Physics-Infused Differential-Algebraic Reduced-Order Models for Multi-Disciplinary Systems

Carlos Vargas Venagas and **Daning Huang**

LLNL Machine Learning for Industry Forum, August 10-12, 2021

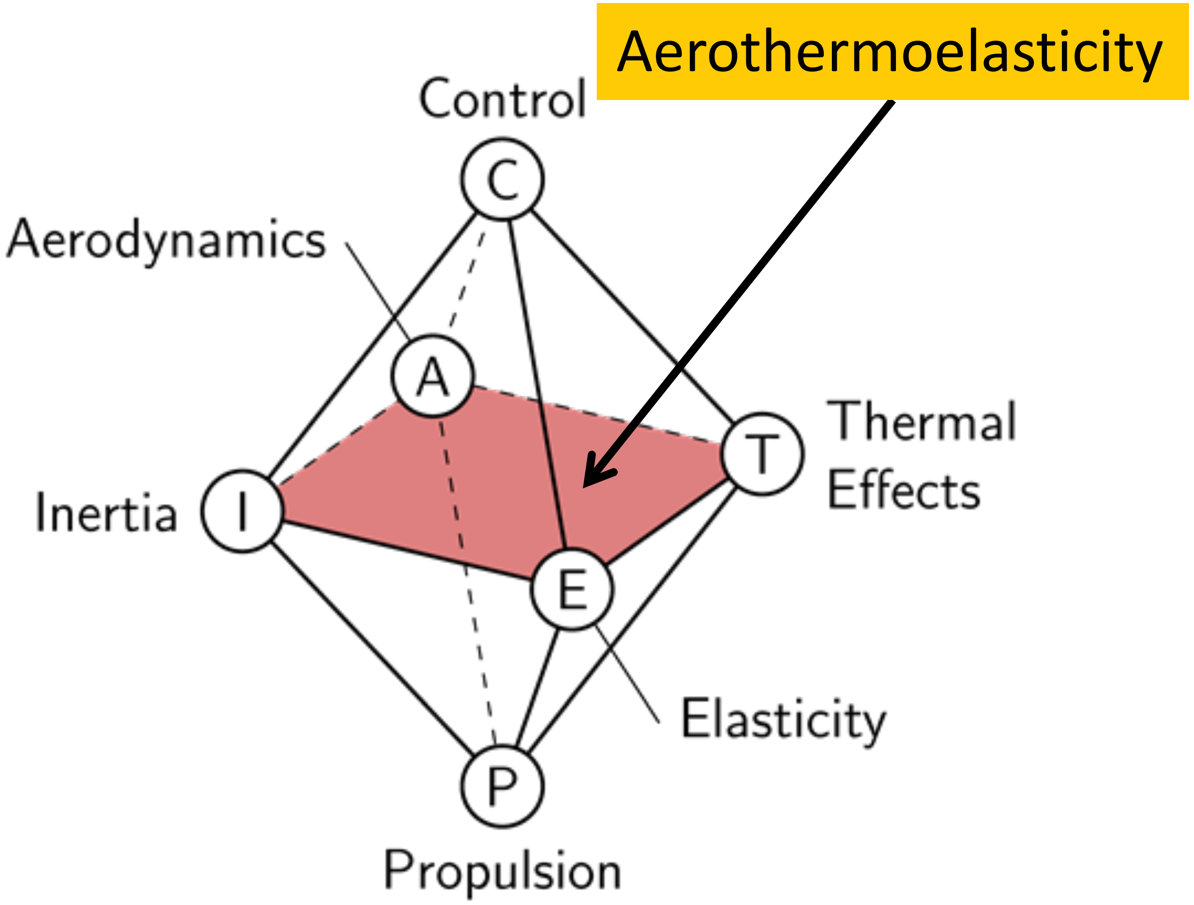
PennState
College of Engineering

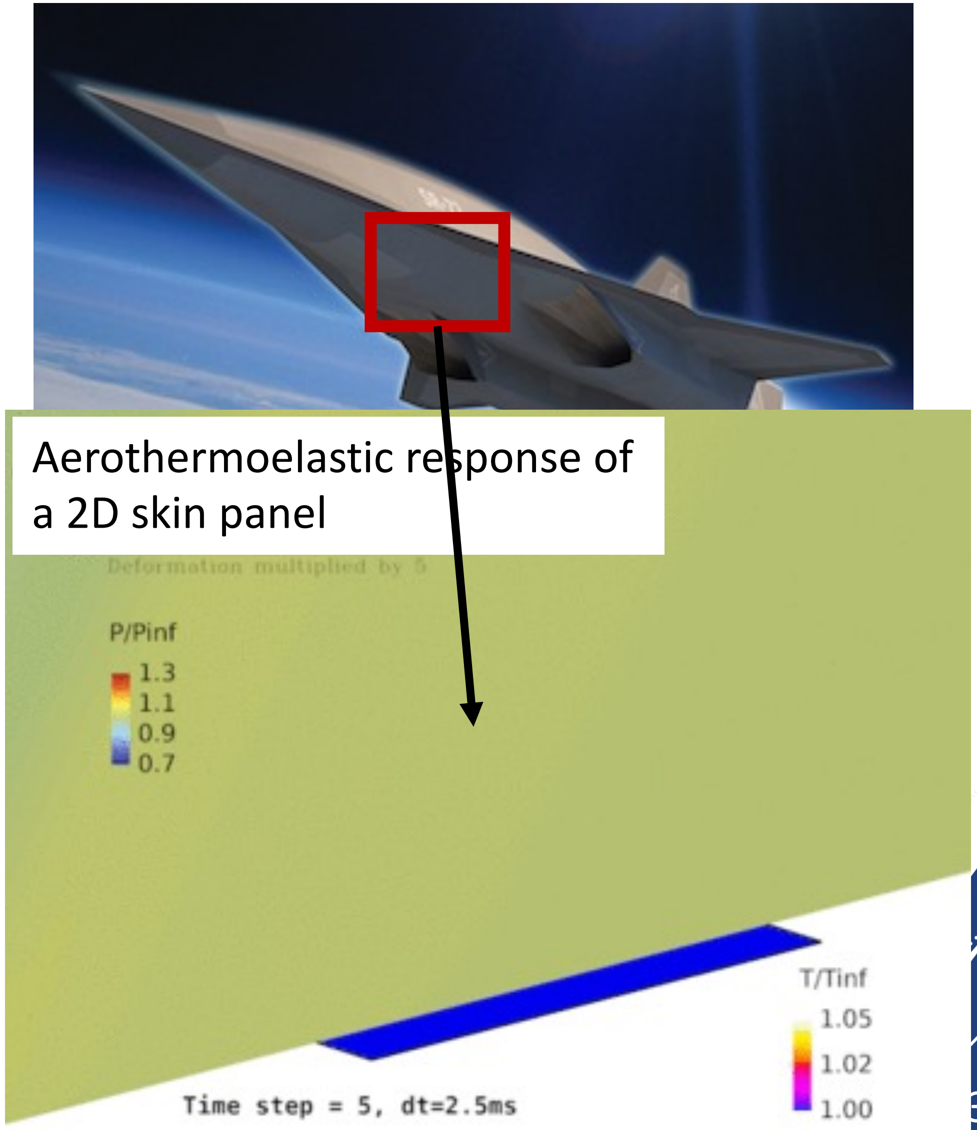
APUS Lab
Aerospace multi-Physical and Unconventional Systems

