated, e.g., with *maximum entropy probability distributions.*

Uncertainty propagators

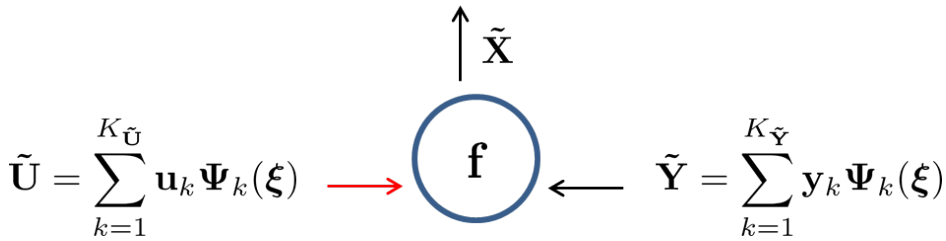
- Stochastic representation
  - PCE/KDE/MGF
  - ...

Uncertainty propagators

- Stochastic representation
  - PCE/KDE/MGF
  - ...
- Handling of deterministic solver
  - **Non-intrusive** ("Solve-then-project")
    Run the deterministic solver several times.

Uncertainty propagators

- Stochastic representation
  - PCE/KDE/MGF
  - ...



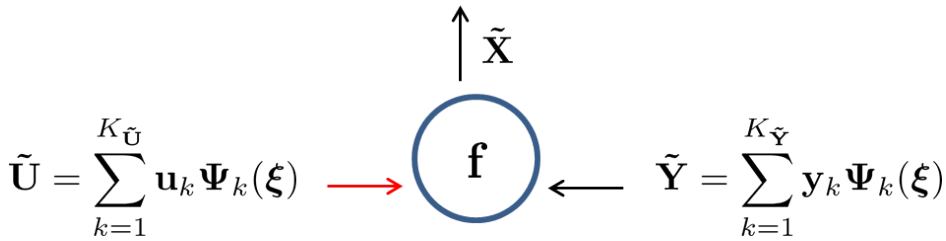
- Handling of deterministic solver
  - **Non-intrusive** ("Solve-then-project")
    Run the deterministic solver several times.

    PC expansion of the output: $\tilde{\mathbf{X}} = \sum_{k=1}^{K_{\tilde{\mathbf{X}}}} \mathbf{x}_k \Psi_k(\boldsymbol{\xi})$, with

$$\mathbf{x}_k = \frac{1}{\|\Psi_k(\boldsymbol{\xi})\|} \int_{\Omega} \tilde{\mathbf{f}} \left(\sum_{k=1}^{K_{\tilde{\mathbf{Y}}}} \mathbf{y}_k \Psi_k(\boldsymbol{\xi}), \sum_{k=1}^{K_{\tilde{\mathbf{U}}}} \mathbf{u}_k \Psi_k(\boldsymbol{\xi}) \right) \Psi_k(\boldsymbol{\xi}) d\boldsymbol{\xi}$$

Uncertainty propagators

- Stochastic representation
  - PCE/KDE/MGF
  - ...



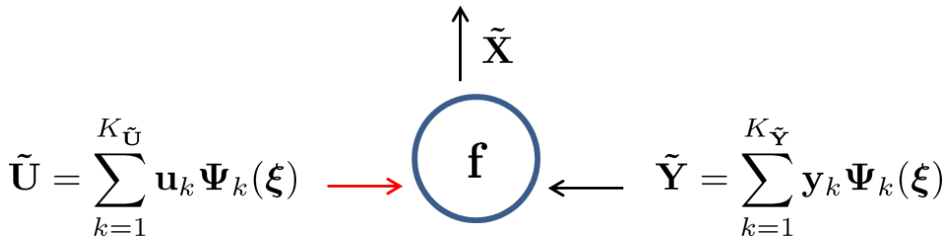
- Handling of deterministic solver
  - **Non-intrusive** ("Solve-then-project")
    Run the deterministic solver several times.
    PC expansion of the output: $\tilde{\mathbf{X}} = \sum_{k=1}^{K_{\tilde{\mathbf{X}}}} \mathbf{x}_k \Psi_k(\boldsymbol{\xi})$, with

$$\mathbf{x}_k = \frac{1}{\|\Psi_k(\boldsymbol{\xi})\|} \int_{\Omega} \tilde{\mathbf{f}} \left(\sum_{k=1}^{K_{\tilde{\mathbf{Y}}}} \mathbf{y}_k \Psi_k(\boldsymbol{\xi}), \sum_{k=1}^{K_{\tilde{\mathbf{U}}}} \mathbf{u}_k \Psi_k(\boldsymbol{\xi}) \right) \Psi_k(\boldsymbol{\xi}) d\boldsymbol{\xi}$$

$$\approx \frac{1}{\|\Psi_k(\boldsymbol{\xi})\|} \sum_{i=1}^{Q} \omega_i \tilde{\mathbf{f}} \left(\sum_{k=1}^{K_{\tilde{\mathbf{Y}}}} \mathbf{y}_k \Psi_k(\boldsymbol{\xi}_i), \sum_{k=1}^{K_{\tilde{\mathbf{U}}}} \mathbf{u}_k \Psi_k(\boldsymbol{\xi}_i) \right) \Psi_k(\boldsymbol{\xi}_i).$$

Uncertainty propagators

- Stochastic representation
  - PCE/KDE/MGF
  - ...



- Handling of deterministic solver
  - **Non-intrusive** ("Solve-then-project")
    Run the deterministic solver several times.
    PC expansion of the output: $\tilde{\mathbf{X}} = \sum_{k=1}^{K_{\tilde{\mathbf{X}}}} \mathbf{x}_k \Psi_k(\boldsymbol{\xi})$, with

$$\mathbf{x}_k = \frac{1}{\|\Psi_k(\boldsymbol{\xi})\|} \int_{\Omega} \tilde{\mathbf{f}} \left(\sum_{k=1}^{K_{\tilde{\mathbf{Y}}}} \mathbf{y}_k \Psi_k(\boldsymbol{\xi}), \sum_{k=1}^{K_{\tilde{\mathbf{U}}}} \mathbf{u}_k \Psi_k(\boldsymbol{\xi}) \right) \Psi_k(\boldsymbol{\xi}) d\boldsymbol{\xi}$$

$$\approx \frac{1}{\|\Psi_k(\boldsymbol{\xi})\|} \sum_{i=1}^{Q} \omega_i \tilde{\mathbf{f}} \left(\sum_{k=1}^{K_{\tilde{\mathbf{Y}}}} \mathbf{y}_k \Psi_k(\boldsymbol{\xi}_i), \sum_{k=1}^{K_{\tilde{\mathbf{U}}}} \mathbf{u}_k \Psi_k(\boldsymbol{\xi}_i) \right) \Psi_k(\boldsymbol{\xi}_i).$$

  - **Intrusive** ("Project-then-solve")
    Require a modification of the solver.

Network UQ

Challenges:

- Long full-system simulation times
- Integration of heterogeneous subsystems

Network UQ

Challenges:

- Long full-system simulation times
- Integration of heterogeneous subsystems

GOAL: Enable **rapid** and **scalable** simulation of uncertainty propagation in **decomposable** systems.

Network UQ

Challenges:

- Long full-system simulation times
- Integration of heterogeneous subsystems

GOAL: Enable **rapid** and **scalable** simulation of uncertainty propagation in **decomposable** systems.

IDEA: Use **bottom-up** approach to

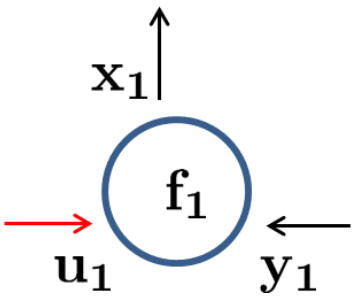
- Perform UQ and MR analysis at the subsystem level
- Propagate the information via overlapping DD techniques

Pros/Cons:

+ No full-system simulations
+ *Global* uncertainties characterized in terms of *local* uncertainties
+ Modularity in network design
- Possible propagation of local truncation errors

