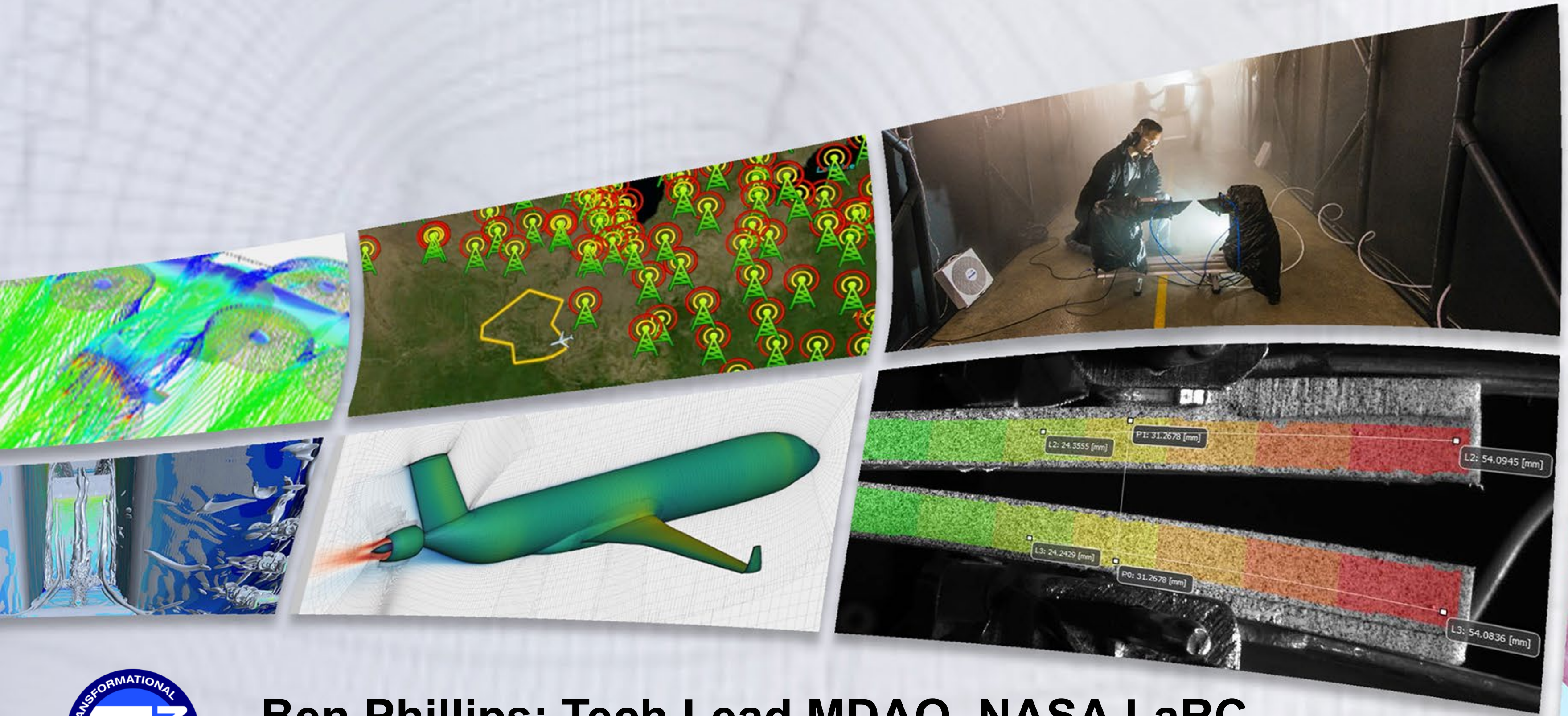
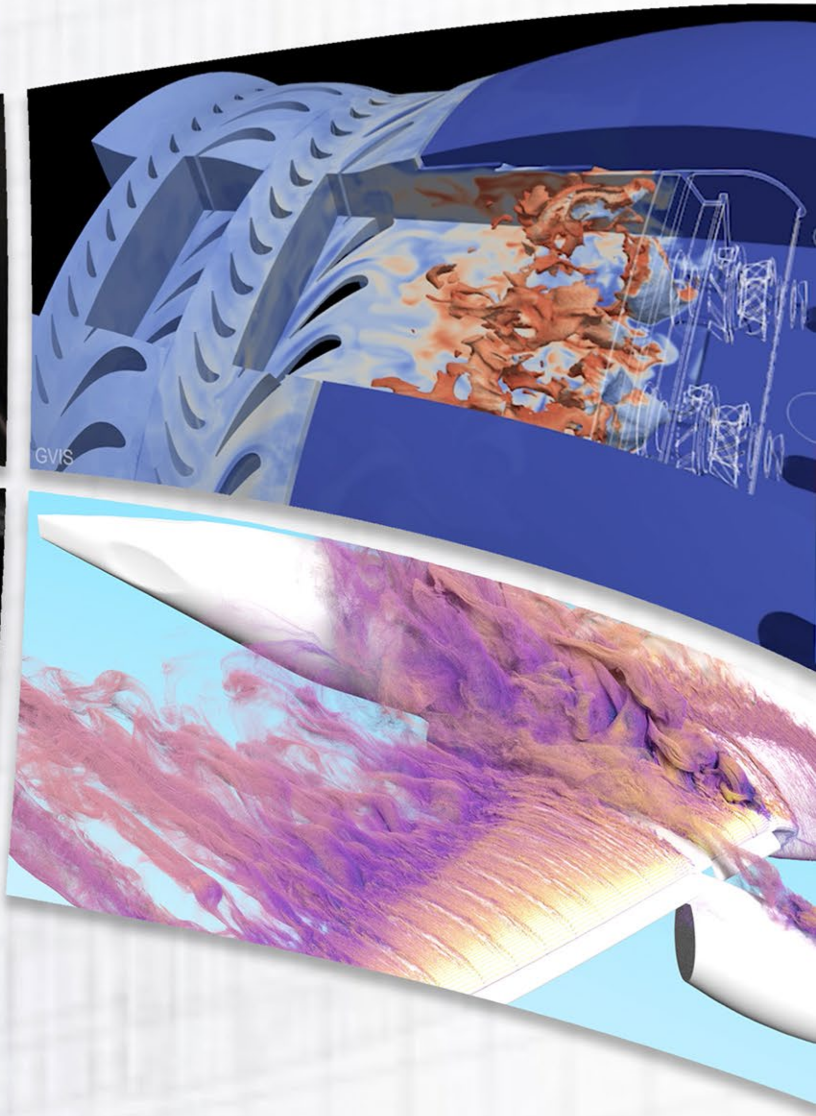
National Aeronautics and Space Administration