Background

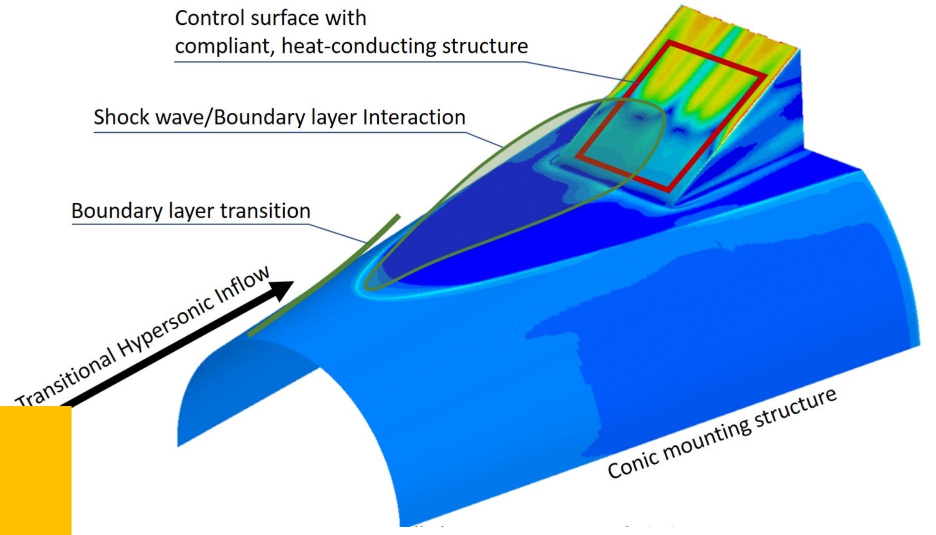
Hypersonic Aerothermoelasticity

Barrier to fly at Hypersonic speed





As a Multi-disciplinary system



- *Real gas effect*
- *Viscous interaction*
- *Compressible turbulence*

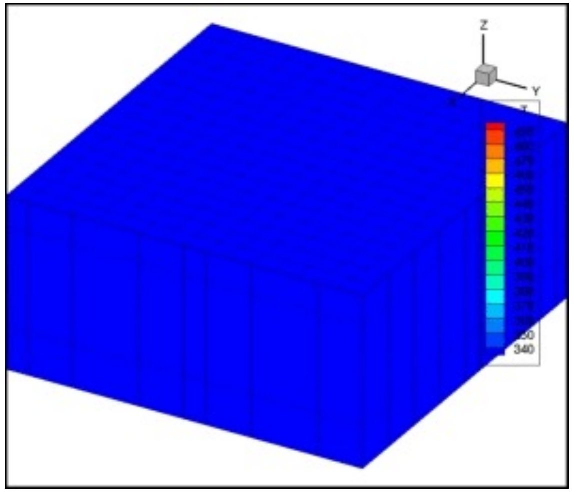
Hypersonic Aerothermodynamics

Temperature

Heat flux

Pressure

Deformation

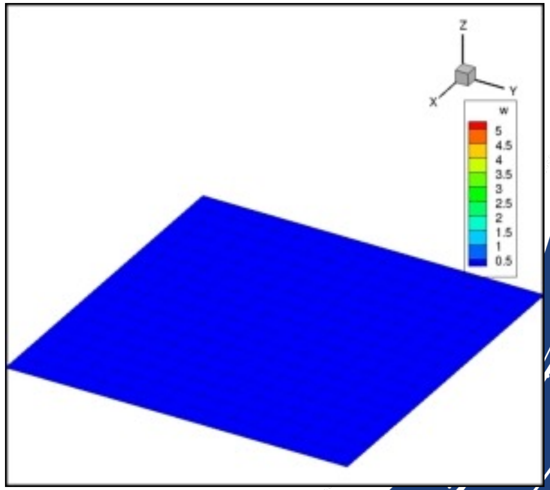


Heat Conduction

Temperature

Deformation

Structural Dynamics

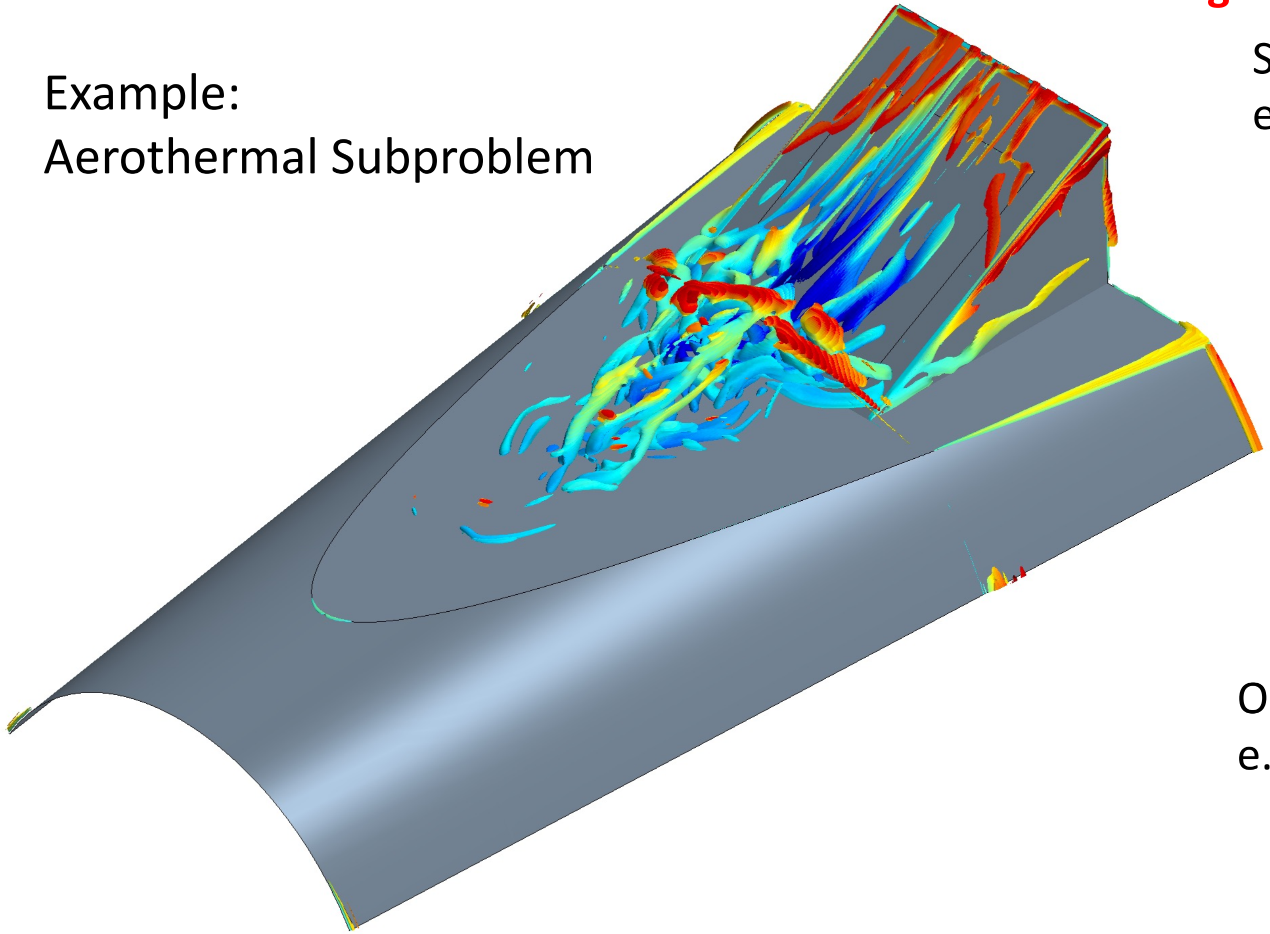


- *Thermal management*
- *Material degradation*
- *Charring and ablation*

- *Flutter and buckling*
- *Fatigue and creep*
- *Reliability assessment*

Challenge to analyze, optimize, control such systems…

Example:
Aerothermal Subproblem



High computational cost

States **>10^9**
e.g. flow states

Vast design space

Parameters **~10^3**
e.g. geometrical configuration

$$\dot{x} = f(\boxed{x}, u; \boxed{\mu})$$
$$\boxed{z} = h(x, \boxed{u}; \mu)$$

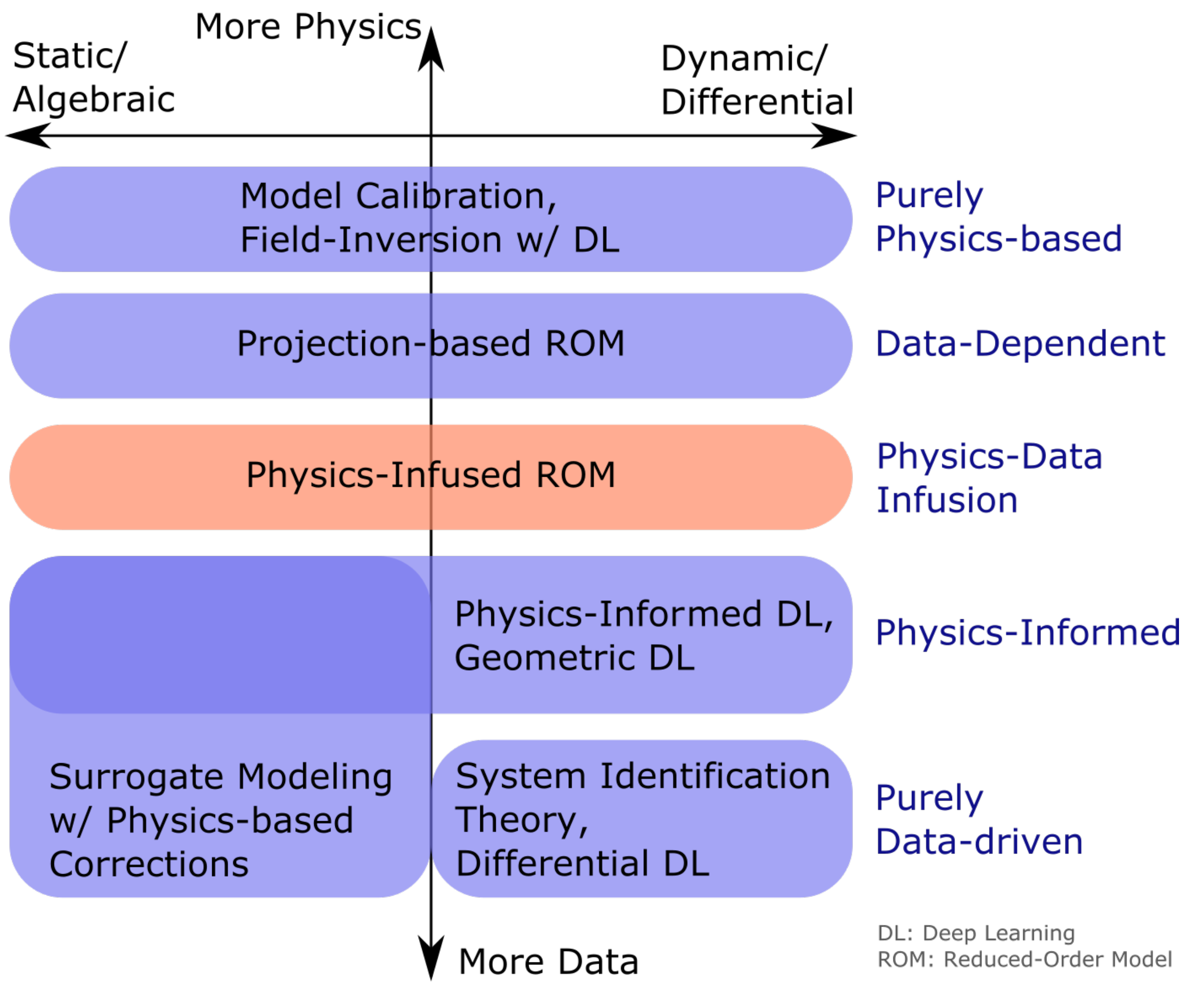
Output **~10^3**
e.g. aerothermal loads

Input **~10^3**
e.g. thermoelastic response,
control commands

High-dimensional optimal control laws & uncertainty quantification

What are the options?

Functional form of predictive model	Generalizability to system parameters	Computational cost
Physics (+ Data)	High	High
Physics	Mid	Mid
Physics + Data	High?	Low?
Data	Mid	Low
Data	Low	Low



DL: Deep Learning
ROM: Reduced-Order Model

Formulation

Physics-Infused Reduced-Order Modeling

Examples	Boundary layer	Slender structure
Full-order	Navier-Stokes Eqn.	Elasticity Eqns.
States	Density, velocity, energy	3D displacement field
Low-order	Momentum integral Eqn.	Euler-Bernoulli Eqn.
State variables	BL thicknesses	1D disp. field
Aux. variables	Shape factor, Skin friction	Bending stiffness

General Idea

- Full-Order Model (FOM)

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}; \boldsymbol{\mu})$$
$$\boldsymbol{z} = \boldsymbol{H}(\boldsymbol{x}, \boldsymbol{u}; \boldsymbol{\mu})$$

- Low-Order Model – *First principle, much less states*

$$0 = \boldsymbol{f}(\boldsymbol{y}, \dot{\boldsymbol{y}}, \boldsymbol{c}, \boldsymbol{u}; \boldsymbol{\mu})$$ ← Differential-algebraic Eqn.

$$\boldsymbol{c} = \boldsymbol{g}(\boldsymbol{y}, \boldsymbol{u}; \boldsymbol{\mu})$$ ← Auxiliary variables

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{y}, \boldsymbol{c}, \boldsymbol{u}; \boldsymbol{\mu})$$

- Physics-Infused Reduced-Order Model

$$0 = \boldsymbol{f}(\boldsymbol{y}, \dot{\boldsymbol{y}}, \boldsymbol{c}, \boldsymbol{u}; \boldsymbol{\mu})$$

$$\boldsymbol{A}\dot{\boldsymbol{c}} = \tilde{\boldsymbol{g}}(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{c}; \boldsymbol{\mu})$$ ← Augmented form

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{y}, \boldsymbol{c}, \boldsymbol{u}; \boldsymbol{\mu})$$

- DAE-constrained optimization

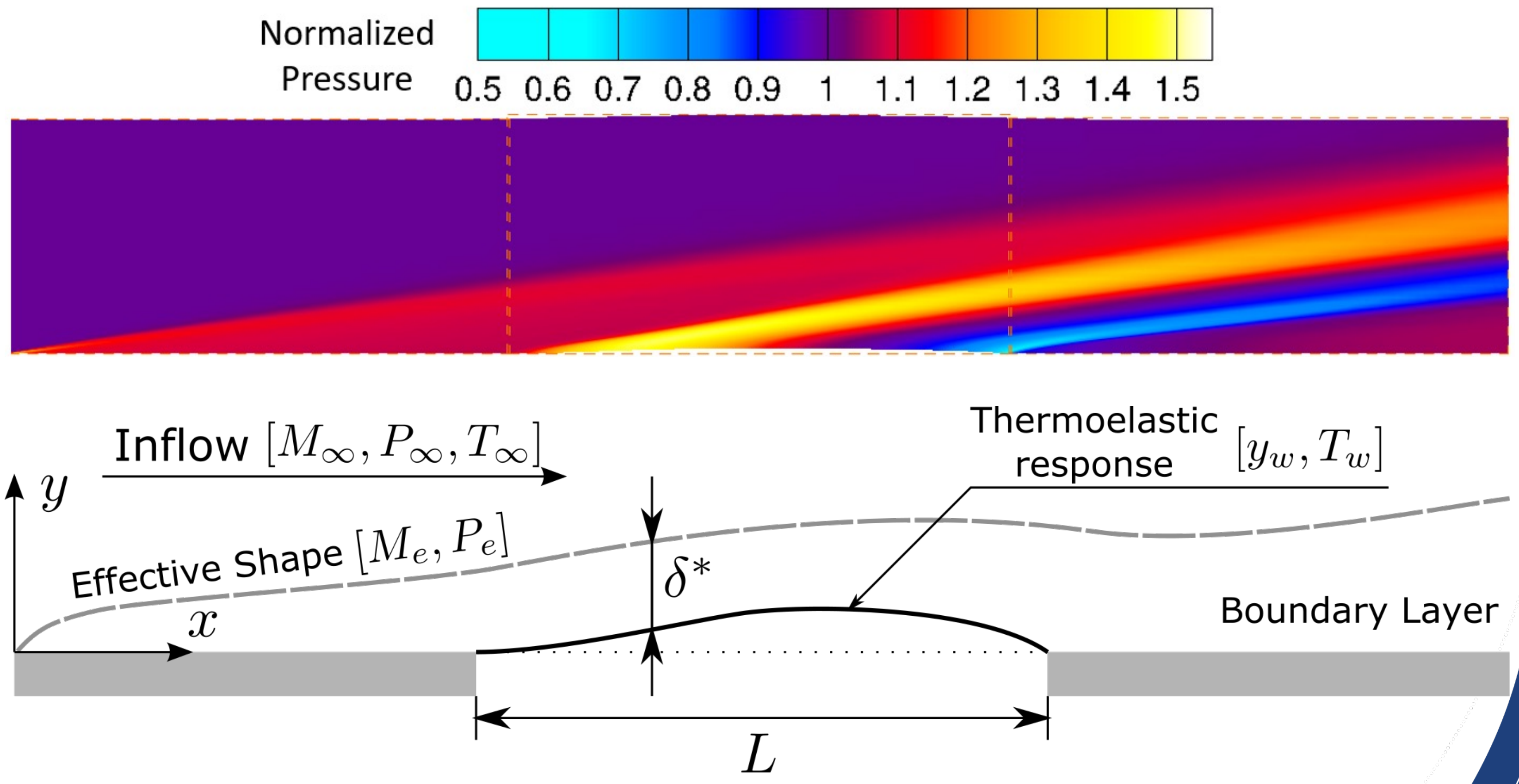
$$\boldsymbol{A}^*, \boldsymbol{\Theta}^* = \underset{\boldsymbol{A}, \boldsymbol{\Theta}}{\operatorname{argmin}} \|\boldsymbol{z}_{FOM} - \boldsymbol{z}\|$$
$$\text{s.t.} \quad 0 = \boldsymbol{f}(\boldsymbol{y}, \dot{\boldsymbol{y}}, \boldsymbol{c}, \boldsymbol{u}; \boldsymbol{\mu})$$
$$\boldsymbol{A}\dot{\boldsymbol{c}} = \tilde{\boldsymbol{g}}(\boldsymbol{y}, \boldsymbol{u}, \boldsymbol{c}; \boldsymbol{\mu}; \boldsymbol{\Theta})$$
$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{y}, \boldsymbol{c}, \boldsymbol{u}; \boldsymbol{\mu})$$