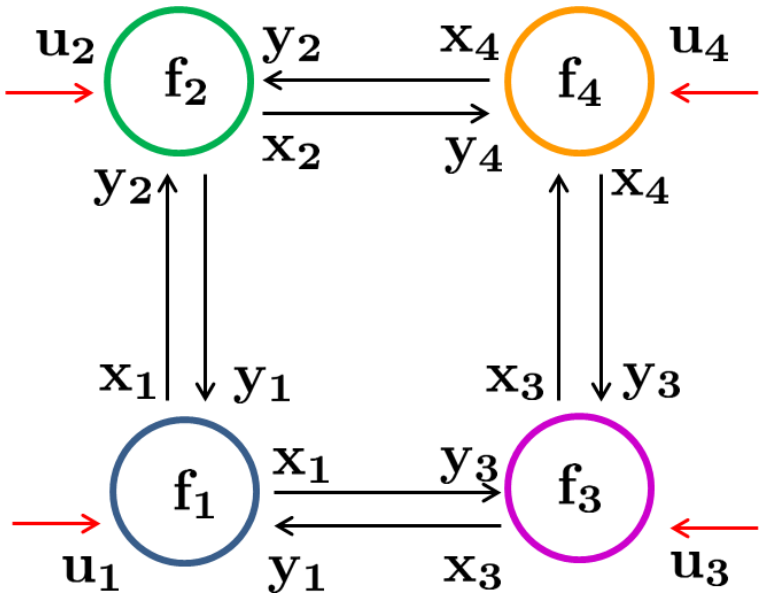
Formulation

Network setup



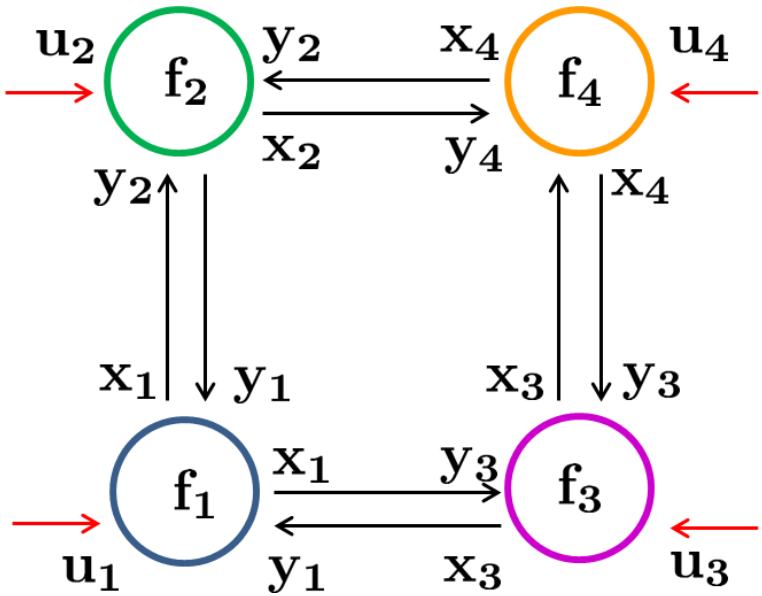
Formulation

Network setup



Formulation

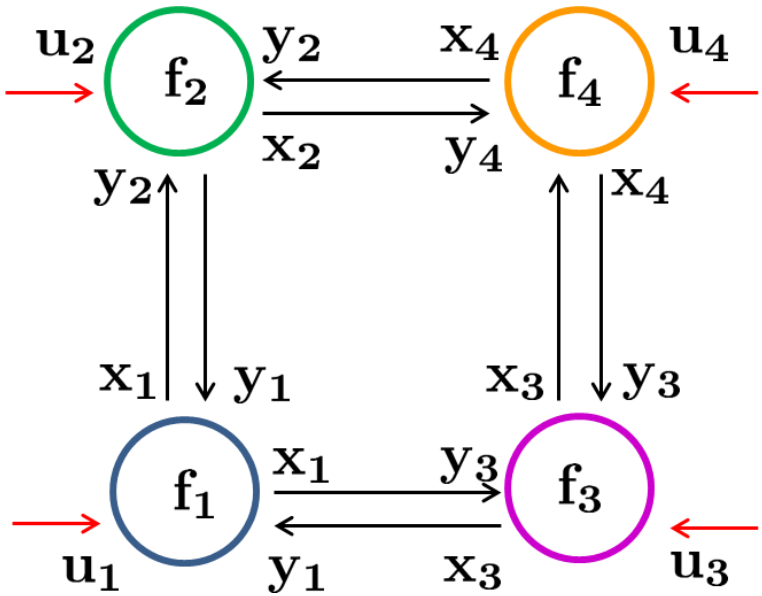
Network setup



- Exogenous inputs
  $\mathbf{u} = [\mathbf{u}_1, \ldots, \mathbf{u}_N]^T$
- Endogenous inputs
  $\mathbf{y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^T$
- Outputs
  $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$
- I/O Selection operator $\mathbf{I}_{\mathbf{x}}^{\mathbf{y}}$

Formulation

Network setup



- Exogenous inputs
  $\mathbf{u} = [\mathbf{u}_1, \ldots, \mathbf{u}_N]^T$
- Endogenous inputs
  $\mathbf{y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^T$
- Outputs
  $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$
- I/O Selection operator $\mathbf{I}_{\mathbf{x}}^{\mathbf{y}}$

Problem formulation

$$\mathbf{x} = \mathbf{f}(\mathbf{I}_{\mathbf{x}}^{\mathbf{y}}\mathbf{x}, \mathbf{u})$$

Network solver

$$\mathbf{x} = \mathbf{f}(\mathbf{I}_{\mathbf{x}}^{\mathbf{y}}\mathbf{x}, \mathbf{u}) \quad \Longleftrightarrow \quad \mathbf{r}(\mathbf{x}, \mathbf{u}) = \mathbf{x} - \mathbf{f}(\mathbf{I}_{\mathbf{x}}^{\mathbf{y}}\mathbf{x}, \mathbf{u}) = \mathbf{0}$$

Network solver

$$\mathbf{x} = \mathbf{f}(\mathbf{l}^{\mathbf{y}}_{\mathbf{x}}\mathbf{x}, \mathbf{u}) \quad \Longleftrightarrow \quad \mathbf{r}(\mathbf{x}, \mathbf{u}) = \mathbf{x} - \mathbf{f}(\mathbf{l}^{\mathbf{y}}_{\mathbf{x}}\mathbf{x}, \mathbf{u}) = \mathbf{0}$$

- **Newton**'s method
  Need $\partial \mathbf{r}/\partial \mathbf{u}$: **Sensitivity analysis**

Network solver

$$\mathbf{x} = \mathbf{f}(\mathbf{l}^{\mathbf{y}}_{\mathbf{x}}\mathbf{x}, \mathbf{u}) \quad \Longleftrightarrow \quad \mathbf{r}(\mathbf{x}, \mathbf{u}) = \mathbf{x} - \mathbf{f}(\mathbf{l}^{\mathbf{y}}_{\mathbf{x}}\mathbf{x}, \mathbf{u}) = \mathbf{0}$$

- **Newton**'s method
  Need $\partial\mathbf{r}/\partial\mathbf{u}$: **Sensitivity analysis**
- **Relaxation** methods (Jacobi, Gauss-Seidel, SOR)
  Equivalent to non-overlapping DD for UQ

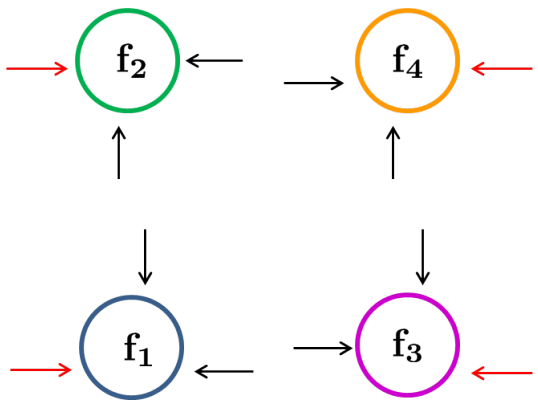
Network solver

$$\mathbf{x} = \mathbf{f}(\mathbf{l}_{\mathbf{x}}^{\mathbf{y}}\mathbf{x}, \mathbf{u}) \quad \Longleftrightarrow \quad \mathbf{r}(\mathbf{x}, \mathbf{u}) = \mathbf{x} - \mathbf{f}(\mathbf{l}_{\mathbf{x}}^{\mathbf{y}}\mathbf{x}, \mathbf{u}) = \mathbf{0}$$

- **Newton**'s method
  Need $\partial \mathbf{r}/\partial \mathbf{u}$: **Sensitivity analysis**
- **Relaxation** methods (Jacobi, Gauss-Seidel, SOR)
  Equivalent to non-overlapping DD for UQ

Jacobi (add. Schwarz)

for i = 1,...,n **do**
$\quad \overline{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{y}_i^k, \mathbf{u}_i)$
end for
$\mathbf{x}^{k+1} = \omega\overline{\mathbf{x}} + (1-\omega)\mathbf{x}^k$
$\mathbf{y}^{k+1} = \mathbf{l}_{\mathbf{x}}^{\mathbf{y}}\mathbf{x}$



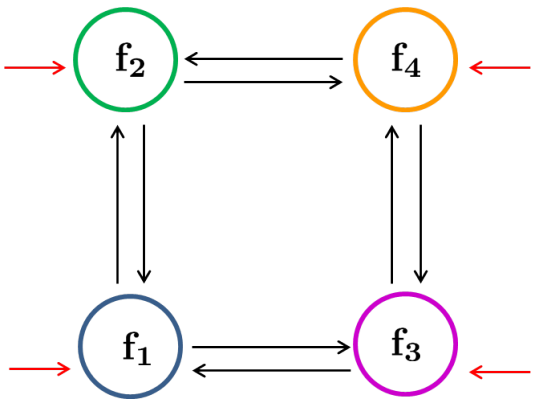
Network solver

$$\mathbf{x} = \mathbf{f}(\mathbf{I}_{\mathbf{x}}^{\mathbf{y}}\mathbf{x}, \mathbf{u}) \quad \Longleftrightarrow \quad \mathbf{r}(\mathbf{x}, \mathbf{u}) = \mathbf{x} - \mathbf{f}(\mathbf{I}_{\mathbf{x}}^{\mathbf{y}}\mathbf{x}, \mathbf{u}) = \mathbf{0}$$

- **Newton**'s method
  Need $\partial \mathbf{r}/\partial \mathbf{u}$: **Sensitivity analysis**
- **Relaxation** methods (Jacobi, Gauss-Seidel, SOR)
  Equivalent to non-overlapping DD for UQ