Transformational Tools and Technologies (T^3) Project

End-to-End Uncertainty Quantification with Analytical Derivatives for Design Under Uncertainty





Ben Phillips: Tech Lead MDAO, NASA LaRC

Joanna Schmidt: NASA LaRC

Rob Falck: NASA GRC

Eliot Aretskin-Hariton: NASA GRC

www.nasa.gov

Outline

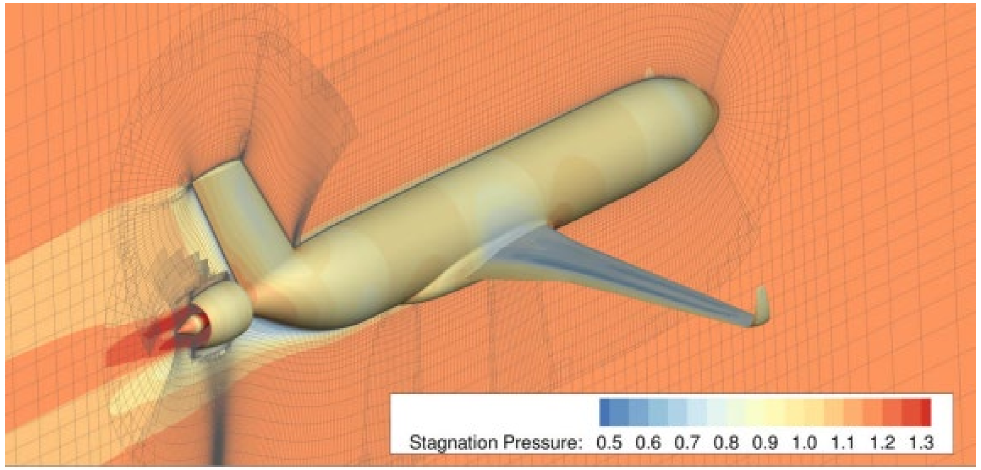
- Objectives for Design Under Uncertainty
- What Do We Have in the toolbox?
- Why Do We Care About Analytical Derivatives?
- How Do We Get Partial Derivatives for a Confidence Interval?
- Computational Costs
- Demonstration Problem

Objectives for Design Under Uncertainty

Recently completed Technical Challenge (TC) in MDAO established gradient-based multidisciplinary design optimization (MDO) that leverages analytical derivatives as a foundational approach throughout NASA's Aeronautics Research Mission Directorate (ARMD).

New concepts bring new challenges and uncertainties

- Extend gradient-based MDO with analytical derivatives to design under uncertainty
- Address the two primary roadblocks to implementation of design under uncertainty
    - Computational costs
    - Additional complexity of incorporating UQ
- Integrate uncertainty analysis into “native” OpenMDAO analysis and optimization



High-fidelity coupled aero-propulsive optimization
Gray et al. https://doi.org/10.2514/1.C036103

Aeroacoustic UAM rotor optimization
Ingraham et al. https://doi.org/10.2514/6.2019-1219