Back to Hypersonic aerothermodynamics

First-principle modeling: Turbulence Viscous-Inviscid Interaction (TVI)

$$\begin{cases} \dfrac{d}{dx}\left(\dfrac{\delta^*}{H}\right) + \left(H_i \dfrac{T_w}{T_o} - 4\right)\dfrac{\delta^*}{HM_e}\dfrac{dM_e}{dx} = \dfrac{C_f}{2} \\ p_e(x) = p_\infty\left(1 + \dfrac{\gamma-1}{2}M_\infty \dfrac{dy_e}{dx}\right)^{\frac{2\gamma}{\gamma-1}} \\ y_e = y_w + \delta^* \end{cases}$$

- Classical integral equations that respect physics
- A system of differential-algebraic equations (DAEs)
- But misses some physics, e.g. High-temperature effects



Casting to state-space form

First-principle modeling: Turbulence Viscous-Inviscid Interaction (TVI)

$$\mathbf{M}(\mathbf{y},\mathbf{c};\boldsymbol{\tau})\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}x} = \mathbf{f}(\mathbf{y},\mathbf{c},x;\mathbf{u},\boldsymbol{\tau})$$

$$\mathbf{c} = \mathbf{g}(\mathbf{y},x;\mathbf{u},\boldsymbol{\tau})$$

$$\mathbf{z} = \mathbf{h}(\mathbf{y},\mathbf{c},x;\mathbf{u},\boldsymbol{\tau})$$

- Classical integral equations that respect physics
- A system of differential-algebraic equations (DAEs)
- But misses some physics, e.g. High-temperature effects

$$\mathbf{M}(\mathbf{y},\mathbf{c};\boldsymbol{\tau}) = \begin{bmatrix} H & \alpha\frac{H\delta^*}{M_e} - 2\delta^* M_e \xi & 0 \\ M_e & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{f}(\mathbf{y},\mathbf{c},x;\mathbf{u},\boldsymbol{\tau}) = \begin{bmatrix} H^2\frac{C_f}{2} \\ \alpha - \frac{dy_w}{dx} \\ P_e - P_o\left(1 + \frac{\gamma-1}{2}M_e^2\right)^{\frac{-\gamma}{\gamma-1}} \end{bmatrix}$$

$$\mathbf{g}(\mathbf{y},x;\mathbf{u},\boldsymbol{\tau}) = \begin{bmatrix} \frac{\gamma-1}{2}M_e(x)^2\left(1 + H_i\frac{T_w(x)}{T_o}\right) \\ 0.026\frac{T_e}{T^*}\left(\frac{\rho_\infty\mu^*}{\rho_e\mu_e}\right)^{1/4} Re_\theta^{-1/4} \\ 1 \end{bmatrix}$$

Aux. variables:

- Shape factor
- Skin friction coefficient
- Pressure ratio

$$\mathbf{h}(\mathbf{y},\mathbf{c},x;\mathbf{u},\boldsymbol{\tau}) = \begin{bmatrix} k_p P_e \\ \frac{C_f}{2Pr^{2/3}}\rho u_e\left[h_{aw}(T_r) - h(T_w)\right] \end{bmatrix}$$

Output variables:

- Surface pressure
- Surface heat flux

Creating the PIRO model

First-principle modeling: Turbulence Viscous-Inviscid Interaction (TVI)

$$\mathbf{M}(\mathbf{y},\mathbf{c};\boldsymbol{\tau})\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}x} = \mathbf{f}(\mathbf{y},\mathbf{c},x;\mathbf{u},\boldsymbol{\tau})$$

$$\mathbf{c} = \mathbf{g}(\mathbf{y},x;\mathbf{u},\boldsymbol{\tau})$$

$$\mathbf{z} = \mathbf{h}(\mathbf{y},\mathbf{c},x;\mathbf{u},\boldsymbol{\tau})$$

- Classical integral equations that respect physics
- A system of differential-algebraic equations (DAEs)
- But misses some physics, e.g. High-temperature effects

Model augmentation by functional correction

$$\mathbf{M}(\mathbf{y},\mathbf{c};\boldsymbol{\tau})\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}x} = \mathbf{f}(\mathbf{y},\mathbf{c},x;\mathbf{u},\boldsymbol{\tau})$$

$$\mathbf{c} = \boxed{\boldsymbol{\beta}(\mathbf{y},x;\mathbf{u},\boldsymbol{\tau})} \odot \mathbf{g}(\mathbf{y},x;\mathbf{u},\boldsymbol{\tau})$$

$$\mathbf{z} = \mathbf{h}(\mathbf{y},\mathbf{c},x;\mathbf{u},\boldsymbol{\tau})$$

- To account for missing physics -
- Taking an algebraic multiplicative form
- A new DAE with unknown functions

Learn unknown functionals from data

$$\mathbf{B}^i = \arg\min_{\mathbf{B}} \quad J(\mathbf{z}^i_{RANS}, \mathbf{z}(\mathbf{B}))$$

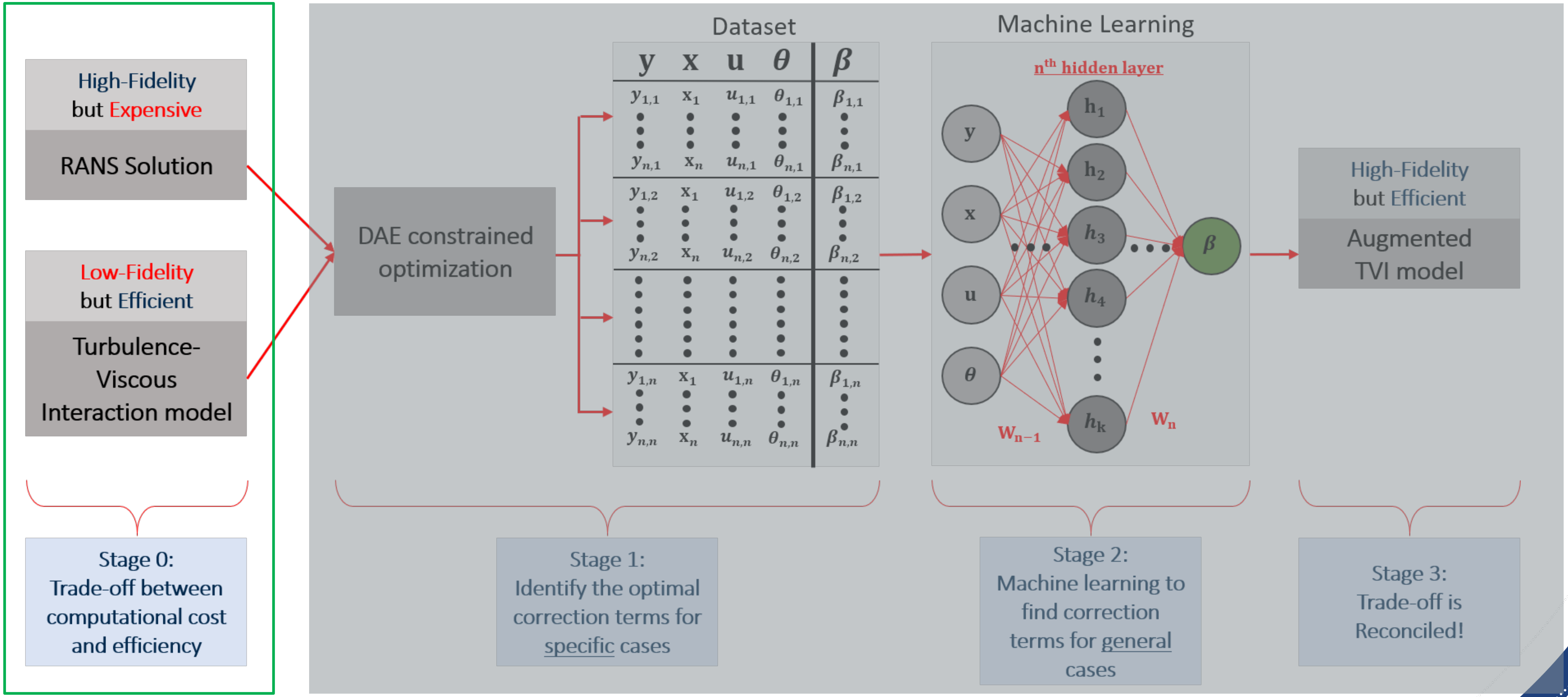
$$s.t. \quad \mathbf{M}(\mathbf{y},\mathbf{c};\boldsymbol{\tau})\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}x} = \mathbf{f}(\mathbf{y},\mathbf{c},x;\mathbf{u},\boldsymbol{\tau})$$

$$\mathbf{c} = \boxed{\boldsymbol{\beta}_{spl}(x;\mathbf{B})} \odot \mathbf{g}(\mathbf{y},x;\mathbf{u},\boldsymbol{\tau})$$

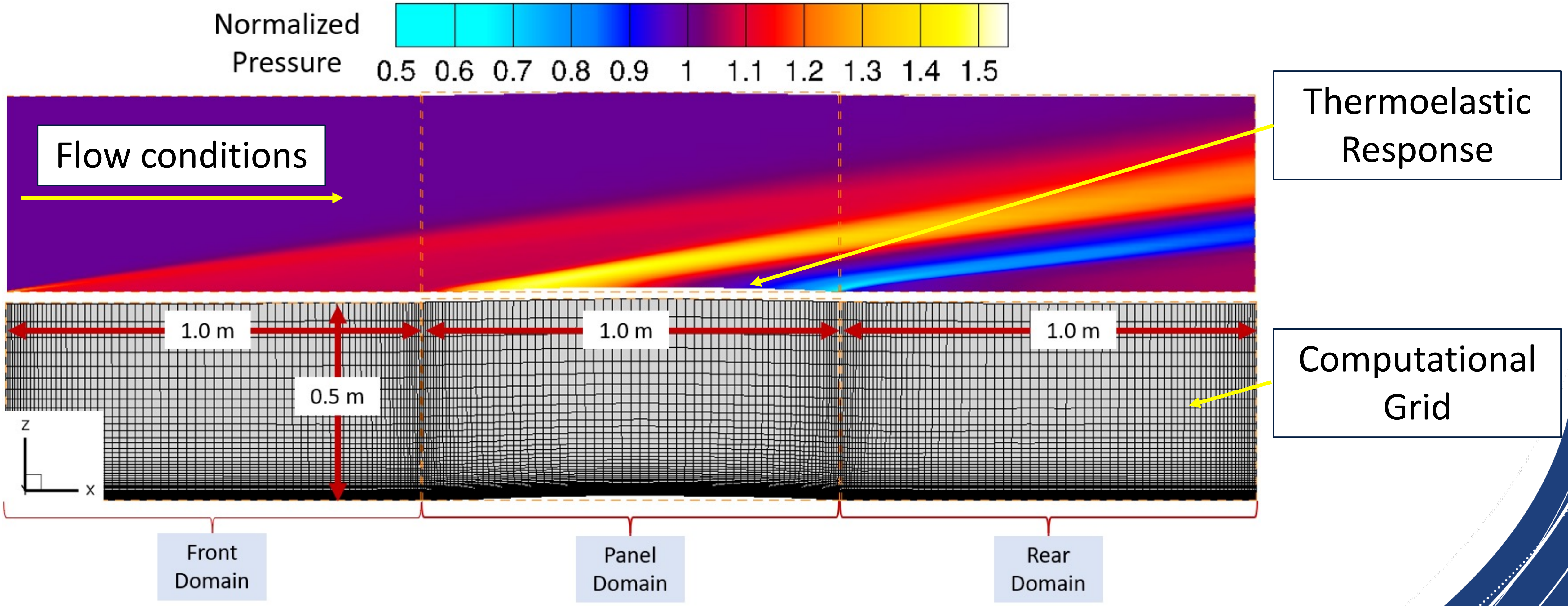
$$\mathbf{z} = \mathbf{h}(\mathbf{y},\mathbf{c},x;\mathbf{u},\boldsymbol{\tau})$$

- Learn corrections by a DAE-constrained optimization
- Works for computational (RANS/LES/DNS) or experimental data
- Captures more physics and is interpretable!