Jacobi (add. Schwarz)

for i = 1,...,n **do**
$\quad \overline{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{y}_i^k, \mathbf{u}_i)$
end for
$\mathbf{x}^{k+1} = \omega\overline{\mathbf{x}} + (1-\omega)\mathbf{x}^k$
$\mathbf{y}^{k+1} = \mathbf{I}_{\mathbf{x}}^{\mathbf{y}}\mathbf{x}$



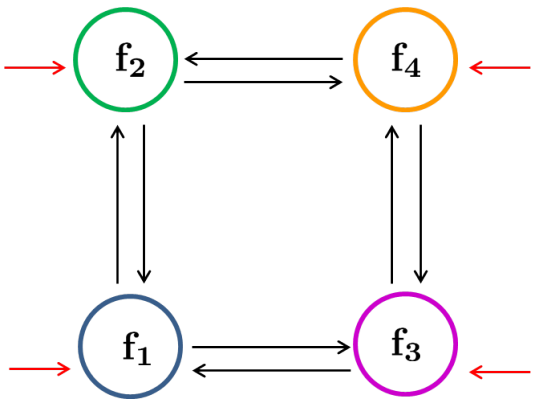
Network solver

$$\mathbf{x} = \mathbf{f}(\mathbf{l}_{\mathbf{x}}^{\mathbf{y}}\mathbf{x}, \mathbf{u}) \quad \Longleftrightarrow \quad \mathbf{r}(\mathbf{x}, \mathbf{u}) = \mathbf{x} - \mathbf{f}(\mathbf{l}_{\mathbf{x}}^{\mathbf{y}}\mathbf{x}, \mathbf{u}) = \mathbf{0}$$

- **Newton**'s method
  Need $\partial \mathbf{r}/\partial \mathbf{u}$: **Sensitivity analysis**
- **Relaxation** methods (Jacobi, Gauss-Seidel, SOR)
  Equivalent to non-overlapping DD for UQ

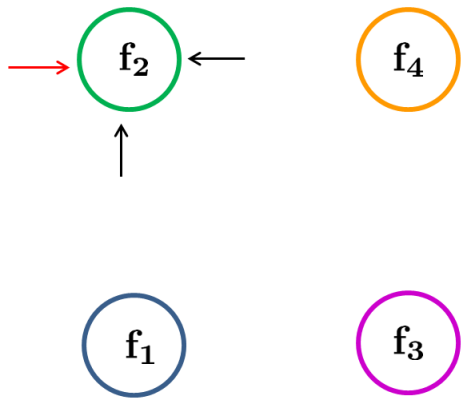
Jacobi (add. Schwarz)

for i = 1,...,n **do**
  $\overline{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{y}_i^k, \mathbf{u}_i)$
end for
  $\mathbf{x}^{k+1} = \omega\overline{\mathbf{x}} + (1-\omega)\mathbf{x}^k$
  $\mathbf{y}^{k+1} = \mathbf{l}_{\mathbf{x}}^{\mathbf{y}}\mathbf{x}$



Gauss-Seidel (mult. Schwarz)

for i = 1,...,n **do**
  $\overline{\mathbf{x}}_{p_i} = \mathbf{f}_{p_i}(\mathbf{y}_{p_i}^{k+1}, \mathbf{u}_{p_i})$
  Update neighbors' inputs $\mathbf{y}^{k+1}$
end for
  $\mathbf{x}^{k+1} = \omega\overline{\mathbf{x}} + (1-\omega)\mathbf{x}^k$



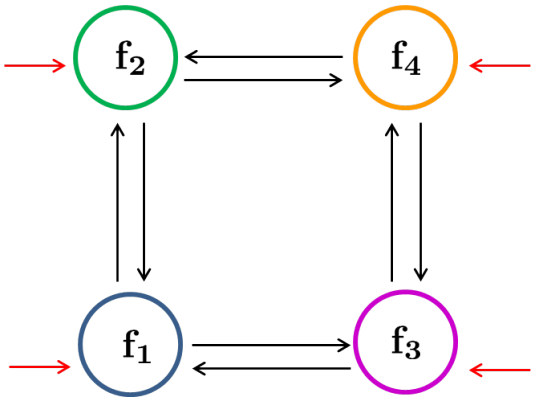
Network solver

$$\mathbf{x} = \mathbf{f}(\mathbf{l}_{\mathbf{x}}^{\mathbf{y}}\mathbf{x}, \mathbf{u}) \quad \Longleftrightarrow \quad \mathbf{r}(\mathbf{x}, \mathbf{u}) = \mathbf{x} - \mathbf{f}(\mathbf{l}_{\mathbf{x}}^{\mathbf{y}}\mathbf{x}, \mathbf{u}) = \mathbf{0}$$

- **Newton**'s method
  Need $\partial \mathbf{r}/\partial \mathbf{u}$: **Sensitivity analysis**
- **Relaxation** methods (Jacobi, Gauss-Seidel, SOR)
  Equivalent to non-overlapping DD for UQ

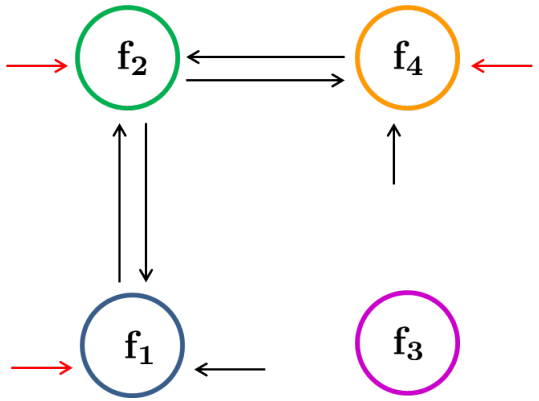
Jacobi (add. Schwarz)

for i = 1,...,n **do**
$\overline{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{y}_i^k, \mathbf{u}_i)$
end for
$\mathbf{x}^{k+1} = \omega\overline{\mathbf{x}} + (1-\omega)\mathbf{x}^k$
$\mathbf{y}^{k+1} = \mathbf{l}_{\mathbf{x}}^{\mathbf{y}}\mathbf{x}$



Gauss-Seidel (mult. Schwarz)

for i = 1,...,n **do**
$\overline{\mathbf{x}}_{p_i} = \mathbf{f}_{p_i}(\mathbf{y}_{p_i}^{k+1}, \mathbf{u}_{p_i})$
Update neighbors' inputs $\mathbf{y}^{k+1}$
end for
$\mathbf{x}^{k+1} = \omega\overline{\mathbf{x}} + (1-\omega)\mathbf{x}^k$



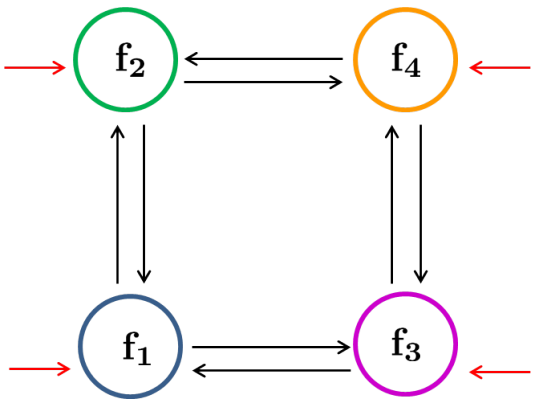
Network solver

$$\mathbf{x} = \mathbf{f}(\mathbf{l}_{\mathbf{x}}^{\mathbf{y}}\mathbf{x}, \mathbf{u}) \quad \Longleftrightarrow \quad \mathbf{r}(\mathbf{x}, \mathbf{u}) = \mathbf{x} - \mathbf{f}(\mathbf{l}_{\mathbf{x}}^{\mathbf{y}}\mathbf{x}, \mathbf{u}) = \mathbf{0}$$

- **Newton**'s method
  Need $\partial \mathbf{r}/\partial \mathbf{u}$: **Sensitivity analysis**
- **Relaxation** methods (Jacobi, Gauss-Seidel, SOR)
  Equivalent to non-overlapping DD for UQ

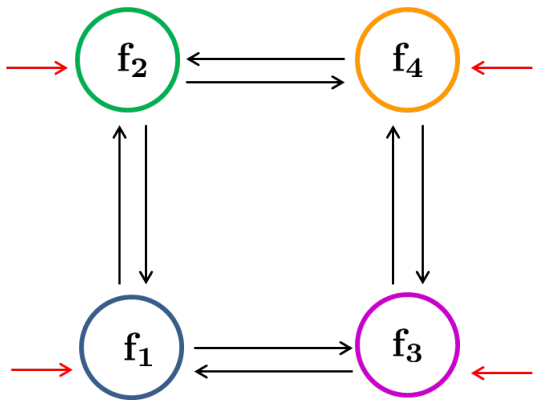
Jacobi (add. Schwarz)

for i = 1,...,n **do**
  $\overline{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{y}_i^k, \mathbf{u}_i)$
end for
  $\mathbf{x}^{k+1} = \omega\overline{\mathbf{x}} + (1-\omega)\mathbf{x}^k$
  $\mathbf{y}^{k+1} = \mathbf{l}_{\mathbf{x}}^{\mathbf{y}}\mathbf{x}$



