

SAND2018-7722C

# MACHINE-LEARNING ERROR MODELS FOR APPROXIMATE SOLUTIONS TO PARAMETERIZED SYSTEMS OF NONLINEAR EQUATIONS

Brian A. Freno  
Kevin T. Carlberg  
Sandia National Laboratories

13th World Congress in Computational Mechanics  
July 25, 2018

# Outline

- Introduction
- Parameterized Nonlinear Algebraic Equations
- Proposed Approach
- Numerical Experiments
- Summary

# Outline

- Introduction
  - Motivation
  - Solution Approximations
  - Uncertainty Quantification
- Parameterized Nonlinear Algebraic Equations
- Proposed Approach
- Numerical Experiments
- Summary

# Motivation

- Many-query problems can impose a formidable computational burden
- **Solution approximations** can exchange fidelity for speed

# Solution Approximations

- **Inexact solutions:** When solving nonlinear equations, prematurely end the iterative process
- **Lower-fidelity models:** Neglect physical phenomena, coarsen the mesh, or use lower-order finite differences or elements
- **Reduced-order models:** Decompose the solution into a linear combination of  $m_{\mathbf{u}} \ll N_{\mathbf{u}}$  basis functions

# Uncertainty Quantification

- Solution approximations require **less time** than high-fidelity models but **introduce an error** (i.e. epistemic uncertainty)
- Ultimate task should account for **all sources of uncertainty**
- We quantify the uncertainty by
  - 1) engineering **features** informative of the error
    - cheaply computable
    - generated by approximate model
  - 2) applying **machine learning regression** techniques to construct statistical model of the error from these features
- This work matures our previously developed capabilities:
  - Hand-selecting one feature and applying Gaussian process regression  
M. Drohmann and K. Carlberg (2015)
  - Modeling dynamical systems error using machine learning methods  
S. Trehan et al. (2017)

# Outline

- Introduction
- Parameterized Nonlinear Algebraic Equations
  - Overview
  - Approximate Solutions
  - Approaches for Error Quantification
- Proposed Approach
- Numerical Experiments
- Summary

# Parameterized Nonlinear Algebraic Equations

Parameterized systems of nonlinear algebraic equations

$$\mathbf{r}_\star(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{0}$$

- $\mathbf{r}_\star : \mathbb{R}^{N_{\mathbf{u}}} \times \mathbb{R}^{N_{\boldsymbol{\mu}}} \rightarrow \mathbb{R}^{N_{\mathbf{u}}}$  residual, nonlinear in at least  $\mathbf{u}(\boldsymbol{\mu})$
- $\mathbf{u} : \mathbb{R}^{N_{\boldsymbol{\mu}}} \rightarrow \mathbb{R}^{N_{\mathbf{u}}}$  state (solution vector)
- $\boldsymbol{\mu} \in \mathcal{D}$  parameters in parameter domain  $\mathcal{D} \subseteq \mathbb{R}^{N_{\boldsymbol{\mu}}}$

# Quantity of Interest

Scalar-valued quantity of interest

$$s(\boldsymbol{\mu}) \equiv g(\mathbf{u}(\boldsymbol{\mu}))$$

- $s : \mathbb{R}^{N_\mu} \rightarrow \mathbb{R}$  quantity of interest
- $g : \mathbb{R}^{N_u} \rightarrow \mathbb{R}$  dependency of the quantity of interest upon the state

# Approximate Solutions

- Computing the exact solution  $\mathbf{u}(\boldsymbol{\mu})$  can be
  - prohibitively expensive (large  $N_{\mathbf{u}}$ )
  - unnecessary (inexact solutions suffice for optimization convergence)
- Such cases require an approximate solution  $\tilde{\mathbf{u}} : \mathbb{R}^{N_{\boldsymbol{\mu}}} \rightarrow \mathbb{R}^{N_{\mathbf{u}}}$
- Approximate solution leads to approximated quantity of interest

$$\tilde{s}(\boldsymbol{\mu}) \equiv g(\tilde{\mathbf{u}}(\boldsymbol{\mu})),$$

where  $\tilde{s} : \mathbb{R}^{N_{\boldsymbol{\mu}}} \rightarrow \mathbb{R}$

# Approximate Solutions (continued)

We consider 3 approaches for computing approximate solutions:

- 1) Premature termination of nonlinear iterations
- 2) Lower-fidelity model
- 3) **Model reduction**

# Model Reduction

Model reduction restricts approximate solution  $\tilde{\mathbf{u}}$  to  $m_{\mathbf{u}}$ -dimensional affine trial subspace  $\bar{\mathbf{u}} + \text{Ran}(\Phi_{\mathbf{u}}) \subseteq \mathbb{R}^{N_{\mathbf{u}}}$  with  $m_{\mathbf{u}} \ll N_{\mathbf{u}}$ :

$$\tilde{\mathbf{u}}(\boldsymbol{\mu}) = \bar{\mathbf{u}} + \Phi_{\mathbf{u}} \hat{\mathbf{u}}(\boldsymbol{\mu})$$

- $\bar{\mathbf{u}} \in \mathbb{R}^{N_{\mathbf{u}}}$  reference state
- $\Phi_{\mathbf{u}} \in \mathbb{R}_{\star}^{N_{\mathbf{u}} \times m_{\mathbf{u}}}$  trial basis, computed using
  - proper orthogonal decomposition (POD)
  - the reduced-basis method
  - variants that employ gradient information
- $\hat{\mathbf{u}} : \mathbb{R}^{N_{\boldsymbol{\mu}}} \rightarrow \mathbb{R}^{m_{\mathbf{u}}}$  generalized coordinates of the approx. solution