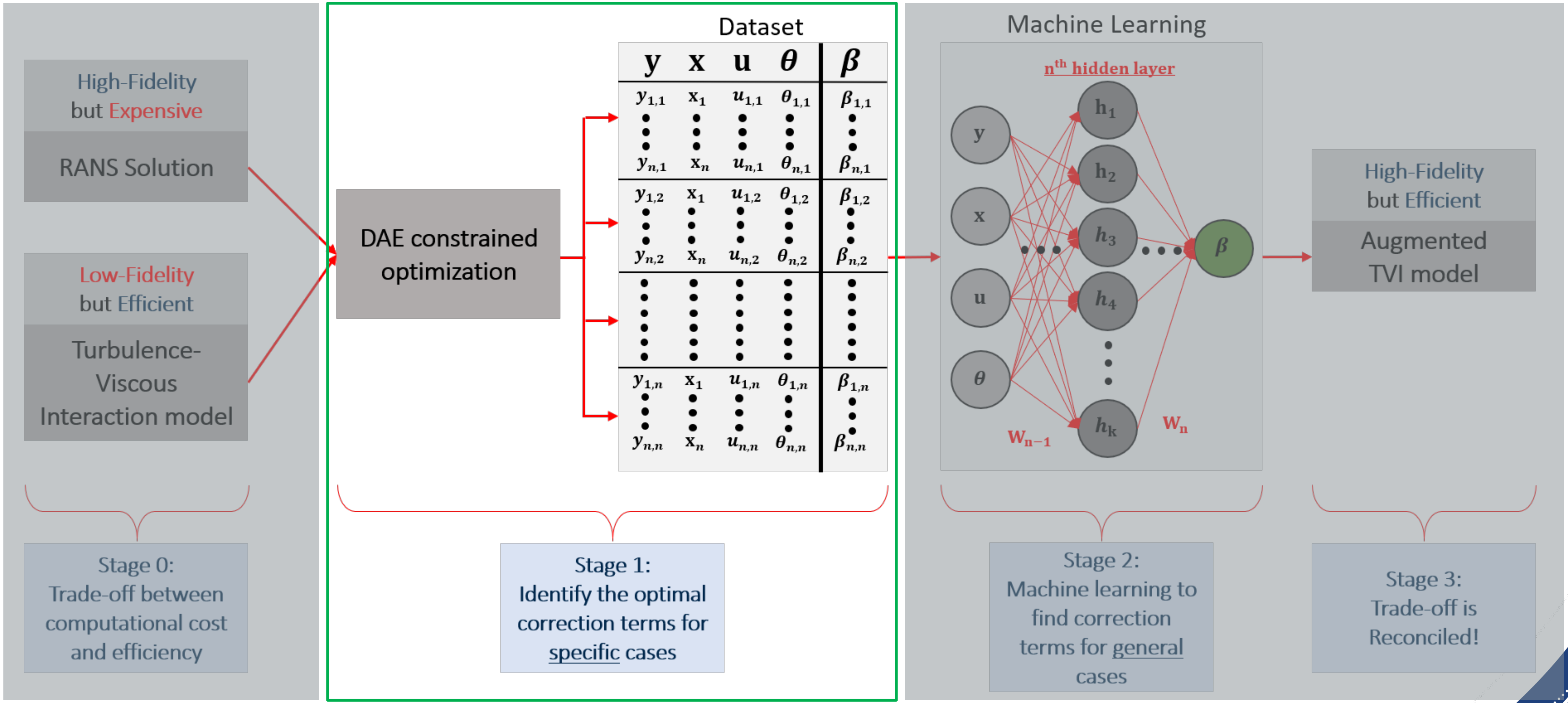
Methodology Overview



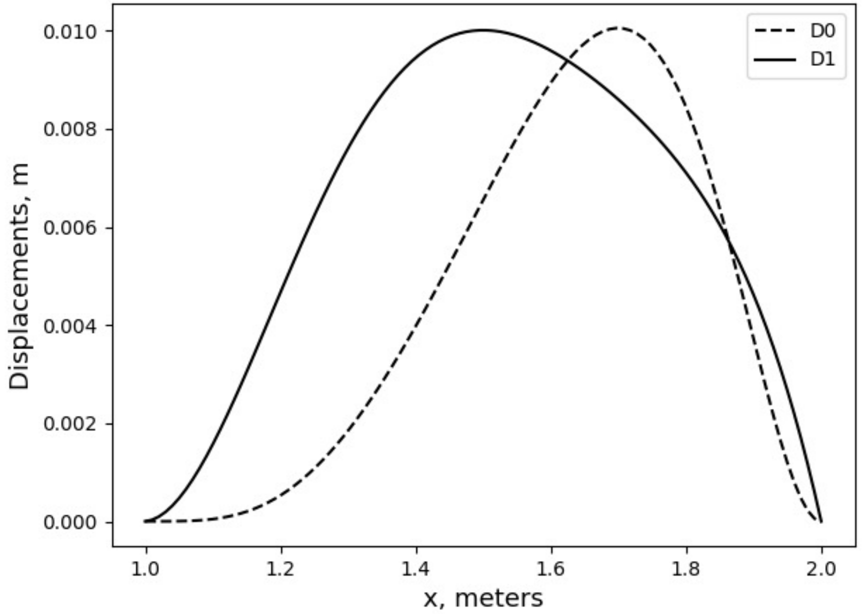
Stage 0: RANS Solutions

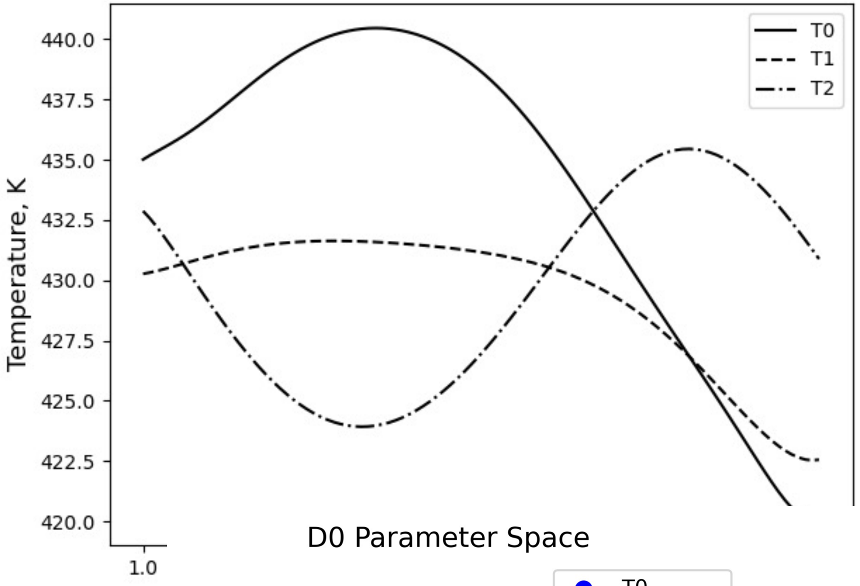


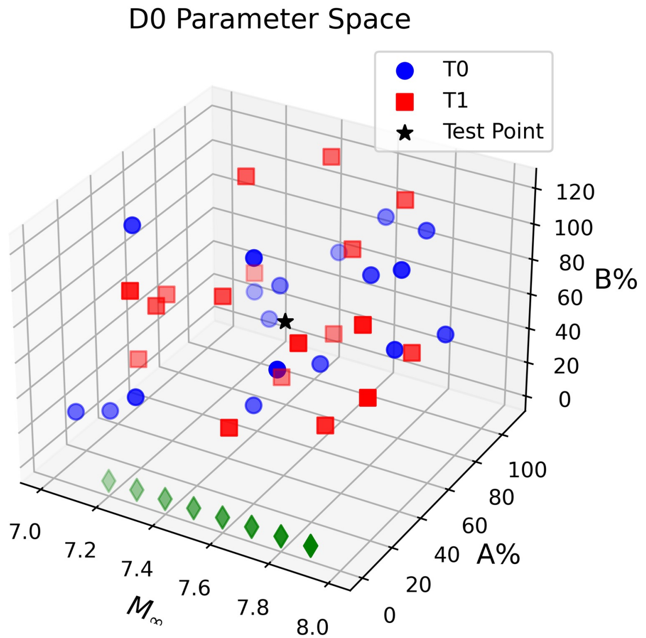
Stage 1: Solving Inverse Problems

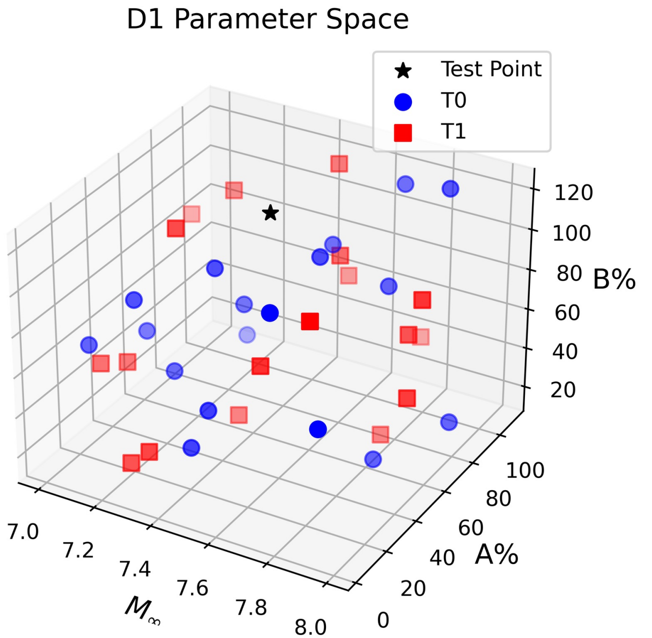


Stage 1, 1/2: Sampling inputs & parameters









Deformation: $y_w^i(x) = A y_{w_{Di}}(x)$

Wall temperature: $T_w^i(x) = T_{ref} + B T_d^i(x)$

APUS Lab

Aerospace multi-**P**hysical and **U**nconventional **S**ystem

Stage 1, 2/2: DAE-Constrained Optimization

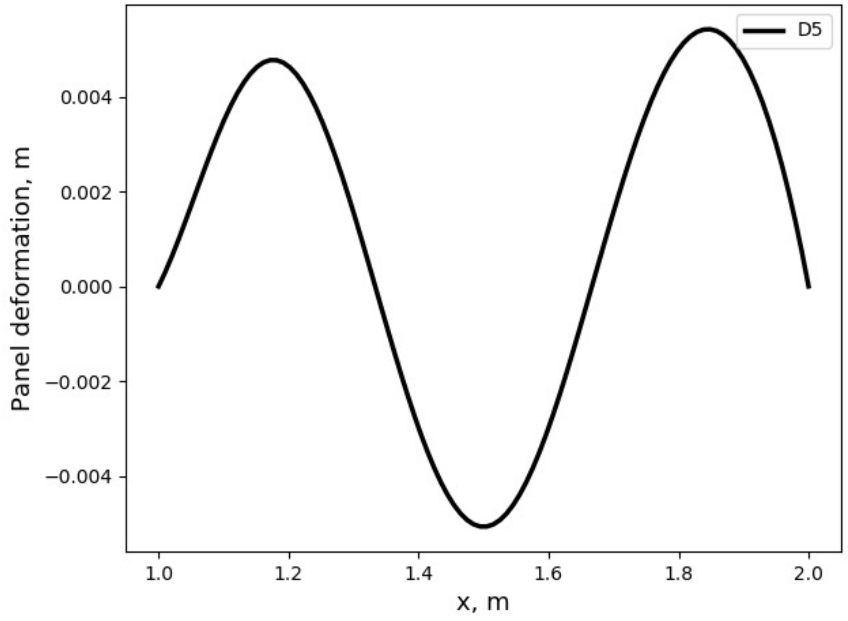
$$\mathbf{B}^i = \arg\min_{\mathbf{B}} \quad J(\mathbf{z}^i_{RANS}, \mathbf{z}(\mathbf{B}))$$