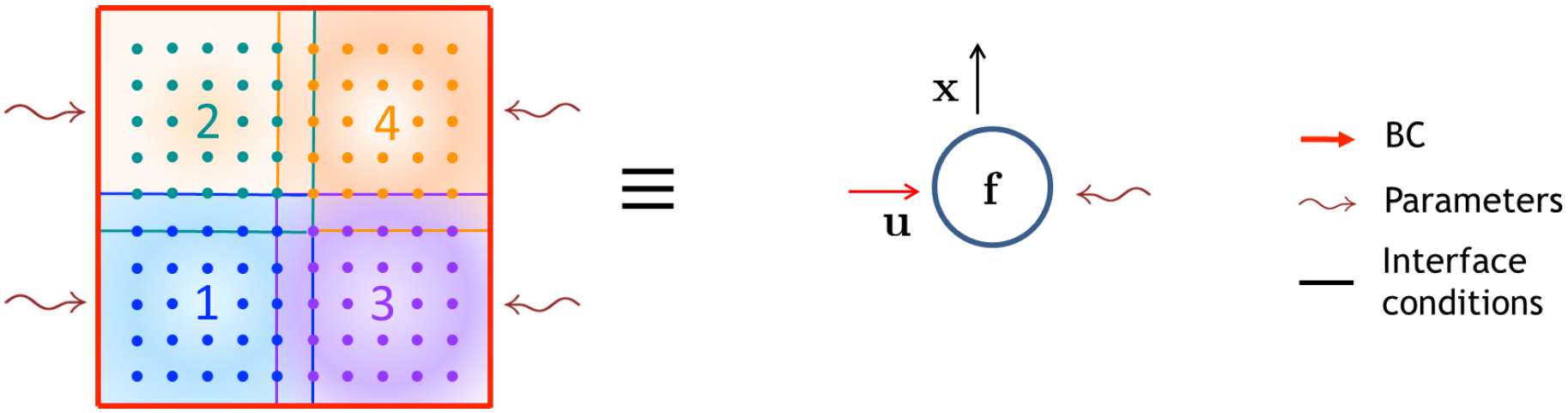
Gauss-Seidel (mult. Schwarz)

for i = 1,...,n **do**
  $\overline{\mathbf{x}}_{p_i} = \mathbf{f}_{p_i}(\mathbf{y}_{p_i}^{k+1}, \mathbf{u}_{p_i})$
  Update neighbors' inputs $\mathbf{y}^{k+1}$
end for
  $\mathbf{x}^{k+1} = \omega\overline{\mathbf{x}} + (1-\omega)\mathbf{x}^k$



UQ for Domain Decomposition

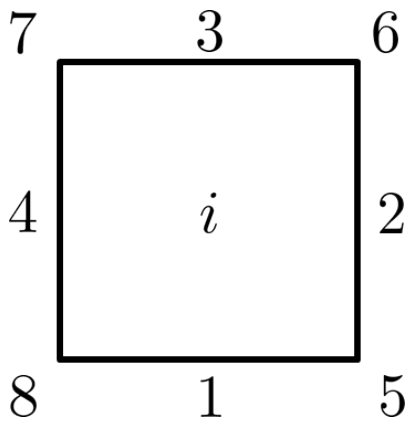
Top-Down approach



UQ for Domain Decomposition

Bottom-Up approach

- **Offline** phase

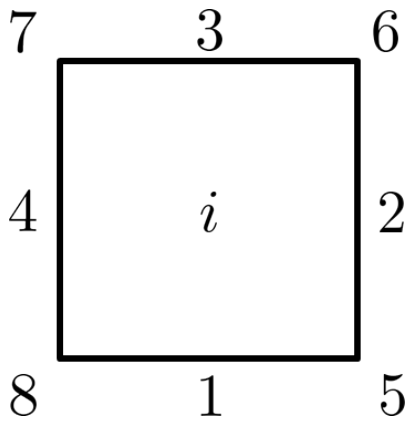


- Design parameters (length/width/...)
- I/O ports (endogenous/exogenous)
- Deterministic and stochastic info (det. solver/stochastic parametrization/...)
- Topology

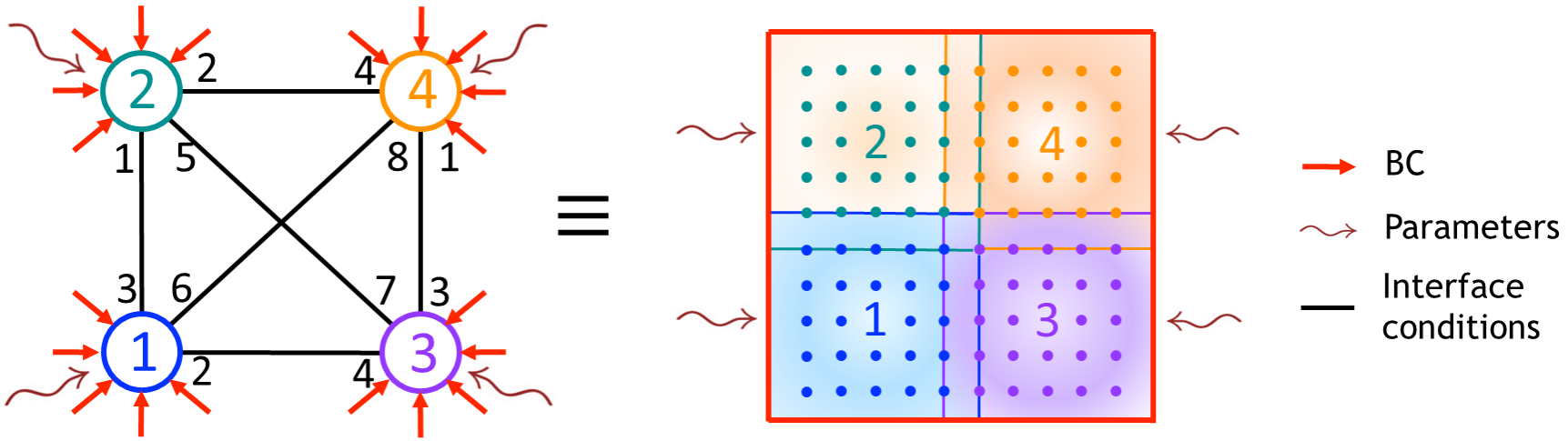
UQ for Domain Decomposition

Bottom-Up approach

- **Offline** phase

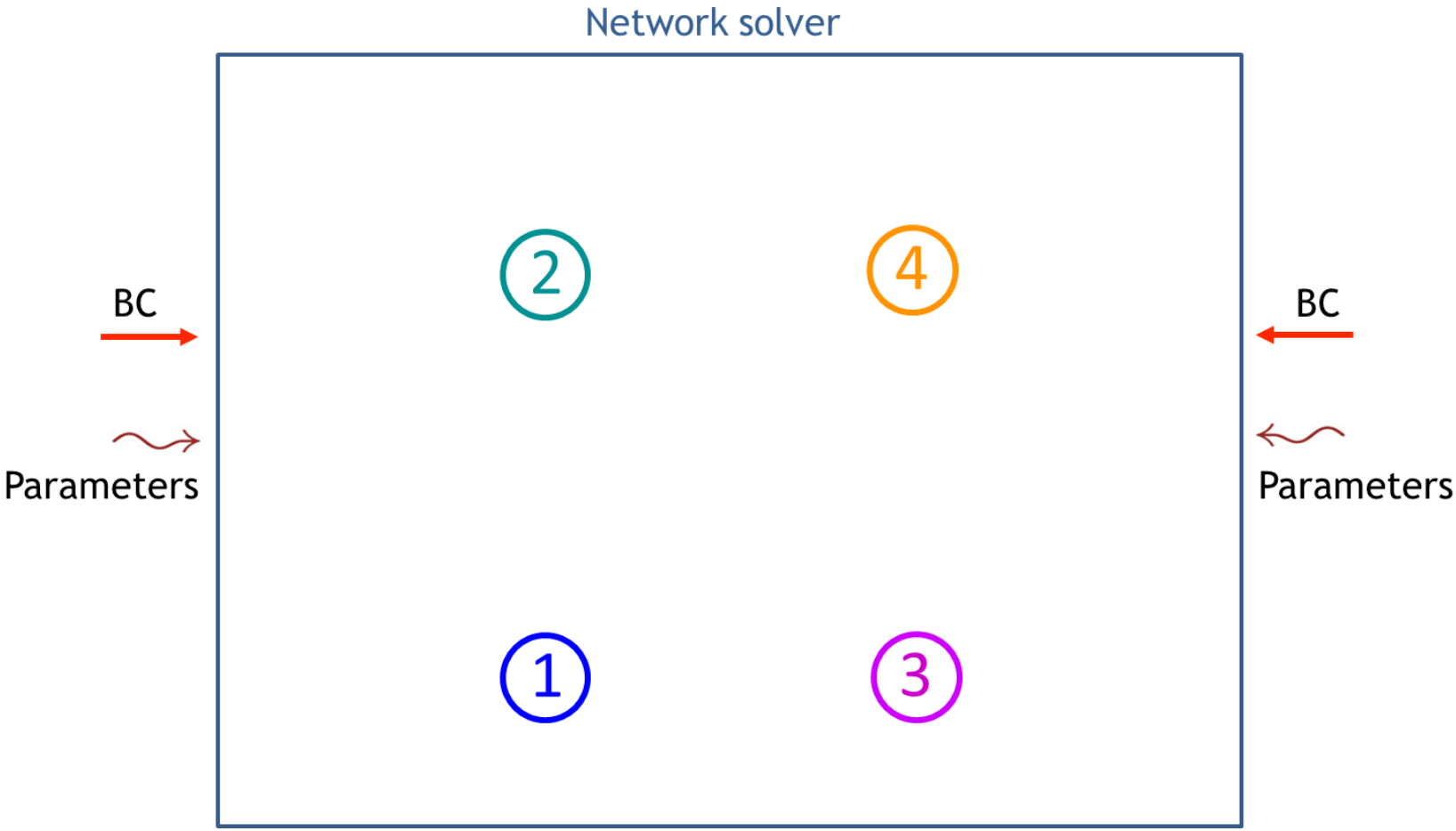


- Design parameters (length/width/...)
- I/O ports (endogenous/exogenous)
- Deterministic and stochastic info (det. solver/stochastic parametrization/...)
- Topology