# Model Reduction (continued)

- $\mathbf{r}_\star(\bar{\mathbf{u}} + \Phi_{\mathbf{u}}\hat{\mathbf{u}}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{0}$  is **overdetermined**:  $N_{\mathbf{u}}$  equations,  $m_{\mathbf{u}}$  unknowns
- Second step projects residual onto an  $m_{\mathbf{u}}$ -dimensional test subspace  $\text{Ran}(\Psi_{\mathbf{u}}) \subseteq \mathbb{R}^{N_{\mathbf{u}}}$ :

$$\Psi_{\mathbf{u}}^T \mathbf{r}_\star(\bar{\mathbf{u}} + \Phi_{\mathbf{u}}\hat{\mathbf{u}}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{0}$$

- $\Psi_{\mathbf{u}} \in \mathbb{R}_\star^{N_{\mathbf{u}} \times m_{\mathbf{u}}}$  is test basis, common choices include
  - Galerkin projection:  $\Psi_{\mathbf{u}} = \Phi_{\mathbf{u}}$
  - Least-squares Petrov–Galerkin projection:  $\Psi_{\mathbf{u}} = \frac{\partial \mathbf{r}_\star}{\partial \mathbf{u}}(\bar{\mathbf{u}} + \Phi_{\mathbf{u}}\hat{\mathbf{u}}(\boldsymbol{\mu}); \boldsymbol{\mu})\Phi_{\mathbf{u}}$

# Approaches for Error Quantification

- Regardless of approach, it is essential to quantify error incurred by employing approximate solution  $\tilde{\mathbf{u}}$  in lieu of exact solution  $\mathbf{u}$
- Existing approaches include
  - Data-fit mapping between parameters and the error
    - Inspired by multifidelity design optimization
  - Reduced-Order Model Error Surrogates (ROMES) method
    - M. Drohmann and K. Carlberg, 2015
    - Quantity of interest error approximated using dual-weighted residuals
    - Normed state-space error approx. using residual norm and error bounds
- This work focuses on quantifying two errors:
  - 1) Error in quantity of interest:  $\delta_s(\boldsymbol{\mu}) \equiv s(\boldsymbol{\mu}) - \tilde{s}(\boldsymbol{\mu})$
  - 2) Normed state-space error:  $\delta_{\mathbf{u}}(\boldsymbol{\mu}) \equiv \|\mathbf{e}(\boldsymbol{\mu})\|_2$ , where  $\mathbf{e}(\boldsymbol{\mu}) \equiv \mathbf{u}(\boldsymbol{\mu}) - \tilde{\mathbf{u}}(\boldsymbol{\mu})$

# State-Space Error

The residual can be approximated about the approximate solution  $\tilde{\mathbf{u}}$ :

$$\mathbf{r}_*(\mathbf{u}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{0} = \mathbf{r}(\boldsymbol{\mu}) + \mathbf{J}(\boldsymbol{\mu})\mathbf{e}(\boldsymbol{\mu}) + \mathcal{O}(\|\mathbf{e}(\boldsymbol{\mu})\|^2)$$

and rearranged to approximate the state-space error:

$$\mathbf{e}(\boldsymbol{\mu}) = -\mathbf{J}(\boldsymbol{\mu})^{-1}\mathbf{r}(\boldsymbol{\mu}) + \mathcal{O}(\|\mathbf{e}(\boldsymbol{\mu})\|^2)$$

- $\mathbf{r}(\boldsymbol{\mu}) \equiv \mathbf{r}_*(\tilde{\mathbf{u}}(\boldsymbol{\mu}); \boldsymbol{\mu})$  residual from approximate solution
- $\mathbf{J}(\boldsymbol{\mu}) \equiv \frac{\partial \mathbf{r}_*}{\partial \mathbf{u}}(\tilde{\mathbf{u}}(\boldsymbol{\mu}); \boldsymbol{\mu}) \in \mathbb{R}^{N_{\mathbf{u}} \times N_{\mathbf{u}}}$  Jacobian of residual at  $\tilde{\mathbf{u}}(\boldsymbol{\mu})$

# Error in the Quantity of Interest

The quantity of interest also can be approximated:

$$s(\boldsymbol{\mu}) = \tilde{s}(\boldsymbol{\mu}) + \frac{\partial g}{\partial \mathbf{u}}(\tilde{\mathbf{u}}(\boldsymbol{\mu}))\mathbf{e}(\boldsymbol{\mu}) + \mathcal{O}(\|\mathbf{e}(\boldsymbol{\mu})\|^2)$$

and combined with the state-space error approximation to yield

$$\delta_s(\boldsymbol{\mu}) = \underbrace{-\frac{\partial g}{\partial \mathbf{u}}(\tilde{\mathbf{u}}(\boldsymbol{\mu}))\mathbf{J}(\boldsymbol{\mu})^{-1}}_{\mathbf{y}(\boldsymbol{\mu})^T} \mathbf{r}(\boldsymbol{\mu}) + \mathcal{O}(\|\mathbf{e}(\boldsymbol{\mu})\|^2)$$

- $\mathbf{y}(\boldsymbol{\mu})$  is the dual or adjoint
- dual-weighted residual  $d$  is weighted sum of residual elements:

$$d(\boldsymbol{\mu}) \equiv \mathbf{y}(\boldsymbol{\mu})^T \mathbf{r}(\boldsymbol{\mu})$$

# Drawbacks to using the Dual-Weighted Residual

- **Computational Cost:** requires solving  $N_{\mathbf{u}}$  linear equations
- **Implementation:** requires Jacobian – not always available
- **Uncertainty Quantification:** low-bias error estimate not assured