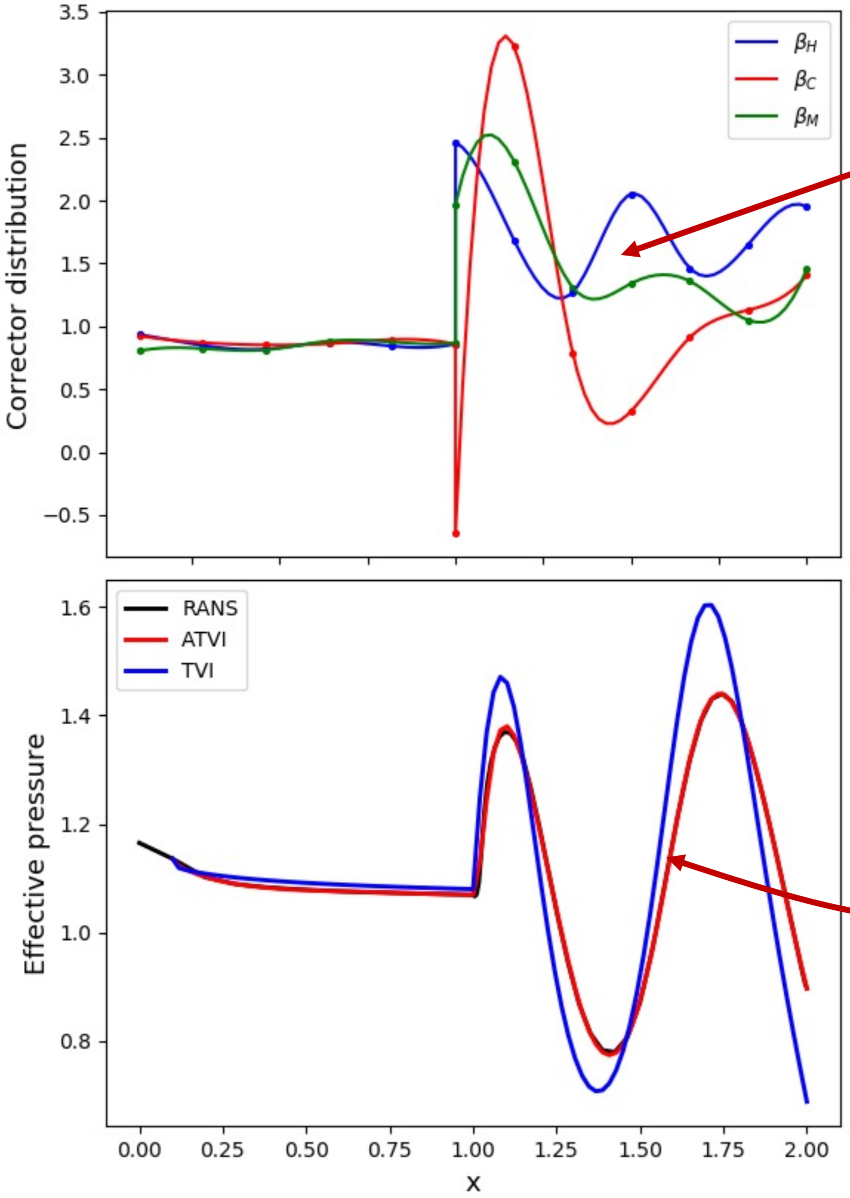
$$s.t. \quad \mathbf{M}(\mathbf{y}, \mathbf{c}; \boldsymbol{\tau})\frac{d\mathbf{y}}{dx} = \mathbf{f}(\mathbf{y}, \mathbf{c}, x; \mathbf{u}, \boldsymbol{\tau})$$

$$\mathbf{c} = \boldsymbol{\beta}_{spl}(x; \mathbf{B}) \odot \mathbf{g}(\mathbf{y}, x; \mathbf{u}, \boldsymbol{\tau})$$

$$\mathbf{z} = \mathbf{h}(\mathbf{y}, \mathbf{c}, x; \mathbf{u}, \boldsymbol{\tau})$$

Example: M76D5-100%

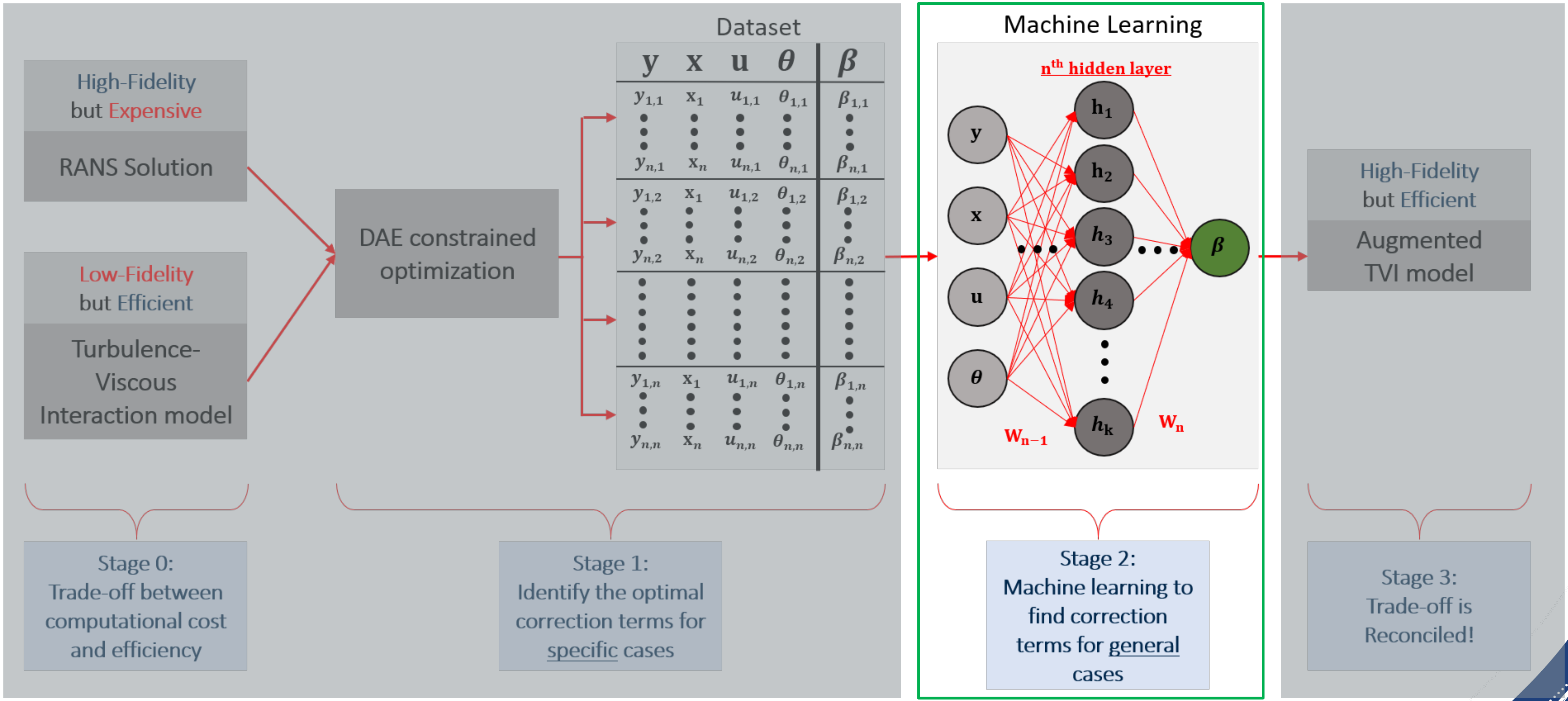




Training Dataset

$\mathbf{y}$	$\mathbf{x}$	$\mathbf{u}$	$\boldsymbol{\theta}$	$\boldsymbol{\beta}$
$y_{1,1}$	x_1	$u_{1,1}$	$\theta_{1,1}$	$\beta_{1,1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$y_{n,1}$	x_n	$u_{n,1}$	$\theta_{n,1}$	$\beta_{n,1}$
$y_{1,2}$	x_1	$u_{1,2}$	$\theta_{1,2}$	$\beta_{1,2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$y_{n,2}$	x_n	$u_{n,2}$	$\theta_{n,2}$	$\beta_{n,2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$y_{1,n}$	x_1	$u_{1,n}$	$\theta_{1,n}$	$\beta_{1,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$y_{n,n}$	x_n	$u_{n,n}$	$\theta_{n,n}$	$\beta_{n,n}$

Stage 2: Functional Representation of Correctors



APUS Lab
Aerospace multi-Physical and Unconventional Systems

PennState
College of Engineering

Stage 2: Machine Learning

Training Dataset

$\boldsymbol{y}$	$\mathbf{x}$	$\mathbf{u}$	$\boldsymbol{\theta}$	$\boldsymbol{\beta}$
$y_{1,1}$	x_1	$u_{1,1}$	$\theta_{1,1}$	$\beta_{1,1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$y_{n,1}$	x_n	$u_{n,1}$	$\theta_{n,1}$	$\beta_{n,1}$
$y_{1,2}$	x_1	$u_{1,2}$	$\theta_{1,2}$	$\beta_{1,2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$y_{n,2}$	x_n	$u_{n,2}$	$\theta_{n,2}$	$\beta_{n,2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$y_{1,n}$	x_1	$u_{1,n}$	$\theta_{1,n}$	$\beta_{1,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$y_{n,n}$	x_n	$u_{n,n}$	$\theta_{n,n}$	$\beta_{n,n}$

Gaussian Process Regression (GPR)

$\mathbf{x}^*$: new input, $\mathbf{x}$: training data point

$$\boldsymbol{y} \sim \mathbf{GP}\left(\mathrm{m}(\mathbf{x}), \mathrm{k}(\mathbf{x}, \mathbf{x}')\right)$$

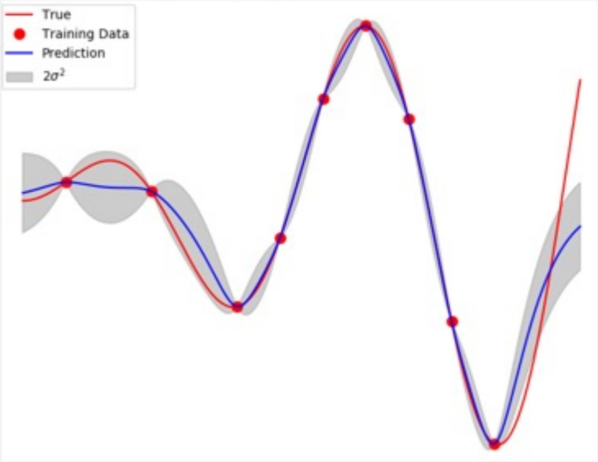
$$\boldsymbol{y_{GPR}} = \boldsymbol{N}\left(\boldsymbol{\mu}(\mathbf{x}^*), \boldsymbol{\sigma}^2(\mathbf{x}^*)\right)$$

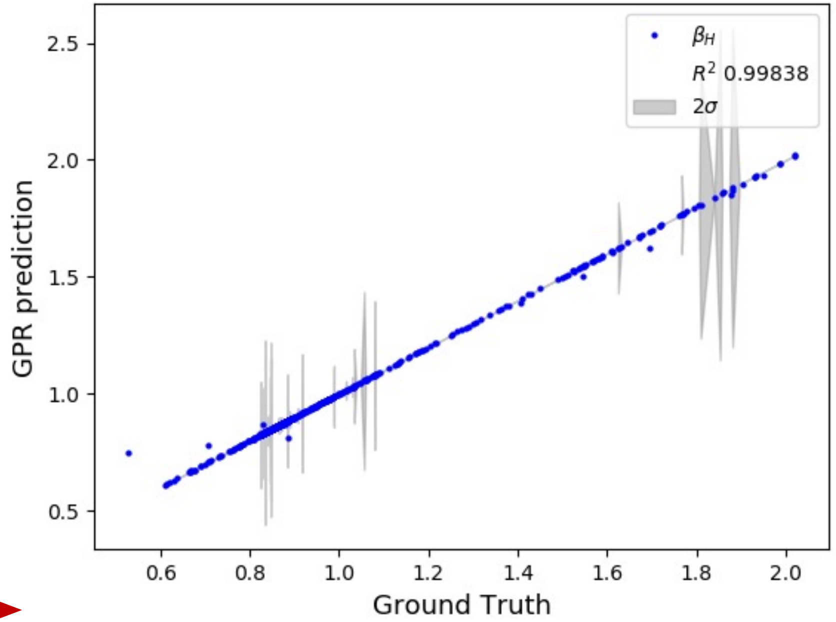
$\rightarrow$ ***Gaussian distribution***