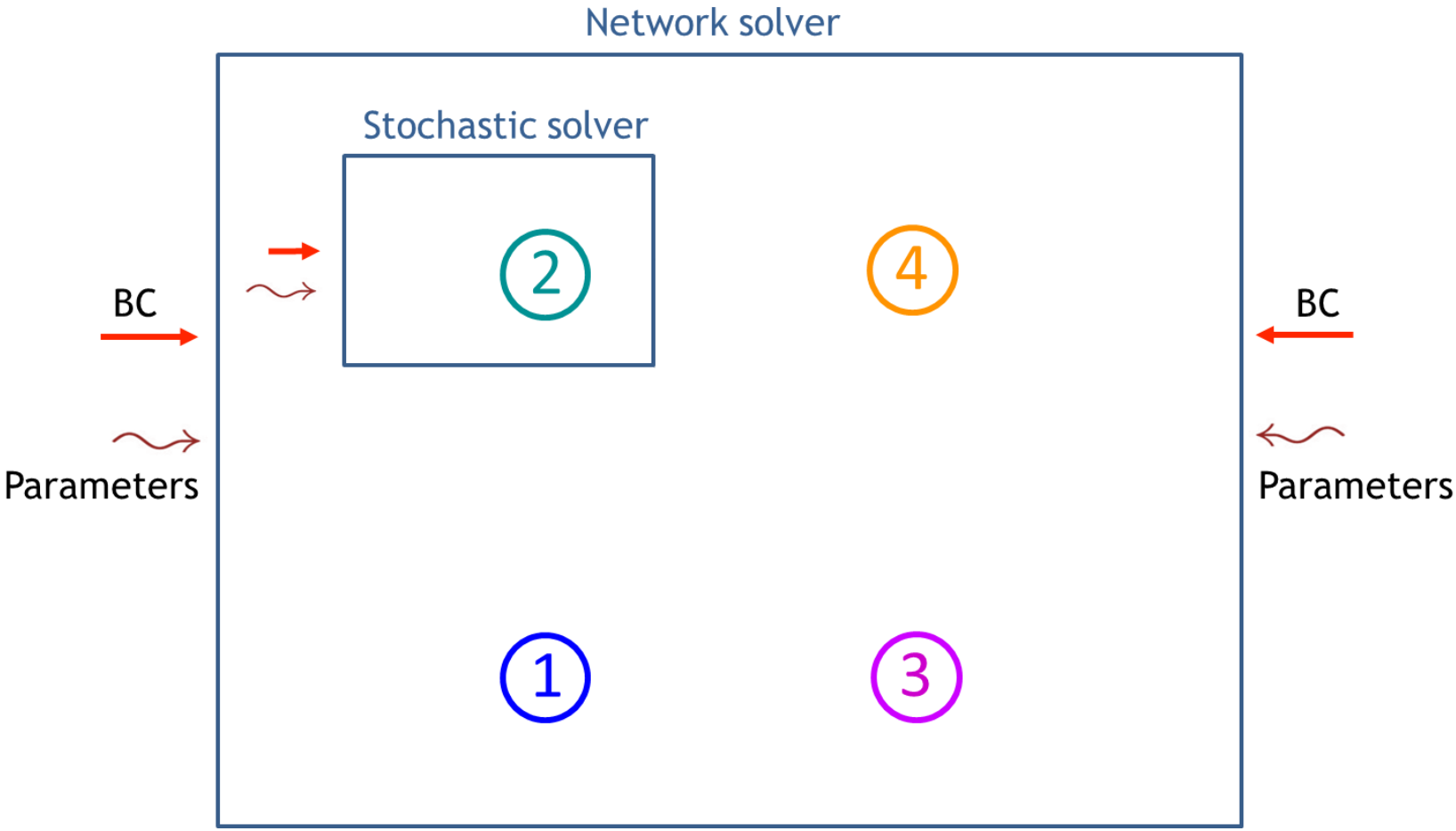
DDUQ solver

- **Online** phase



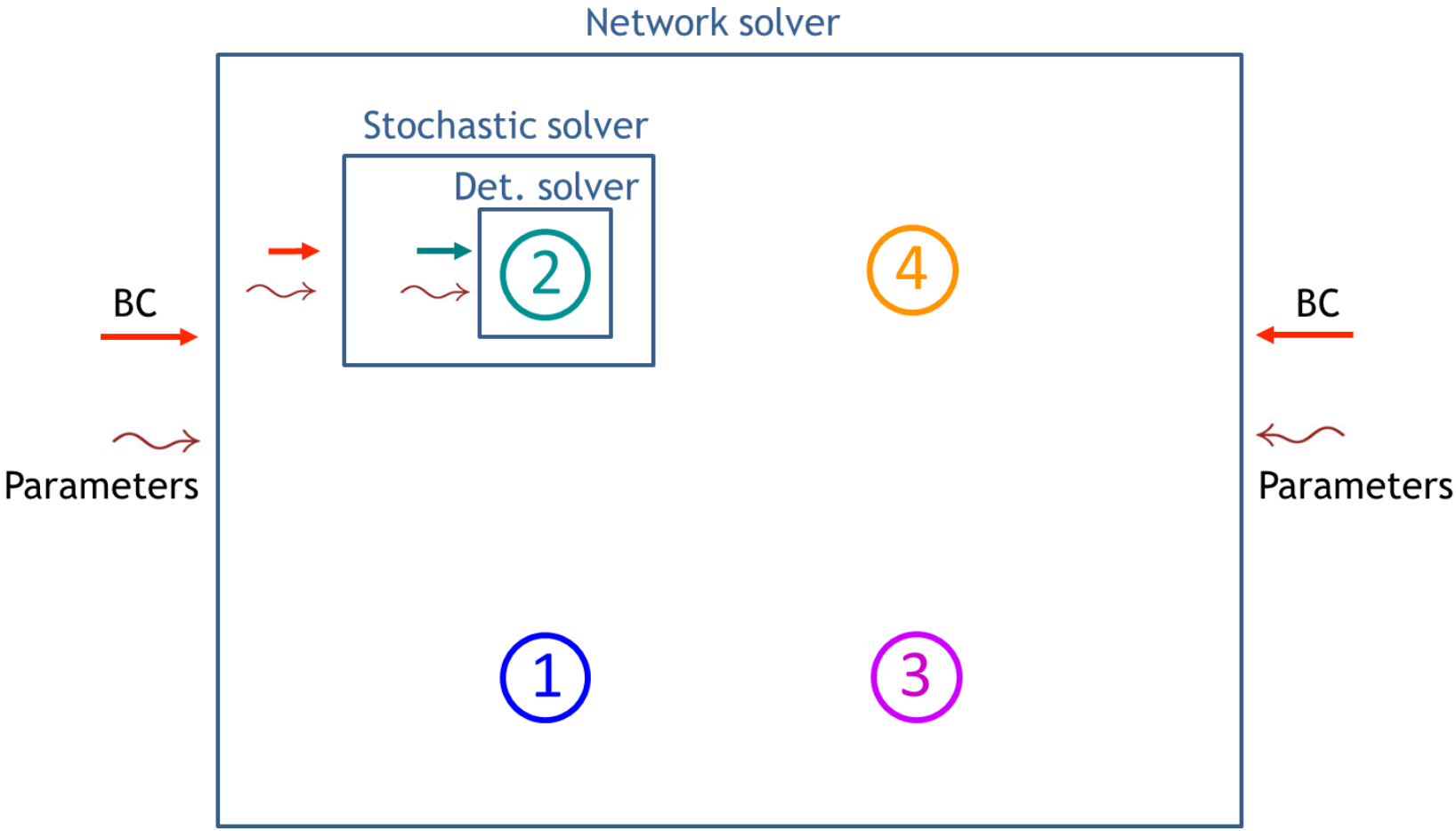
DDUQ solver

- **Online** phase



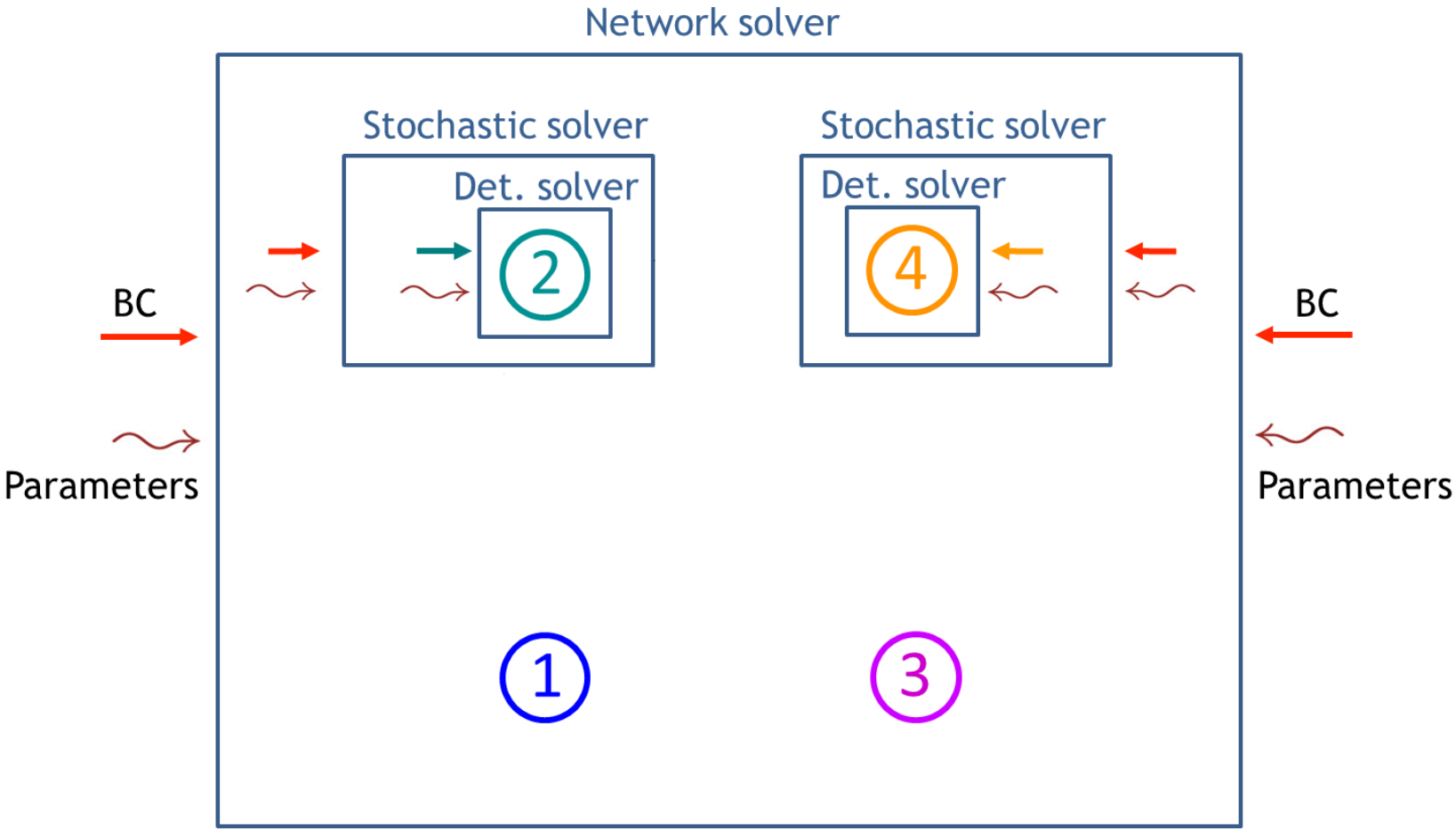
DDUQ solver

- **Online** phase