Nonetheless, structure provides insight into quantity-of-interest error

## Normed State-Space Error

- Residual-based bounds commonly used to quantify  $\delta_{\mathbf{u}}(\boldsymbol{\mu})$   
A. Buffa et al., 2012; M. A. Grepl and A. T. Patera, 2005; G. Rozza et al., 2008

- Assuming Lipschitz continuity for the residual  $\mathbf{r}_{\star}(\cdot; \boldsymbol{\mu})$ , then

$$\frac{\|\mathbf{r}(\boldsymbol{\mu})\|_2}{\beta(\boldsymbol{\mu})} \leq \delta_{\mathbf{u}}(\boldsymbol{\mu}) \leq \frac{\|\mathbf{r}(\boldsymbol{\mu})\|_2}{\alpha(\boldsymbol{\mu})},$$

where  $\alpha$  and  $\beta$  are Lipschitz constants

- Drawbacks to using error bounds
  - **Sharpness:** upper/lower bounds can overpredict/underpredict actual error by several orders of magnitude
  - **Implementation:** difficult to compute true Lipschitz constants
  - **Uncertainty Quantification:** do not produce statistical distribution over  $\delta_{\mathbf{u}}(\boldsymbol{\mu})$  – cannot quantify epistemic uncertainty

# Outline

- Introduction
- Parameterized Nonlinear Algebraic Equations
- Proposed Approach
  - Overview
  - Feature Engineering
  - Regression-Function Approximation
  - Training and Test Data
- Numerical Experiments
- Summary

# Overview

- We aim to construct statistical models of
  - quantity-of-interest error  $\delta_s$
  - normed state-space error  $\delta_{\mathbf{u}}$
- We apply high-dimensional regression methods from machine learning
- We use a large number of **inexpensive** error indicators, resulting in **less costly**, more **accurate** error models

# Error Model

- Assume there exist  $N_{\mathbf{x}}$  *error indicators* or *features*  $\mathbf{x}(\boldsymbol{\mu}) \in \mathbb{R}^{N_{\mathbf{x}}}$ 
  - **available** from solution approximation
  - **cheaply computable**
  - **informative** of the error  $\delta(\boldsymbol{\mu}) \in \mathbb{R}$
- We model the nondeterministic mapping  $\mathbf{x}(\boldsymbol{\mu}) \mapsto \delta(\boldsymbol{\mu})$

$$\delta(\boldsymbol{\mu}) = f(\mathbf{x}(\boldsymbol{\mu})) + \epsilon(\mathbf{x}(\boldsymbol{\mu}))$$

- $f$ : deterministic regression function
- $\epsilon$ : nondeterministic noise
  - Mean-zero random variable
  - Accounts for irreducible error due to missing features
  - Epistemic – additional features can enable zero noise

# Regression Model

- Regression function defines conditional expectation of error given the features:

$$E[\delta(\boldsymbol{\mu}) \mid \mathbf{x}(\boldsymbol{\mu})] = f(\mathbf{x}(\boldsymbol{\mu}))$$

- We construct approximations of
  - deterministic regression function  $\hat{f}(\approx f)$
  - nondeterministic noise  $\hat{\epsilon}(\approx \epsilon)$ ,

which yield a statistical model for the approximate-solution error

$$\hat{\delta}(\boldsymbol{\mu}) = \hat{f}(\mathbf{x}(\boldsymbol{\mu})) + \hat{\epsilon}(\mathbf{x}(\boldsymbol{\mu}))$$

# Regression Model Objectives

- **Cheap:** Should employ cheaply computable features  $\mathbf{x}$
- **Low Noise Variance:** Should exhibit low noise variance, reduce epistemic uncertainty introduced by approximate solution
- **Numerically Validated:** Empirical distributions of  $\hat{\delta}$  and  $\delta$  should be close on test set **not** used to train model – should not overfit on training data

# Regression Model Construction Steps

## 1) Feature engineering

- Cheaply computable features  $\mathbf{x}$  from approximate model
- Informative of the error – construct low-noise-variance model
- Low dimensional (small  $N_{\mathbf{x}}$ ) such that less training data is needed

## 2) Regression-function approximation

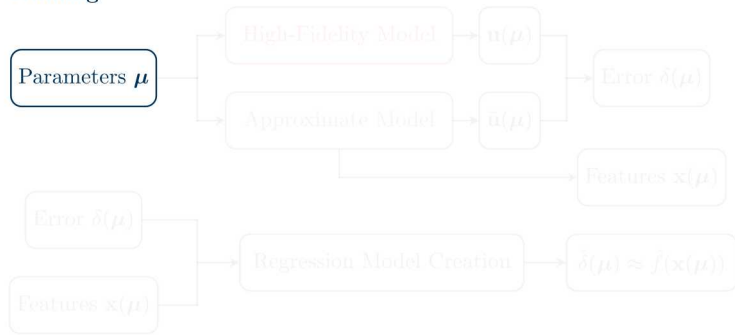
- Construct  $\hat{f}$  using methods from machine learning
- Approximate mapping from features  $\mathbf{x}$  to error  $\delta$  on a training set

## 3) Noise approximation

- Mean-zero, constant-variance Gaussian random variable:  $\hat{\epsilon} \sim \mathcal{N}(0, \hat{\sigma}^2)$
- $\hat{\sigma}^2$  is sample variance of regression-model noise on test set  
(mean squared error on test set)