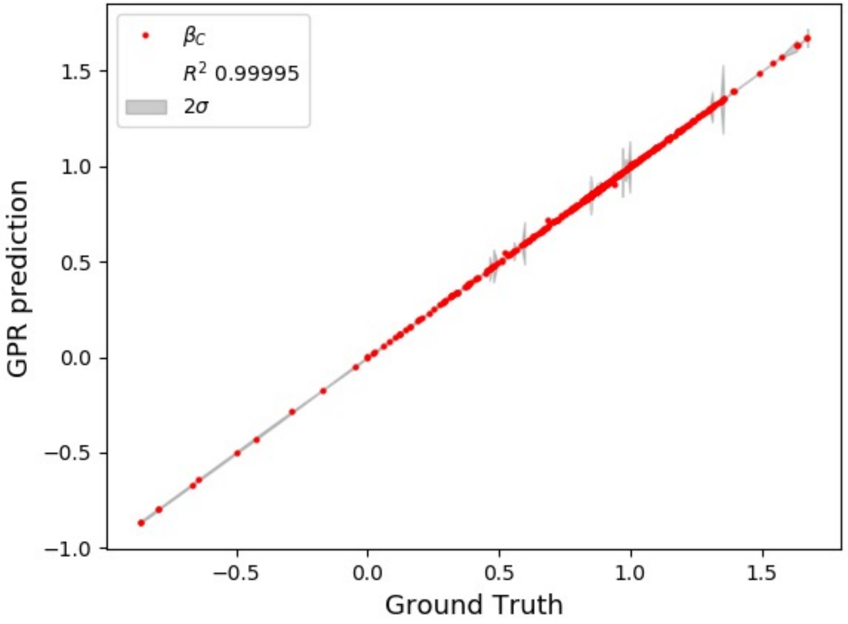
$$\boldsymbol{\mu}(\mathbf{x}^*) = \mathrm{k}(\mathbf{x}^*)^{\mathbf{T}}\mathbf{K}^{-1}\mathbf{y}$$

$$\boldsymbol{\sigma}^2(\mathbf{x}^*) = \mathrm{k}(\mathbf{x}^*, \mathbf{x}) - \mathrm{k}(\mathbf{x}^*)^{T}\mathbf{K}^{-1}\mathbf{k}(\mathbf{x}^*)$$

$$\boldsymbol{\beta} = \boldsymbol{\beta}(\boldsymbol{y}, x, \boldsymbol{u}, \boldsymbol{\theta})$$

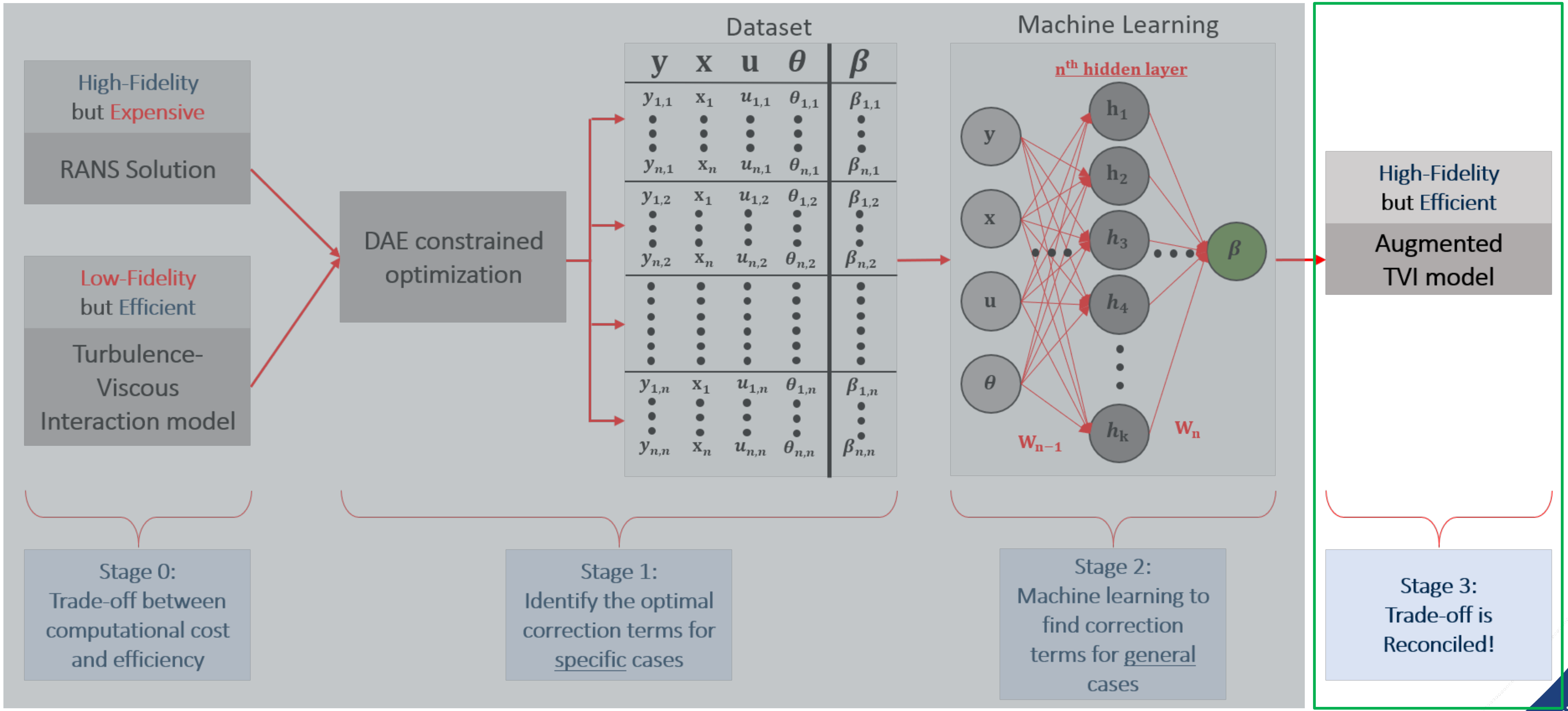






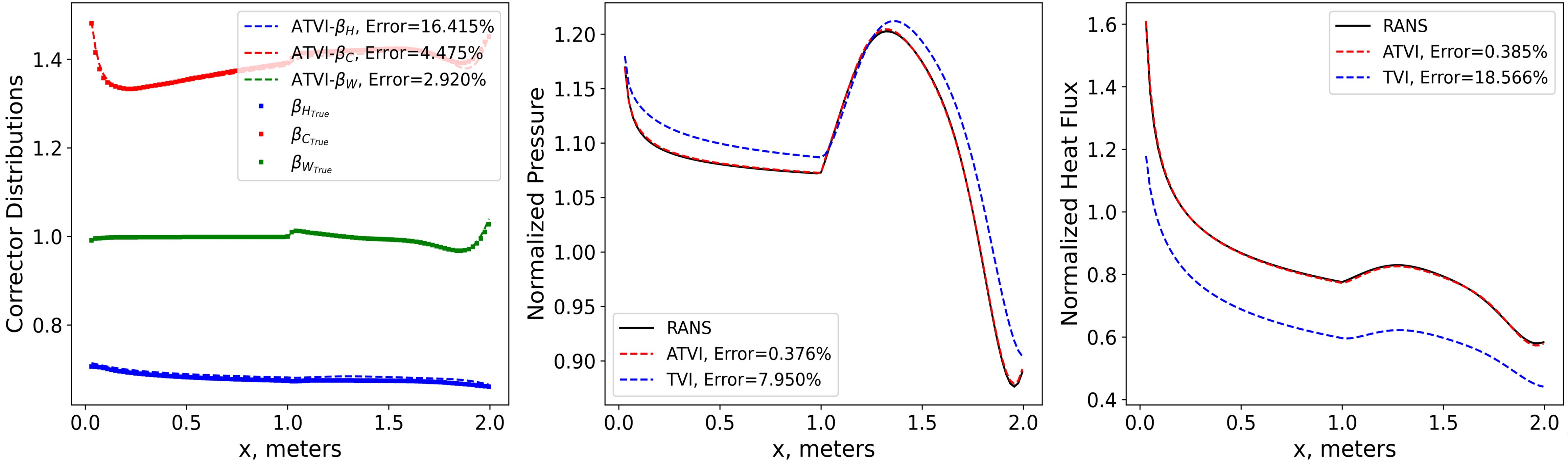
Going fancier, we could use tools like <u>symbolic regression</u> to get analytical expressions for the correction terms!

Stage 3: Reconciling Trade-off



Demo: New response, New flow conditions

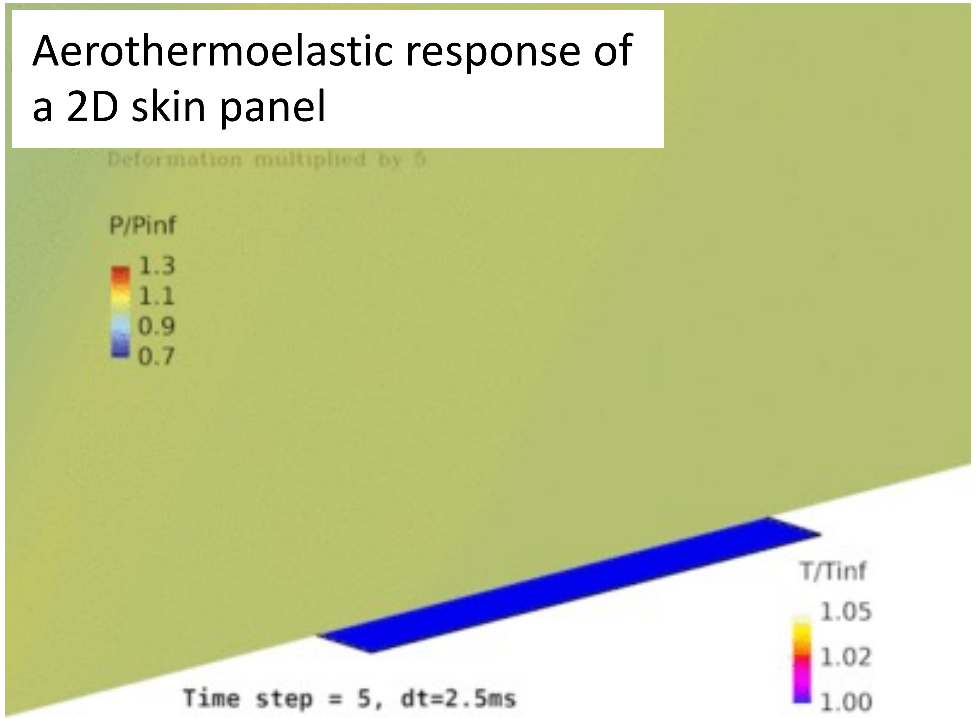
Displacement: $y_w(x) = 0.6[y_w^0(x) + y_w^1(x)]/2$
Wall temperature: $T_w(x) = T_{ref} + 0.7[T_w^0(x) + T_w^1(x)]/2$
Mach number: $M = 7.5$