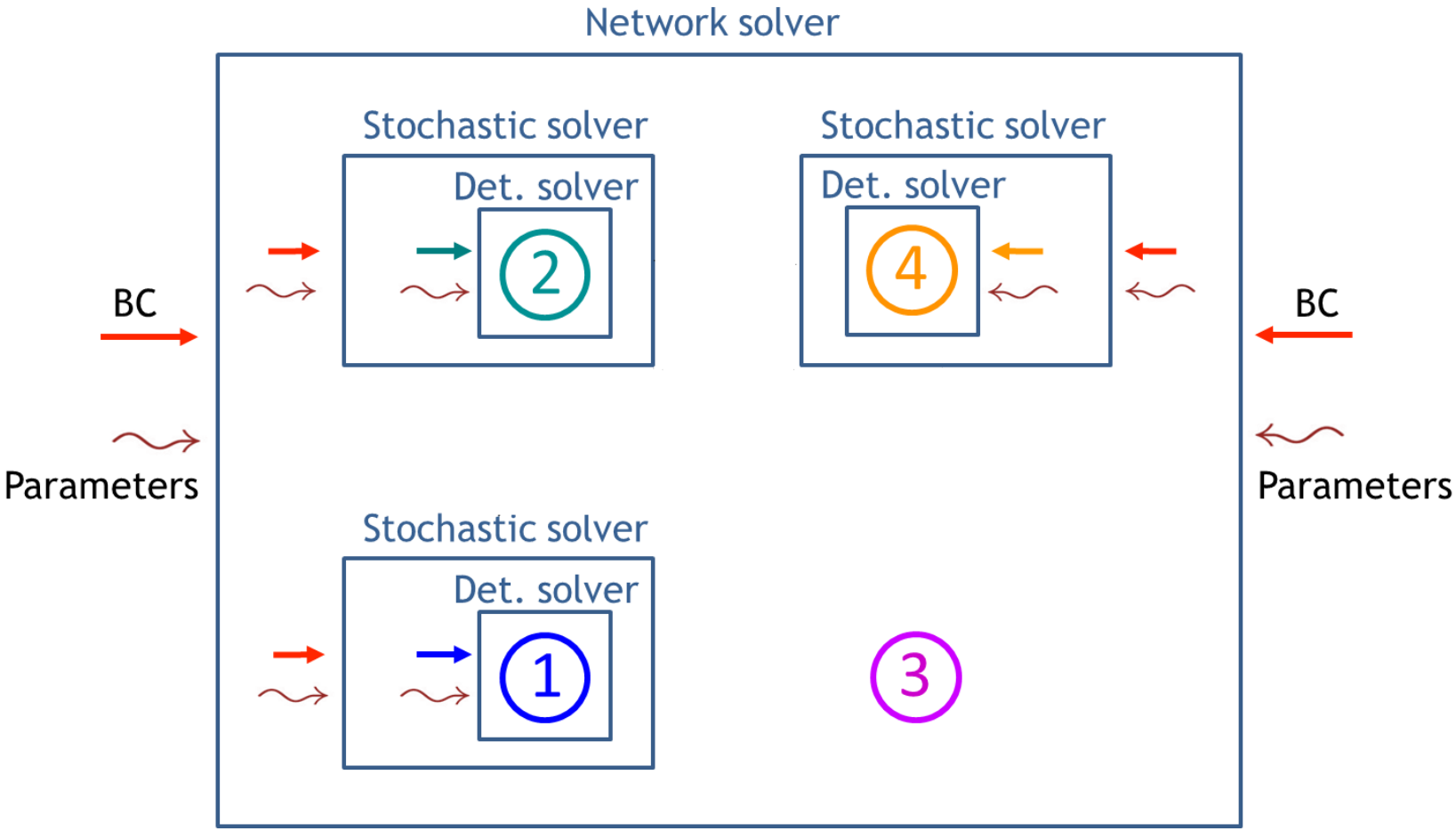
DDUQ solver

- **Online** phase



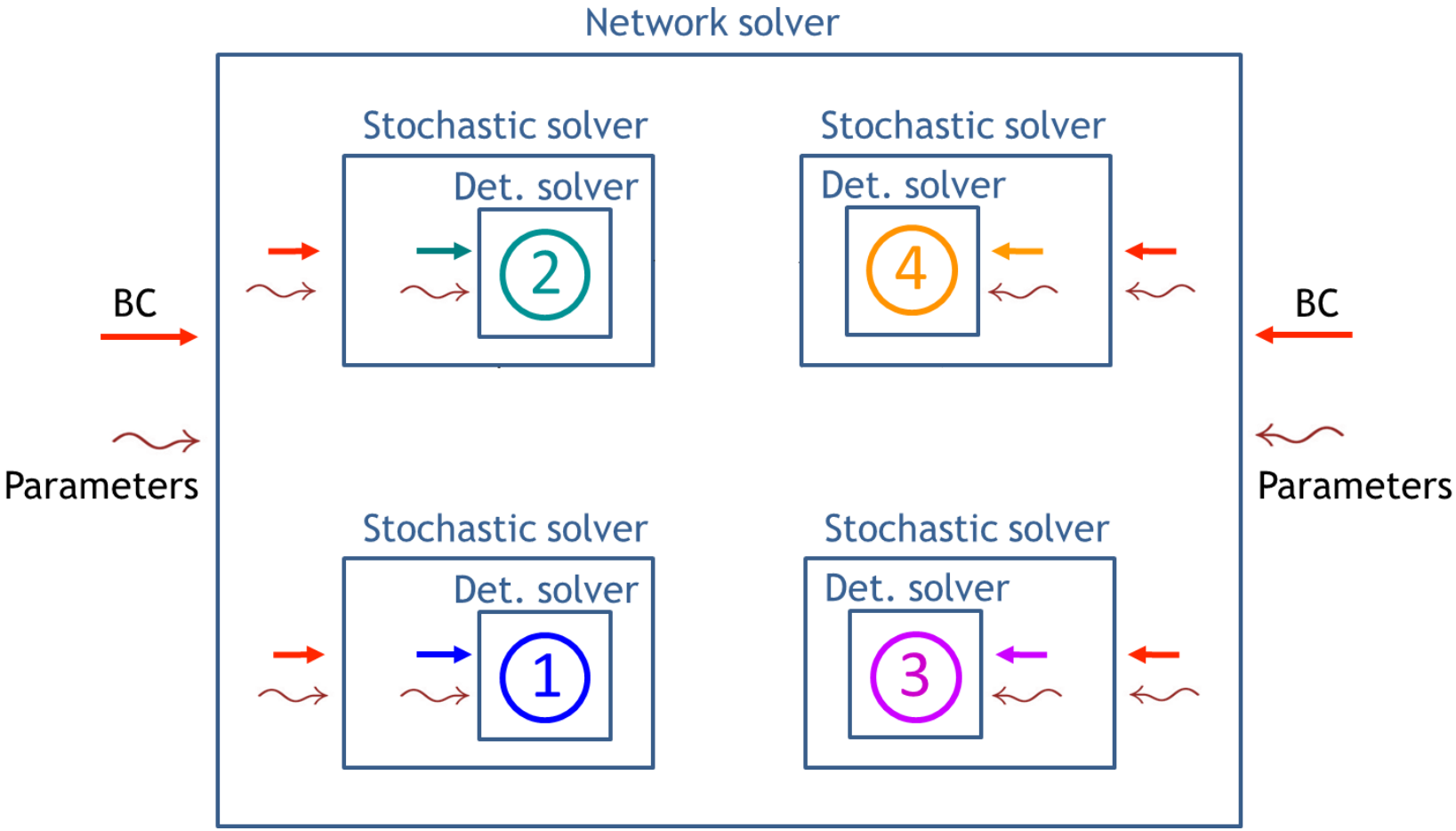
DDUQ solver

- **Online** phase



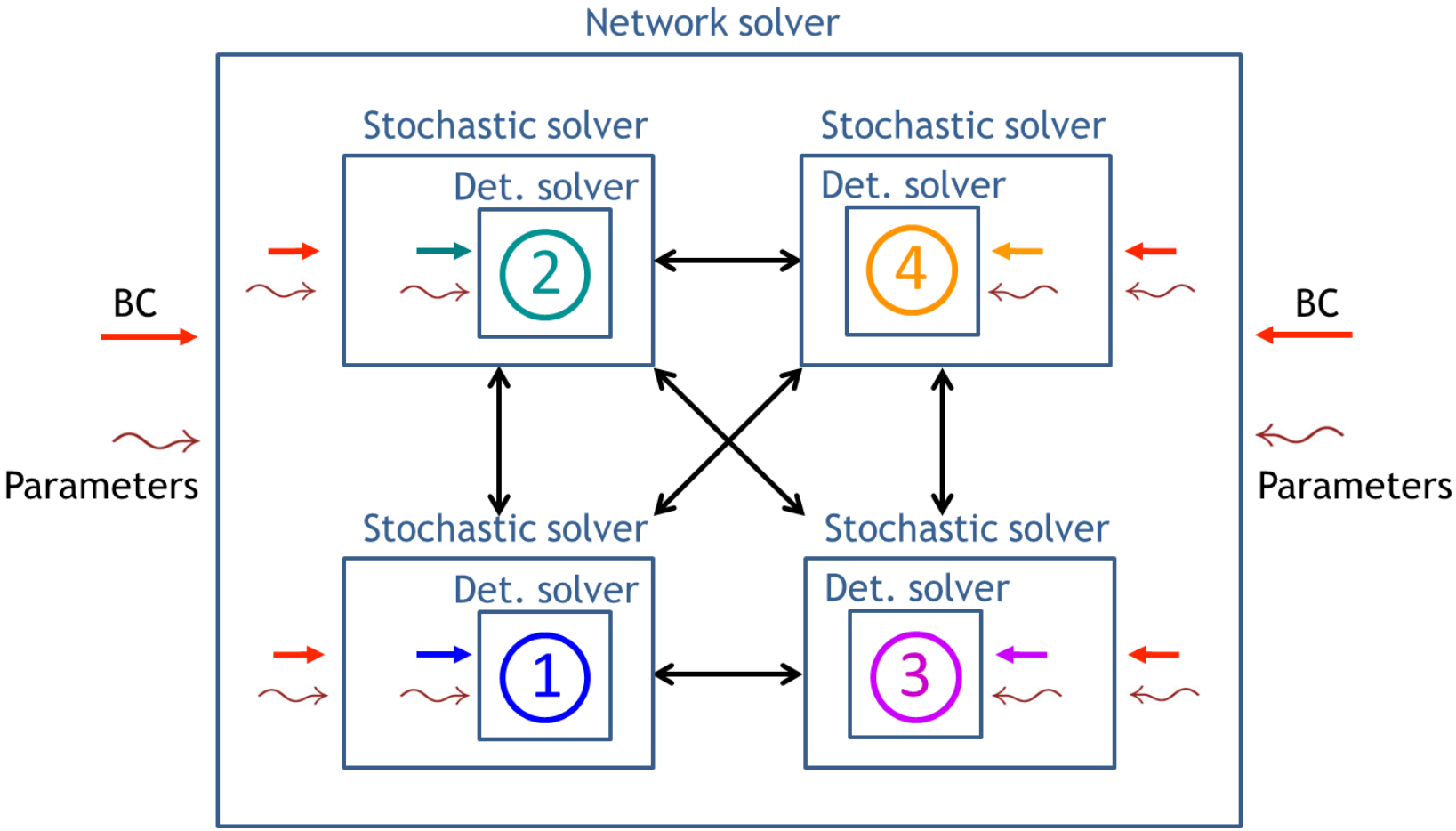
DDUQ solver

- **Online** phase