# Summary

## Training

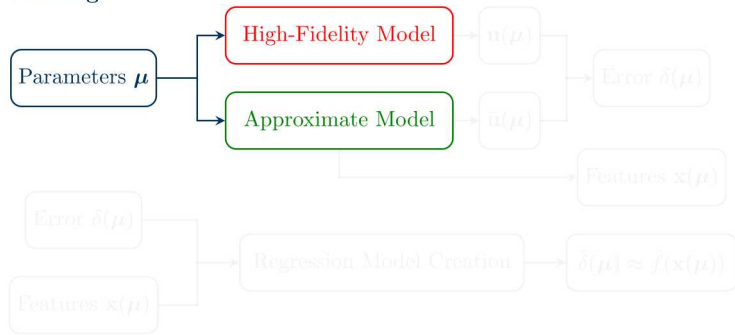


## Application



# Summary

## Training

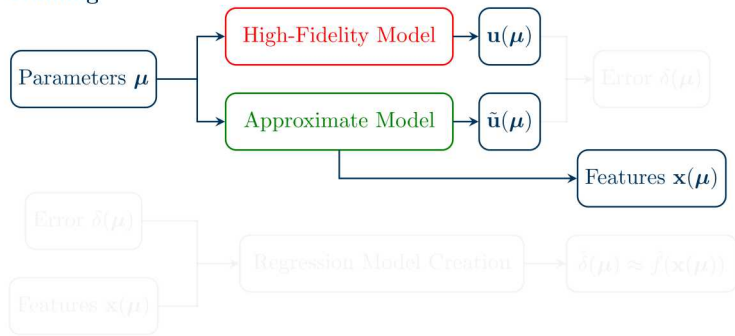


## Application



# Summary

## Training

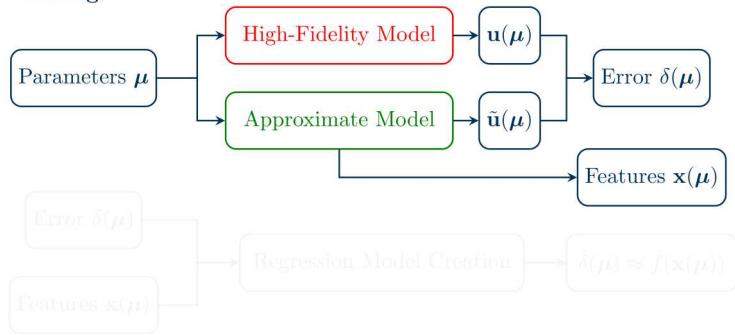


## Application



# Summary

## Training

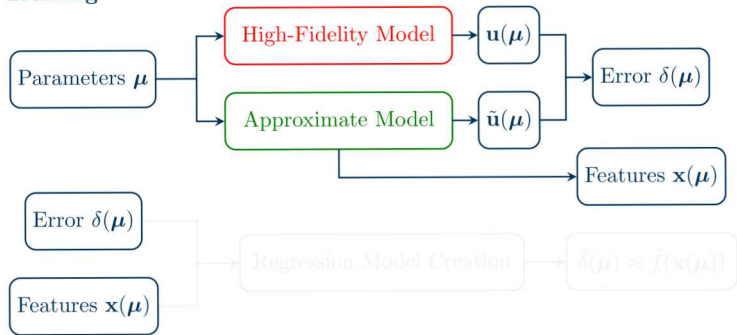


## Application



# Summary

## Training

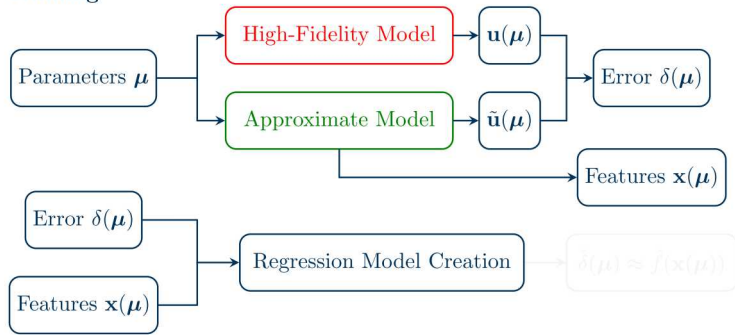


## Application



# Summary

## Training

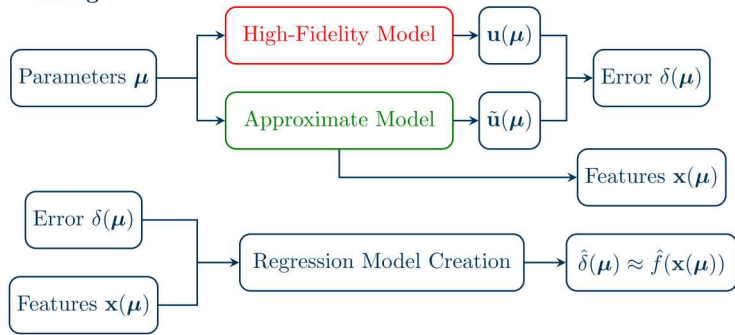


## Application



# Summary

## Training

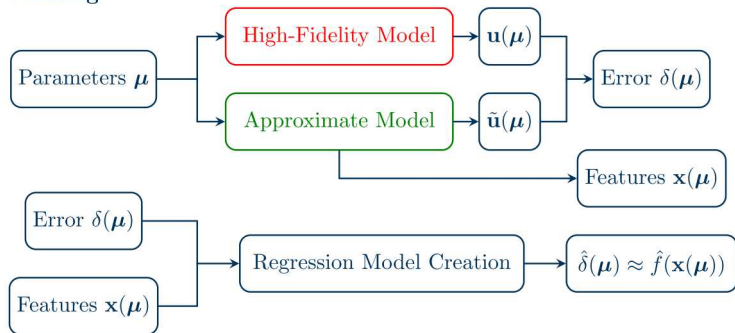


## Application



# Summary

## Training

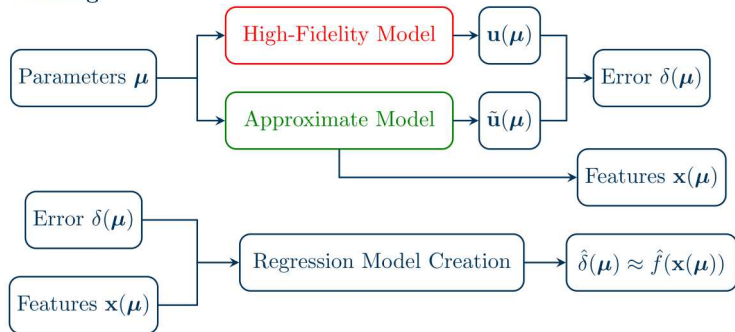


## Application

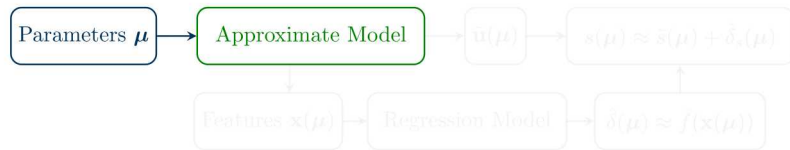


# Summary

## Training

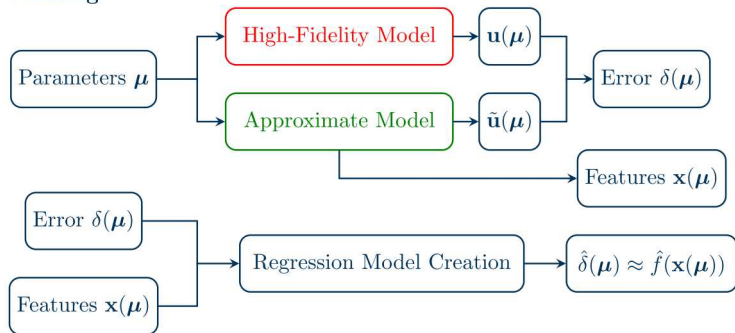


## Application

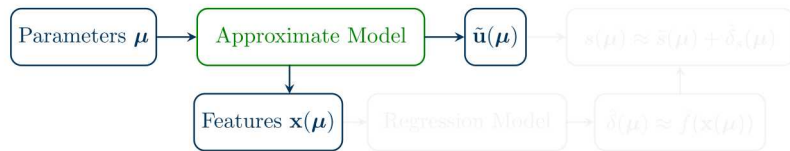


# Summary

## Training

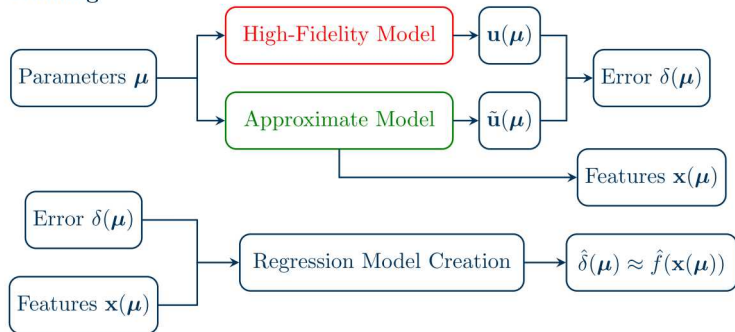


## Application

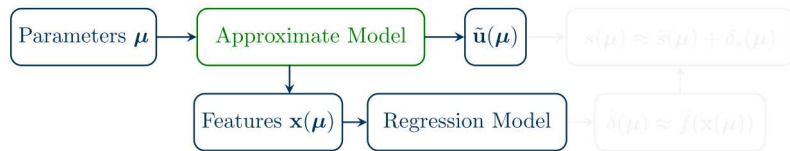


# Summary

## Training

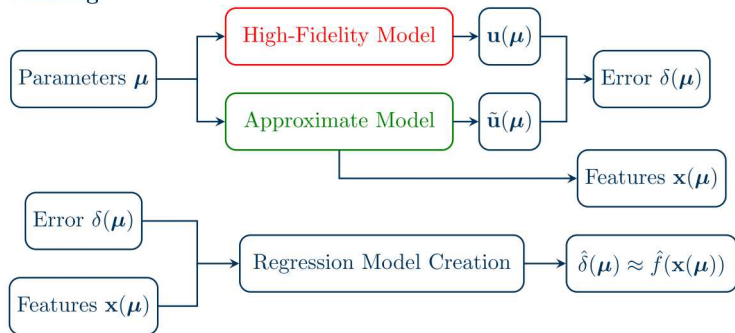


## Application

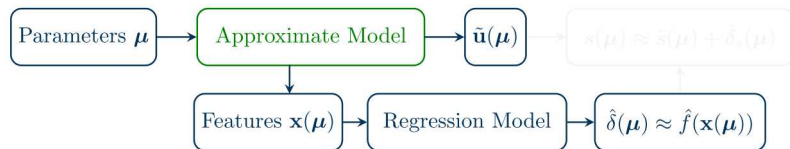


# Summary

## Training

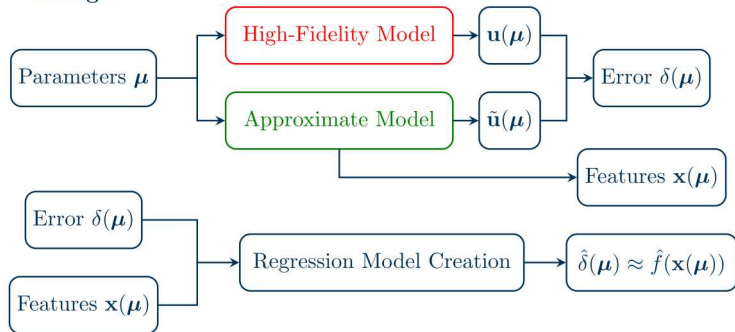


## Application

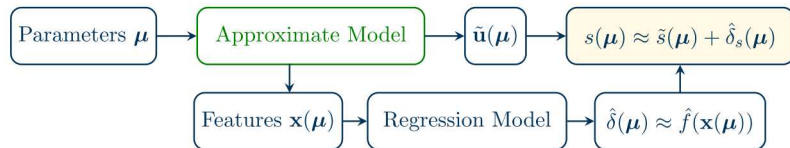


# Summary

## Training



## Application



# Feature Engineering: Parameters

$$\mathbf{x}(\boldsymbol{\mu}) = \boldsymbol{\mu}$$

- The mapping  $\boldsymbol{\mu} \mapsto \delta(\boldsymbol{\mu})$  is **deterministic**, but often **complex**