Application

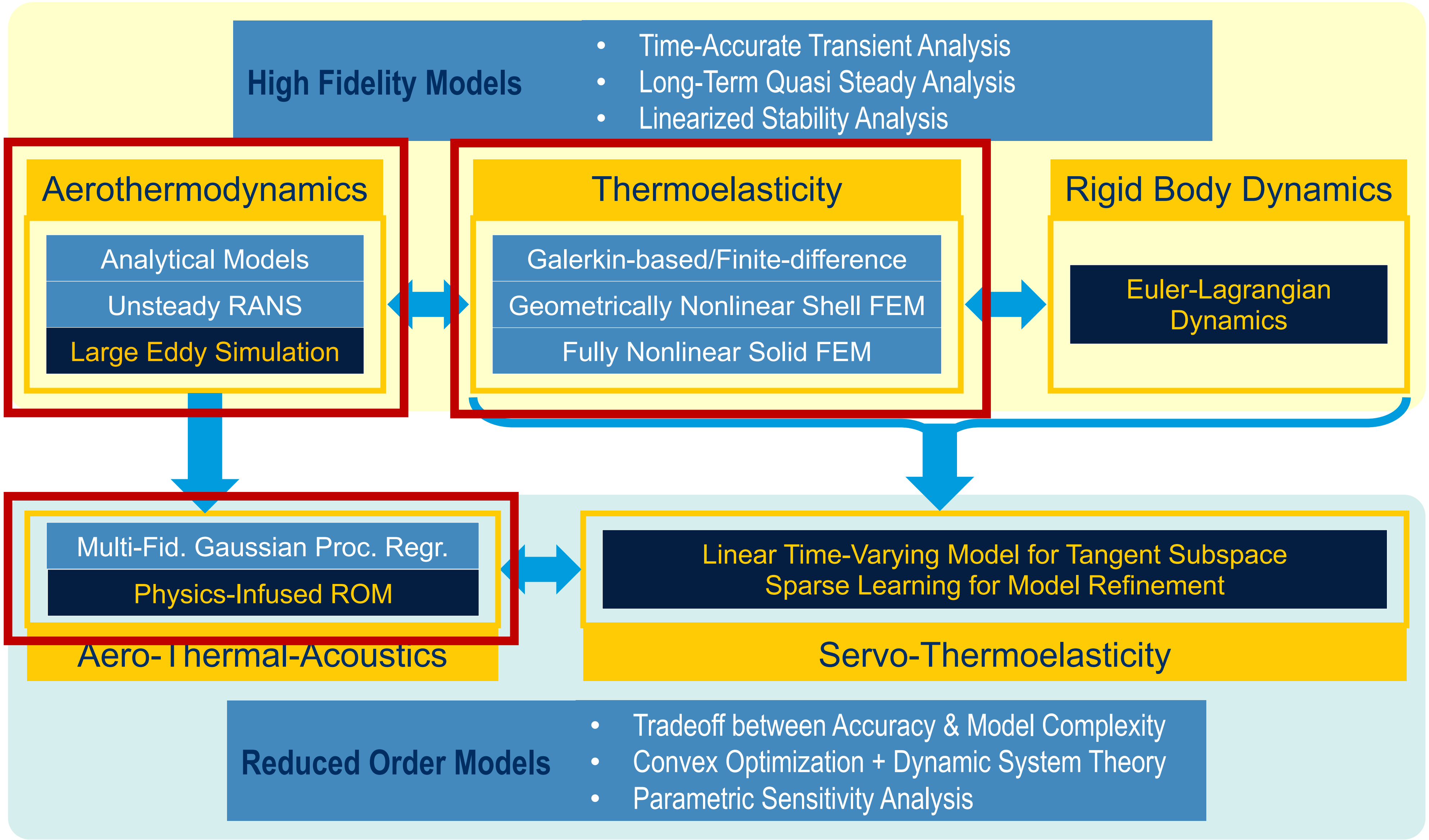
Back to Hypersonic Aerothermoelasticity

Benchmark case for aerothermoelasticity



Name	M_∞	P_∞ (Pa)	T_∞ (K)	Leading edge BC	Trailing edge BC
M7.523CP	7.523	3759.678	466.200	Clamped	Pinned
M7.750CC	7.750	3802.521	452.500	Clamped	Clamped
M7.400SS	7.400	3473.935	390.942	Pinned	Pinned
M7.523CX	7.523	3759.678	466.200	Clamped	Spring
M7.400XS	7.400	3473.935	390.942	Spring	Pinned

HYPATE-X: **HYP**ersonic **A**ero**T**hermo**E**lastic e**X**tended



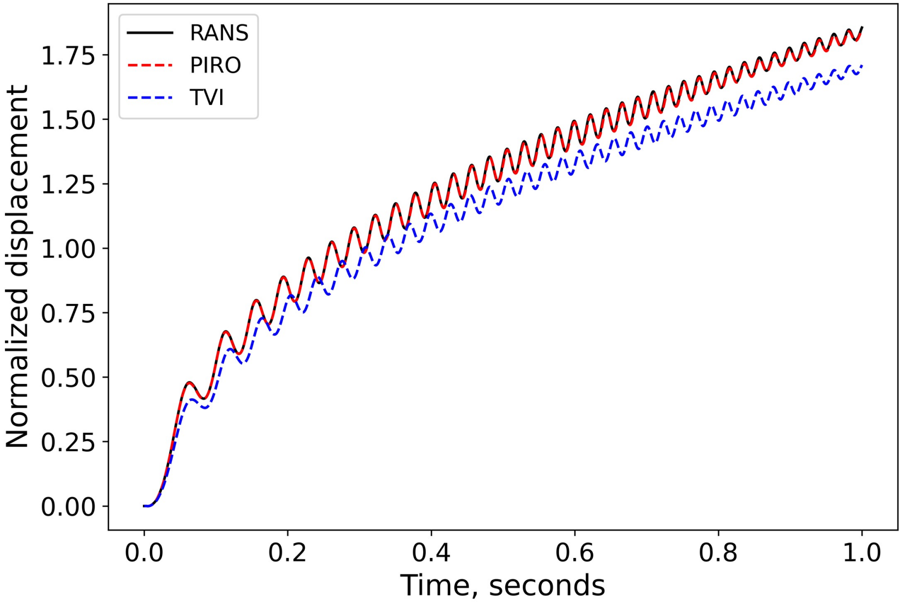
Collaborators:

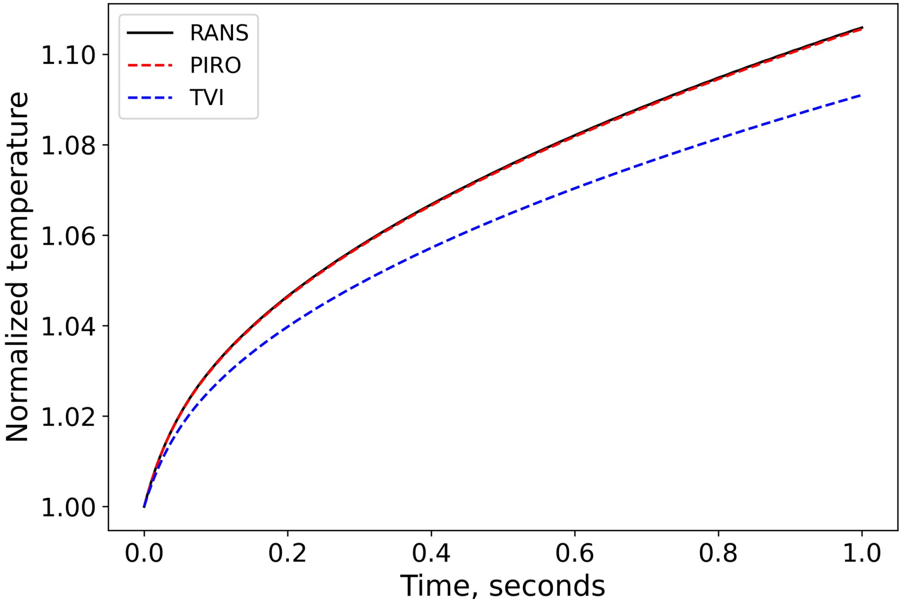
- Drs. P.P. Friedmann and T. Rokita (UMich)
- Drs. P. Singla and X.I.A. Yang (PSU)
- Dr. K.M. Hanquist (UofAz)