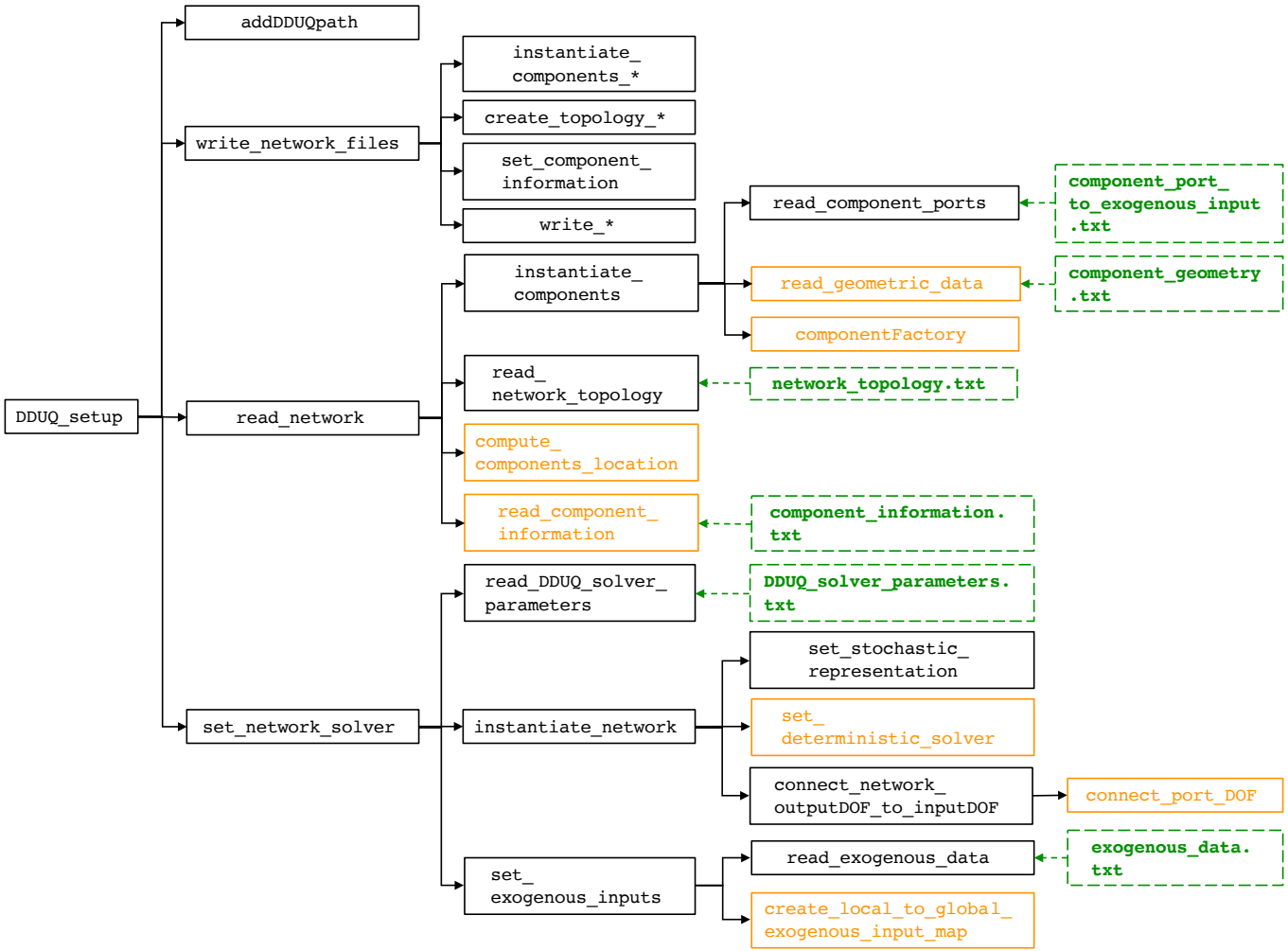
DDUQ solver

- **Online** phase



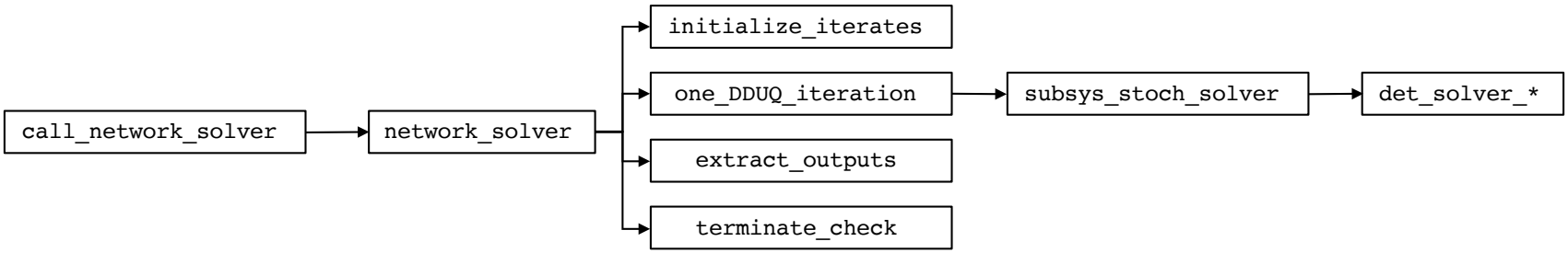
DDUQTk

Matlab toolbox for uncertainty propagation in networks via DD.
Based on `UQTk` (www.sandia.gov/uqtoolkit/).