  - Can be **oscillatory** for ROMs since  $\delta(\boldsymbol{\mu}) \approx 0$  when  $\boldsymbol{\mu} \in \mathcal{D}_{\text{Train}}^{\text{ROM}}$
- Could yield **zero** noise variance if
  - **Large** amounts of training data
  - Sufficiently flexible regression model
- Low-quality features
- Used by ‘multifidelity correction’ methods for optimization  
Alexandrov et al., 2001; Gano et al., 2005; Eldred et al., 2004

# Feature Engineering: Dual-Weighted Residual

$$\mathbf{x}(\boldsymbol{\mu}) = d(\boldsymbol{\mu}) \equiv \mathbf{y}(\boldsymbol{\mu})^T \mathbf{r}(\boldsymbol{\mu})$$

- **Second-order-accurate** approximation of QoI error  $\delta_s(\boldsymbol{\mu})$
- **Small number** ( $N_{\mathbf{x}} = 1$ ) of **high-quality** features
- **High** computational cost and **significant** implementation effort
- ROMES method uses approximation for dual-weighted residual

M. Drohmann and K. Carlberg, 2015

# Feature Engineering: Parameters and Residual (Approximations)

$$\mathbf{x}(\boldsymbol{\mu}) = [\boldsymbol{\mu}; \mathbf{r}(\boldsymbol{\mu})]$$

- DWR is weighted sum of residual vector elements  $d(\boldsymbol{\mu}) \equiv \mathbf{y}(\boldsymbol{\mu})^T \mathbf{r}(\boldsymbol{\mu})$
- **Avoids** implementation and costs associated with dual vector  $\mathbf{y}(\boldsymbol{\mu})$
- **Large number** ( $N_{\mathbf{x}} = N_{\boldsymbol{\mu}} + N_{\mathbf{u}}$ ) of **low-quality** features
- Approaches to **reduce** number of features and **improve** quality