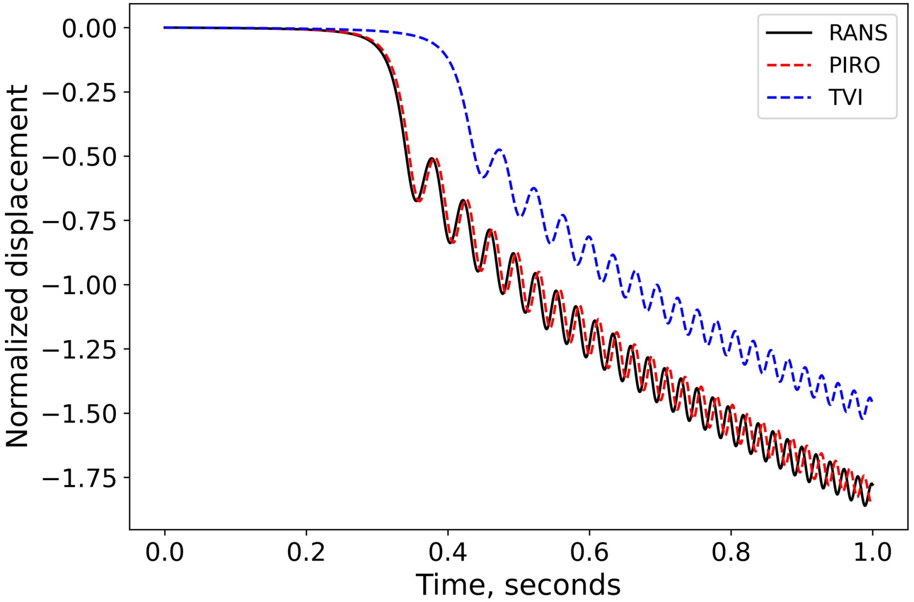
Existing modules

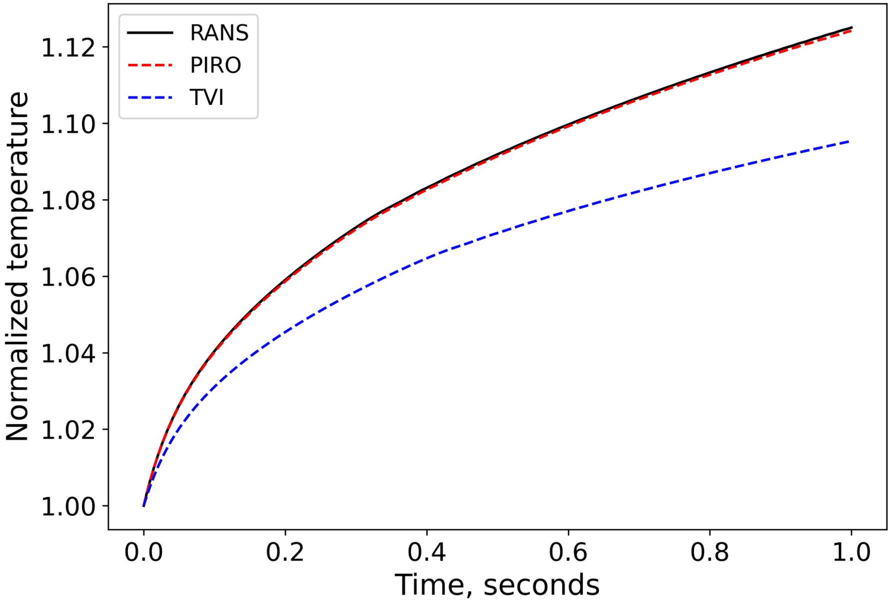
Developing modules

Accuracy of RANS at cost of milli-secs

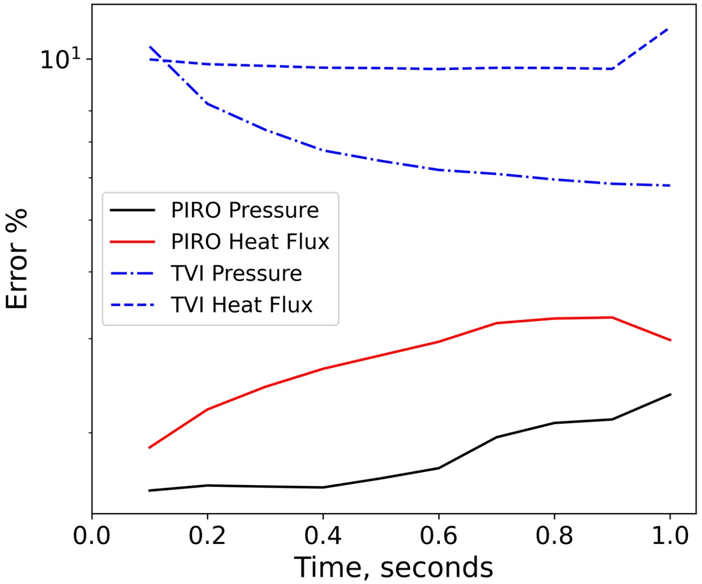


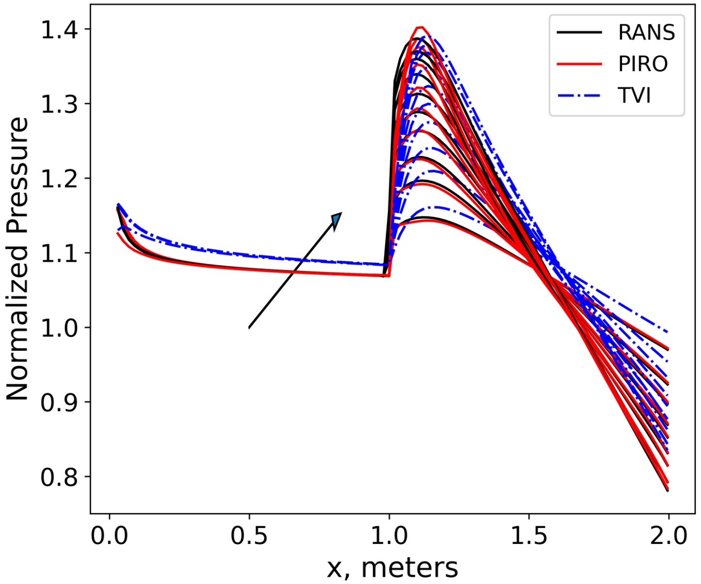


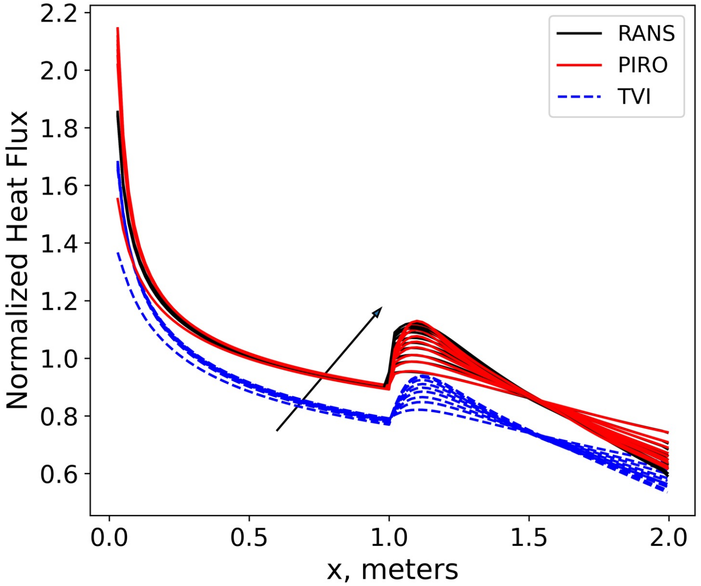


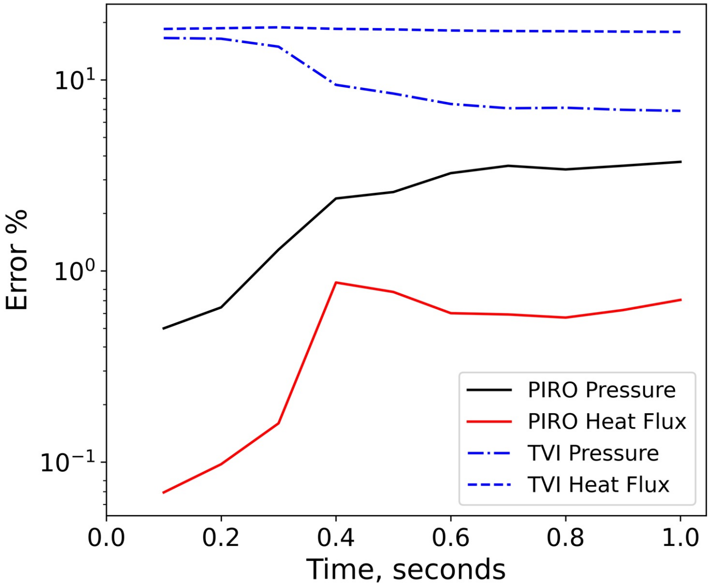


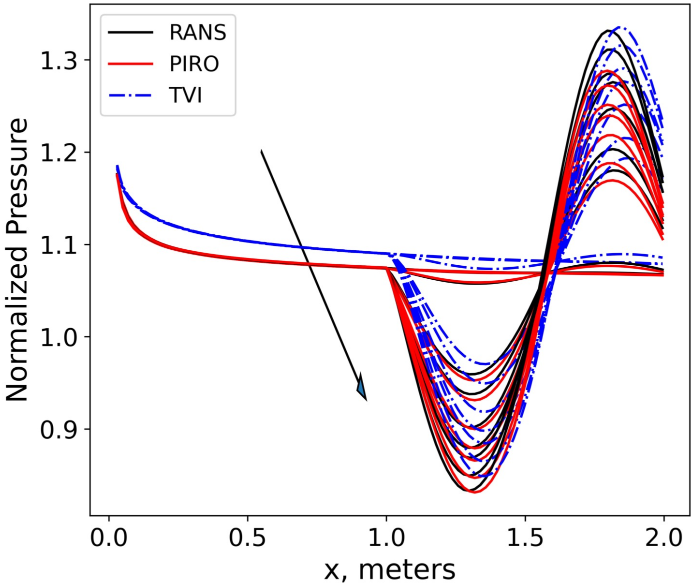
A closer look at the responses

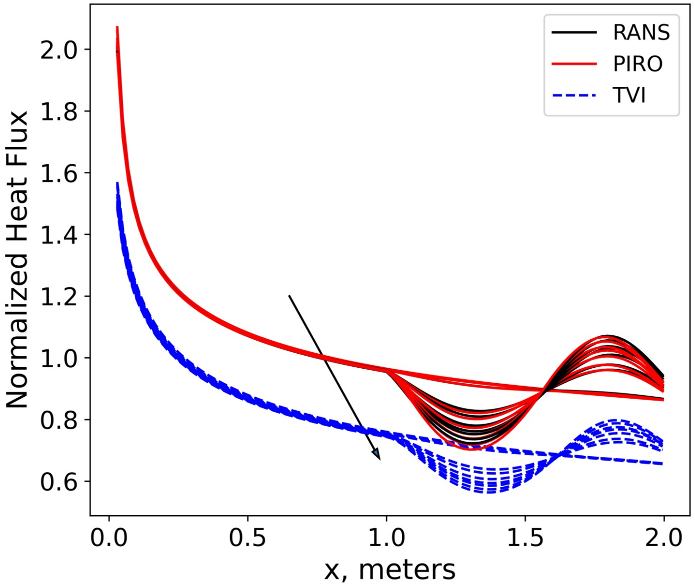






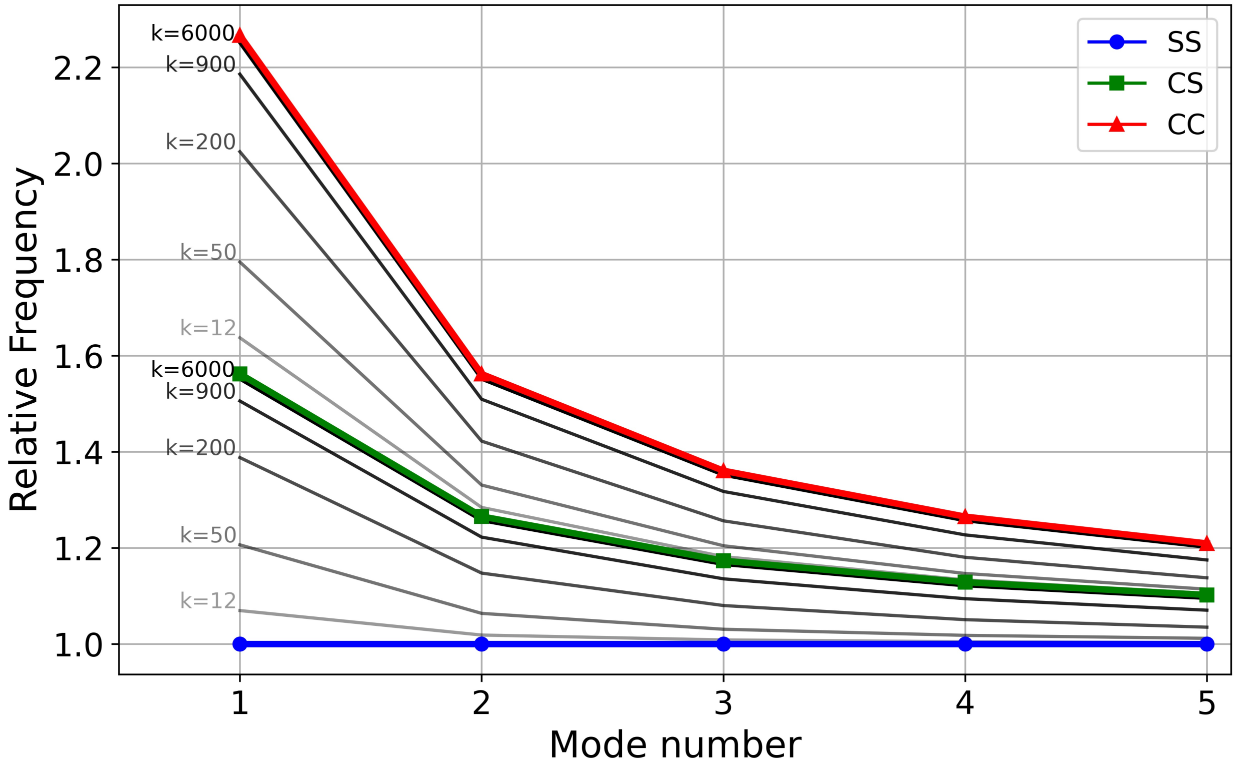


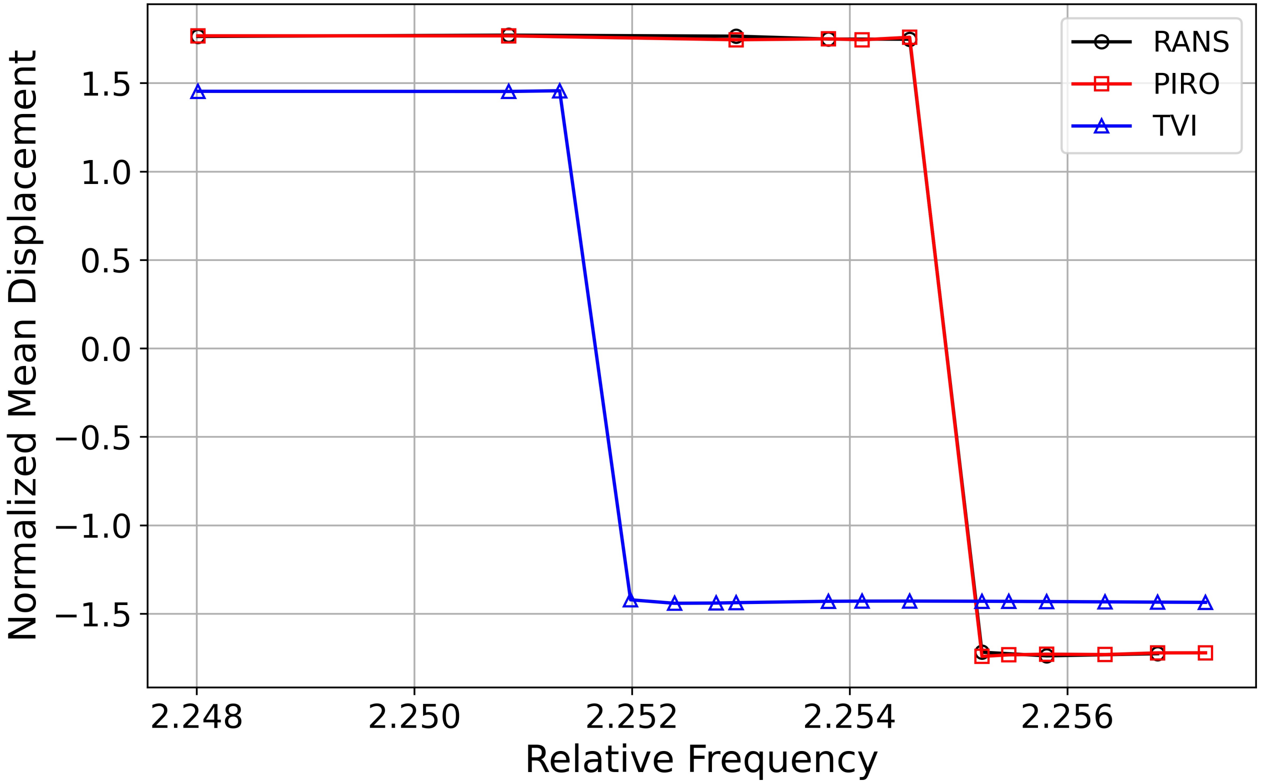




APUS Lab
Aerospace mu

Enabling parametric study as well





Key takeaways

Summary:

- Presented the formulation of Physics-Infused Reduced-Order Modeling.
- Demonstrated the methodology for a hypersonic aerothermodynamic application.
- Comparing to conventional aerothermal surrogate:
  - Generalize well to operating conditions and thermoelastic responses not in the training data set.
  - Requires <10^2 samples for any response, v.s. 10^3-10^4 samples → Much less samples
  - Computational cost 90 ms, v.s. 50 ms → Similar computational efficiency

Future Work:

- Extend the methodology for general DAE problems – Open to collaborations!
- Develop a general framework for systematic creation of physics-infused ROM.
- Couple to frameworks of multi-disciplinary optimization.

Thank you!

Questions?

Contact: daning@psu.edu
Lab website: apus.psu.edu