DDUQTk

Matlab toolbox for uncertainty propagation in networks via DD.
Based on `UQTk` (www.sandia.gov/uqtoolkit/).



Numerical tests

- Linear 1D heat equation
- Incompressible 2D cavity flow
- Non-linear 2D heat equation

$$-\nabla^2 u(x,y) + s(u(x,y);\mu) = A \sin\left(\frac{2\pi}{L_x}x\right) \sin\left(\frac{2\pi}{L_y}y\right),$$

with

$$s(u;\mu) = \frac{\mu_1}{\mu_2}\left(e^{\mu_2 u} - 1\right).$$

Parameters setting:

- $\mu_2(\xi_1, \xi_2) = 1 + 0.1(\xi_2^2 - 1)$
- BC $\sim \mathcal{N}(0, 0.3)$
- $\mu_1 = 0.1$
- $A = 10$
- $L_x = 1,\ L_y = 1$
- Stopping criterion: PCE coefficients, $\varepsilon = 5e-2$

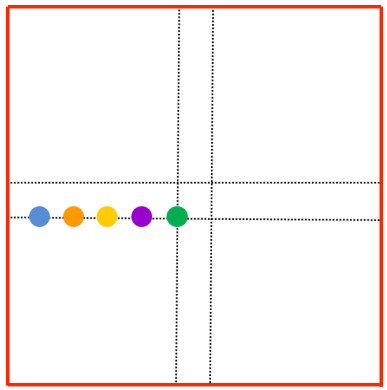


Figure: Locations of QoI

Non-linear 2D heat problem

Iterations

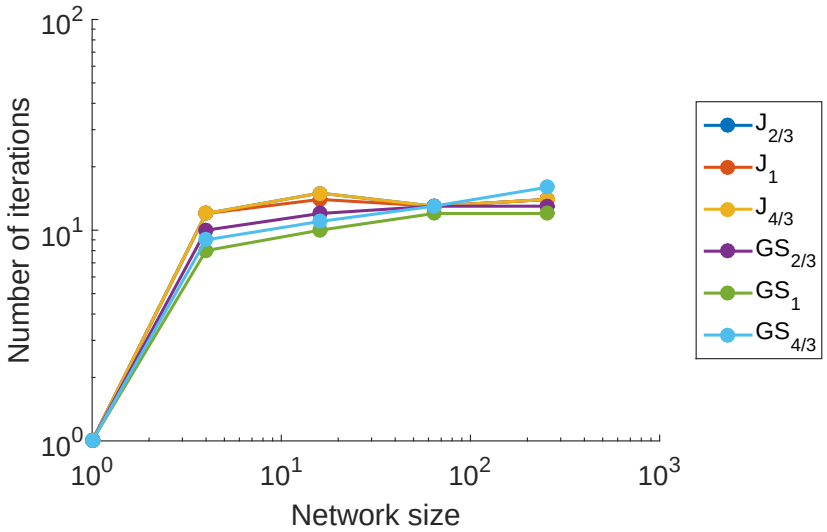


Figure: Number of iterations for different solvers.

- Iteration count roughly **independent** of network size

Accuracy

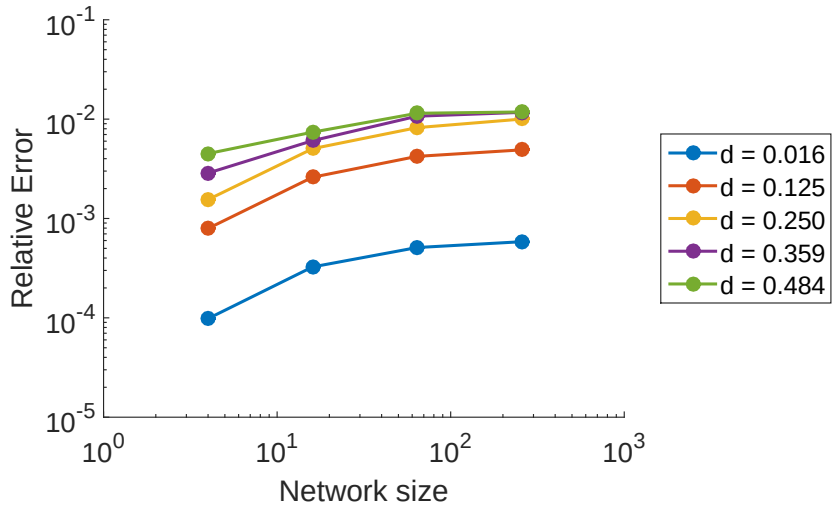


Figure: Relative error (normalized wrt the global solution) vs. network size.

- Relative error **bounded** for all network sizes
- Dependence on the **location** of the QoI (d: distance from the boundary)

Non-linear 2D heat problem

Serial Time

$$T_S(n) = N_{it}(n) \times n \times c(n)$$

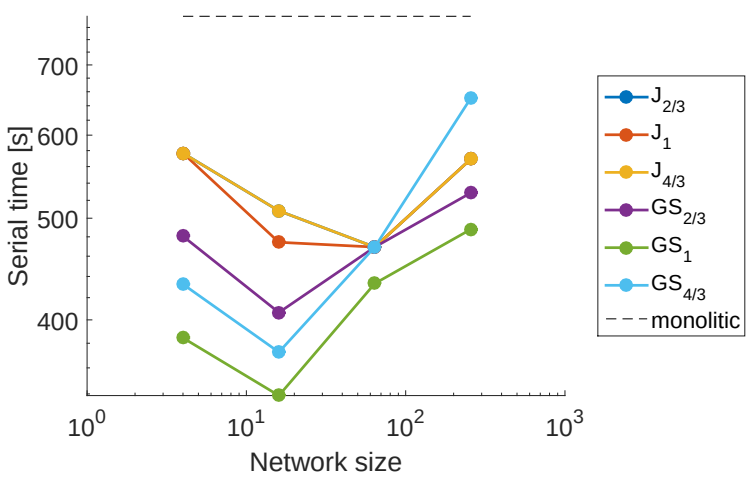


Figure: Serial time

- **Non-monotonic** trend
- **Gauss-Seidel** faster than Jacobi

Parallel Time

$$T_P(n) = N_{it}(n) \times n_P \times c(n)$$

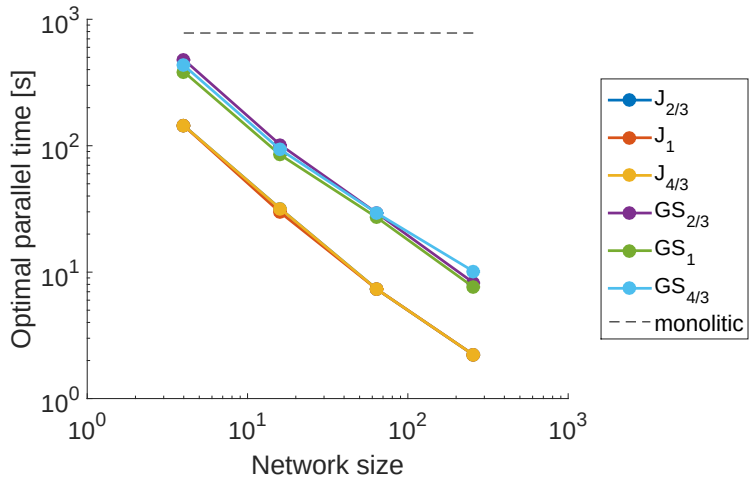


Figure: Ideal parallel time

- Strong scaling **efficiency**
- **Jacobi** faster than Gauss-Seidel

Summary

- Uncertainty Quantification (UQ)
- UQ in networks
- UQ via Domain Decomposition (DDUQ)
- `DD-UQTk`

Summary

- Uncertainty Quantification (UQ)
- UQ in networks
- UQ via Domain Decomposition (DDUQ)
- `DD-UQTk`

Conclusions

Bottom-up approach for UQ in **decomposable systems** is promising because

- No full-system simulations
- Targeted uncertainty reduction
- Modular analysis and design

On-going work

- Different types of propagations
- Different stochastic representations on different components
- Anderson Acceleration for the network solver
- Network MultiGrid
- Error analysis

Thank you!