  - Use  $m_{\mathbf{r}} \ll N_{\mathbf{u}}$  principal component coefficients:  $\hat{\mathbf{r}}(\boldsymbol{\mu})$
  - Sample  $n_{\mathbf{r}} \ll N_{\mathbf{u}}$  elements of residual:  $\mathbf{P}\mathbf{r}(\boldsymbol{\mu})$ , where  $\mathbf{P} \in \{0, 1\}^{n_{\mathbf{r}} \times N_{\mathbf{u}}}$
  - Use  $m_{\mathbf{r}} \ll N_{\mathbf{u}}$  gappy principal component coefficients:  $\hat{\mathbf{r}}_{\mathbf{g}}(\boldsymbol{\mu})$

# Feature Engineering: Residual Norm with/without Parameters

$$\mathbf{x}(\boldsymbol{\mu}) = \|\mathbf{r}(\boldsymbol{\mu})\|_2 \quad \text{or} \quad \mathbf{x}(\boldsymbol{\mu}) = [\boldsymbol{\mu}; \|\mathbf{r}(\boldsymbol{\mu})\|_2]$$

- DWR can be bounded using the Cauchy–Schwarz inequality:

$$|d(\boldsymbol{\mu})| \leq \|\mathbf{y}(\boldsymbol{\mu})\|_2 \|\mathbf{r}(\boldsymbol{\mu})\|_2$$

- Normed state-space error  $\delta_{\mathbf{u}}(\boldsymbol{\mu})$  can be bounded:

M. Drohmann and K. Carlberg, 2015

$$\frac{\|\mathbf{r}(\boldsymbol{\mu})\|_2}{\beta(\boldsymbol{\mu})} \leq \delta_{\mathbf{u}}(\boldsymbol{\mu}) \leq \frac{\|\mathbf{r}(\boldsymbol{\mu})\|_2}{\alpha(\boldsymbol{\mu})}$$

- $\boldsymbol{\mu}$  can be omitted ( $\mathbf{x}(\boldsymbol{\mu}) = \|\mathbf{r}(\boldsymbol{\mu})\|_2$ ) if
  - $\boldsymbol{\mu}$  is not indicative of error
  - $N_{\boldsymbol{\mu}}$  is too large relative to training data
- Requires computing **entire** residual vector  $\mathbf{r}(\boldsymbol{\mu})$
- **Small number** of potentially **low-quality** features

# Regression-Function Approximation

We consider several different regression models

- Ordinary least squares (OLS)
  - Linear (OLS: Linear)
  - Quadratic expansion of features (OLS: Quadratic)
- Support vector regression (SVR)
  - Linear kernel (SVR: Linear)
  - Gaussian (radial basis function) kernel (SVR: RBF)
- Random forest (RF)
- $k$ -nearest neighbors ( $k$ -NN)
- Artificial neural network / multilayer perceptron (MLP)

# Training and Test Data

## Training Data

- Consists of parameter  $\mu$  subset from parameter space  $\mathcal{D}$
- High-fidelity and approximate solutions train regression models
- Cross-validated to tune regression model hyper-parameters
- Used to compute principal components of residuals

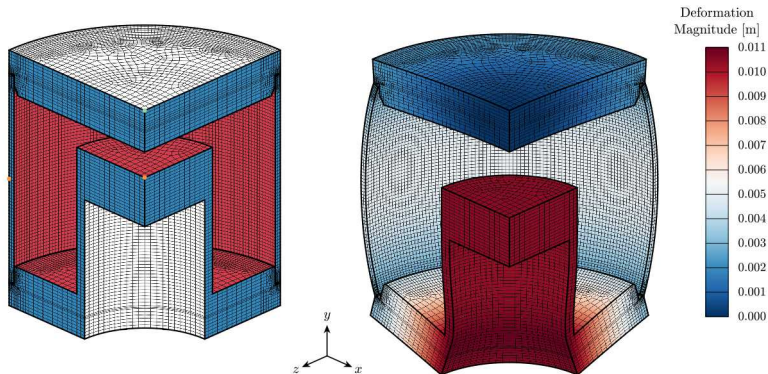
## Test Data

- Consists of parameter  $\mu$  choices **not** used for training data
- Used to assess regression models and quantify nondeterministic noise

# Outline

- Introduction
- Parameterized Nonlinear Algebraic Equations
- Proposed Approach
- Numerical Experiments
  - PCAP: Reduced-Order Modeling
- Summary

## Predictive Capability Assessment Project: Reduced-Order Modeling

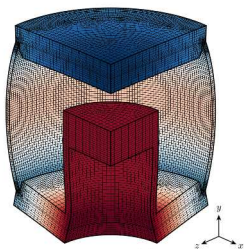


- Applied pressure (Neumann boundary condition)
- Planar constraint (Dirichlet boundary condition)
- Complete constraint (Dirichlet boundary condition)
- Nodes of interest

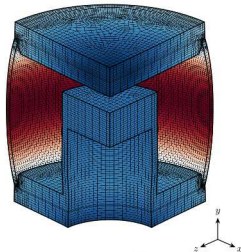
# PCAP: Overview

- $N_{\mathbf{u}} = 278,301$  for quarter of domain
- $N_{\boldsymbol{\mu}} = 3$  parameters:  $\boldsymbol{\mu} = [E; \nu; p]$ 
  - $E \in [50, 100]$  GPa,  $\nu \in [0.20, 0.35]$ ,  $p \in [6, 10]$  GPa
- 8 HF runs  $\rightarrow$  up to  $m_{\mathbf{u}} = 8$  ROM basis functions (5 used – 99.90%)
- 30 training runs for regression model

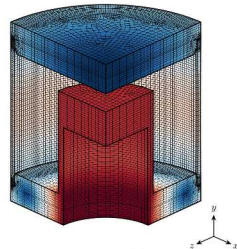
## PCAP: Basis Functions



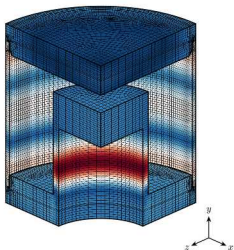
1: 85.03%



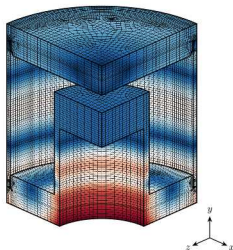
2: 95.69%



3: 99.35%

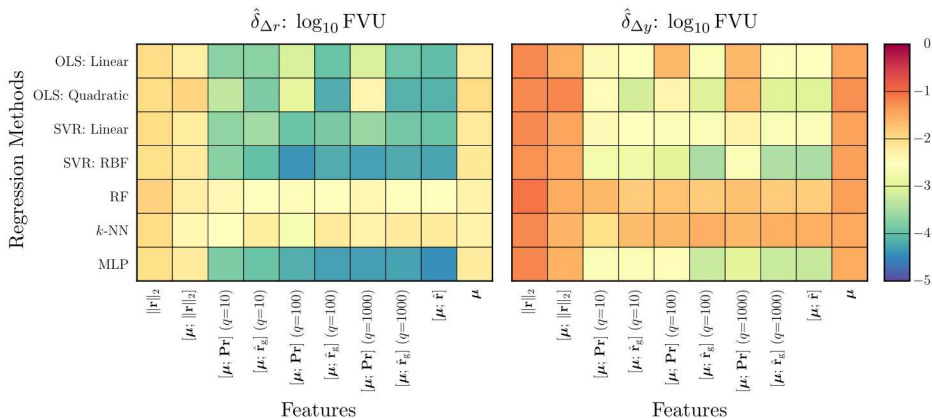


4: 99.77%



5: 99.90%

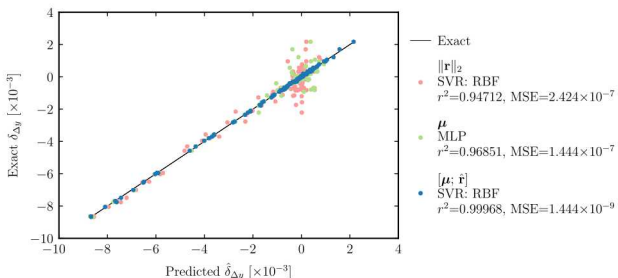
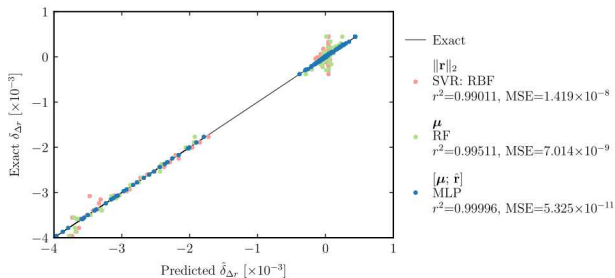
## PCAP: FVU for QoI Error Prediction



Fraction of variance unexplained (FVU) is  $1 - r^2$  ( $r^2$  is coefficient of determination)

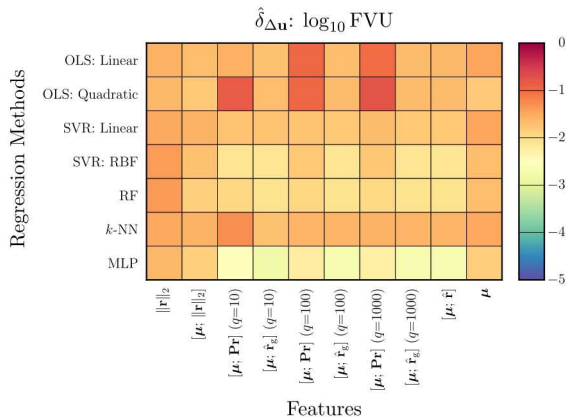
- SVR: RBF and MLP perform the best
- $[\mu; \hat{r}_g]$  and  $[\mu; \mathbf{Pr}]$  well with **only  $q = 100$  samples** (compared to  $N_{\mathbf{u}} = 278, 301$ )

## PCAP: QoI Error Predictions



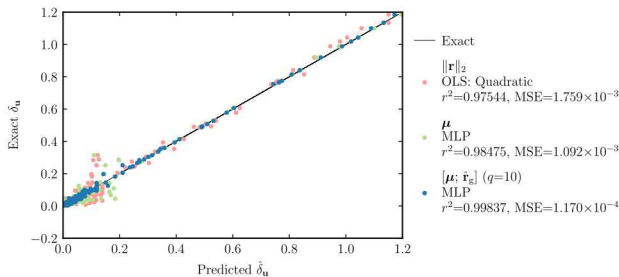
- Our methods beat previous state-of-the-art methods with  $r^2 > 0.9996$  in both cases

# PCAP: FVU for Normed State-Space Error Prediction



- MLP performs the best
- $[\boldsymbol{\mu}; \hat{\mathbf{r}}_g]$  and  $[\boldsymbol{\mu}; \mathbf{P}\mathbf{r}]$  perform well with **only  $q = 10$  samples** (compared to  $N_{\mathbf{u}} = 278, 301$ )

# PCAP: Normed State-Space Error Predictions



- Our methods beat previous state-of-the-art methods with  $r^2 > 0.998$

# Outline

- Introduction
- Parameterized Nonlinear Algebraic Equations
- Proposed Approach
- Numerical Experiments
- Summary
  - Feature Choices
  - Feature Reduction

# Feature Choices

- Norm of the residual,  $\|\mathbf{r}\|_2$ 
  - **Low**-quality **single** feature
  - **Expensive** to compute and performs **poorly**
- Dual-weighted residual,  $d$ 
  - **High**-quality **single** feature
  - Performs **well** for **small** amounts of training data
  - Very **expensive** to compute
- Parameters  $\mu$ 
  - **Only** perform well with SVR: RBF or MLP
  - Do **not** perform well with OLS: Linear
- Parameters and gappy principal components of residual,  $[\mu; \hat{\mathbf{r}}_g]$ 
  - Perform the best with  $r^2 > 0.998$  for each experiment
  - Only require about **13 features**

# Feature Reduction

- Gappy PCA more effective than directly sampling the residual
- Little benefit to using  $q \geq 100$  samples; more samples are more expensive and do not perform much better
- Often, only  $q = 10$  samples are necessary to get accurate prediction

# References

- B. Freno and K. Carlberg  
Machine-learning error models for approximate solutions to parameterized systems of nonlinear equations  
*In preparation*
- N. Alexandrov et al.  
Approximation and model management in aerodynamic optimization with variable-fidelity models  
*AIAA Journal of Aircraft* (2001)
- A. Buffa et al.  
A priori convergence of the greedy algorithm for the parametrized reduced basis method  
*ESAIM: Mathematical Modelling and Numerical Analysis* (2012)
- M. Drohmann and K. Carlberg  
The ROMES method for statistical modeling of reduced-order-model error  
*SIAM/ASA Journal on Uncertainty Quantification* (2015)
- M. S. Eldred et al.  
Second-order corrections for surrogate-based optimization with model hierarchies  
*AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference* (2004)
- S. E. Gano et al.  
Hybrid variable fidelity optimization by using a kriging-based scaling function  
*AIAA Journal* (2005)
- M. A. Grepl and A. T. Patera  
A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations  
*ESAIM: Mathematical Modelling and Numerical Analysis* (2005)
- G. Rozza et al.  
Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations  
*Archives of Computational Methods in Engineering* (2008)
- S. Trehan et al.  
Error modeling for surrogates of dynamical systems using machine learning  
*International Journal for Numerical Methods in Engineering* (2017)

# Questions?

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

This presentation describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the presentation do not necessarily represent the views of the U.S. Department of Energy or the United States Government.