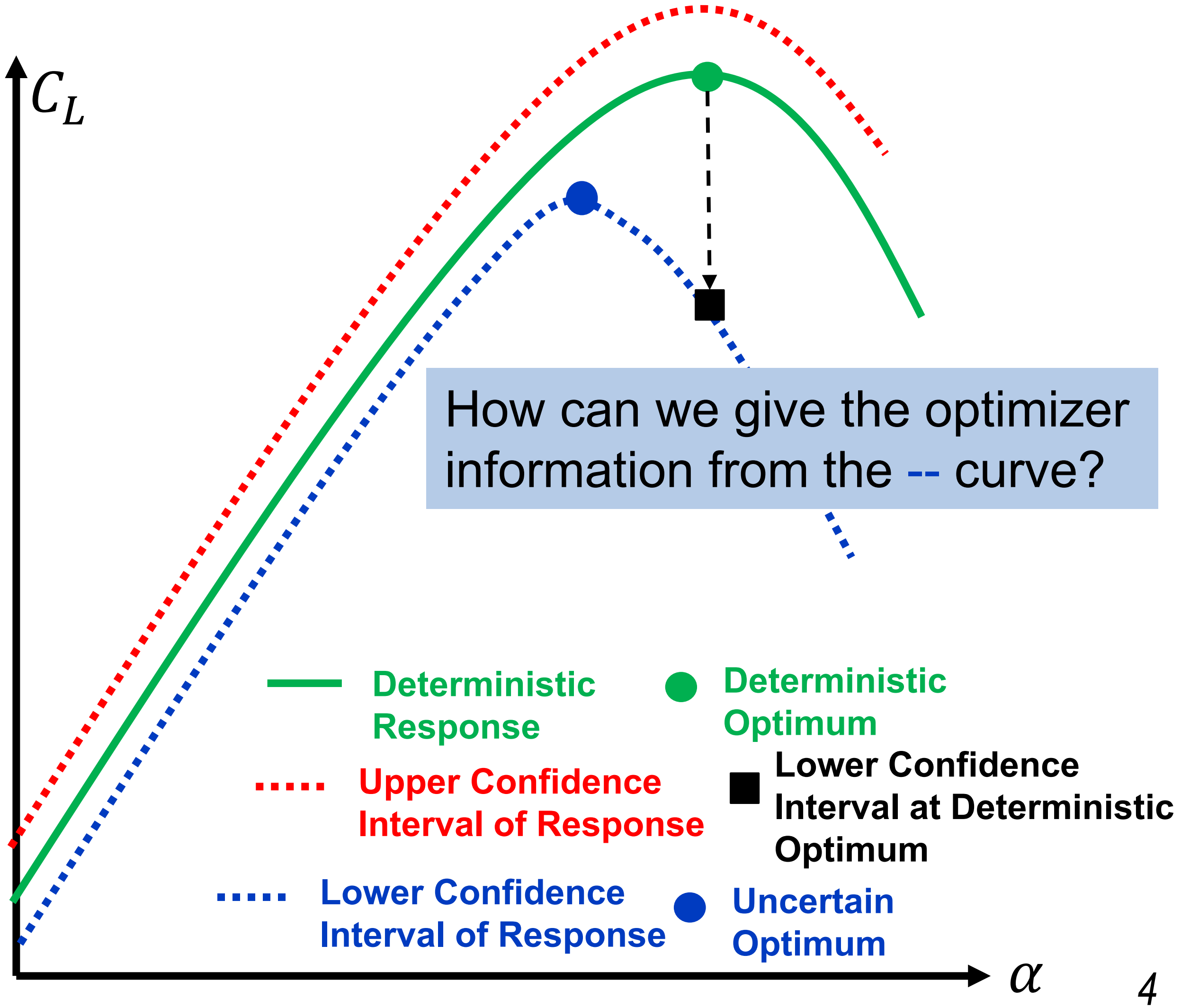
Objectives for Design Under Uncertainty

Example of $C_{L_{max}}$ optimization

$J = f(X)$ Optimization objective function —
Solution: ●

$J = f(X, U)$ Optimization objective function --
Solution: ●

- Optimizing — yields higher expected value for $C_{L_{max}}$ ● but a probabilistic chance (95% CI) of significantly lower actual CL_{max} ■
- Optimizing -- yields lower expected value for $C_{L_{max}}$ but higher maximum of lower probabilistic $C_{L_{max}}$ ●
- No free lunch! Trading lower expected value for higher "worst case" value



What do we have in the toolbox?

OpenMDAO:
- Open source python software
- Developed and supported by TTT
- Open source platform for systems analysis and MDO
- Enables decomposition of models easing implementation and maintenance
- Focused on gradient-based optimization of tightly coupled MDO problems with analytical derivatives
- https://github.com/OpenMDAO/OpenMDAO

UQPCE: Uncertainty Quantification with Polynomial Chaos Expansion
- Open source python software
- Originally developed under Commercial Supersonic Technology (CST), supported by TTT
- Tractable, modular non-deterministic analysis & design
- Efficient generalized nonintrusive point-collocation PCE
- External wrapper and “Black Box” ability, now integrated into OpenMDAO
- https://github.com/nasa/UQPCE/

[Uncertainty Quantification with Polynomial Chaos Expansion]

5

Why Do We Care About Analytical Derivatives?

Analytical derivatives are “cheat codes” for the optimizer

- In gradient-based optimization, the optimizer needs to estimate what direction to go next
- Usually, methods like finite difference or complex step are used to estimate gradients

$$C_L(\alpha) = C_{L_{\alpha=0}} + C_{L_\alpha}\alpha$$

Analytical partial derivative
Computationally Cheap*

$$\frac{\partial C_L}{\partial \alpha} = C_{L_\alpha}$$

Finite Difference Approach
Computationally very expensive and inaccurate

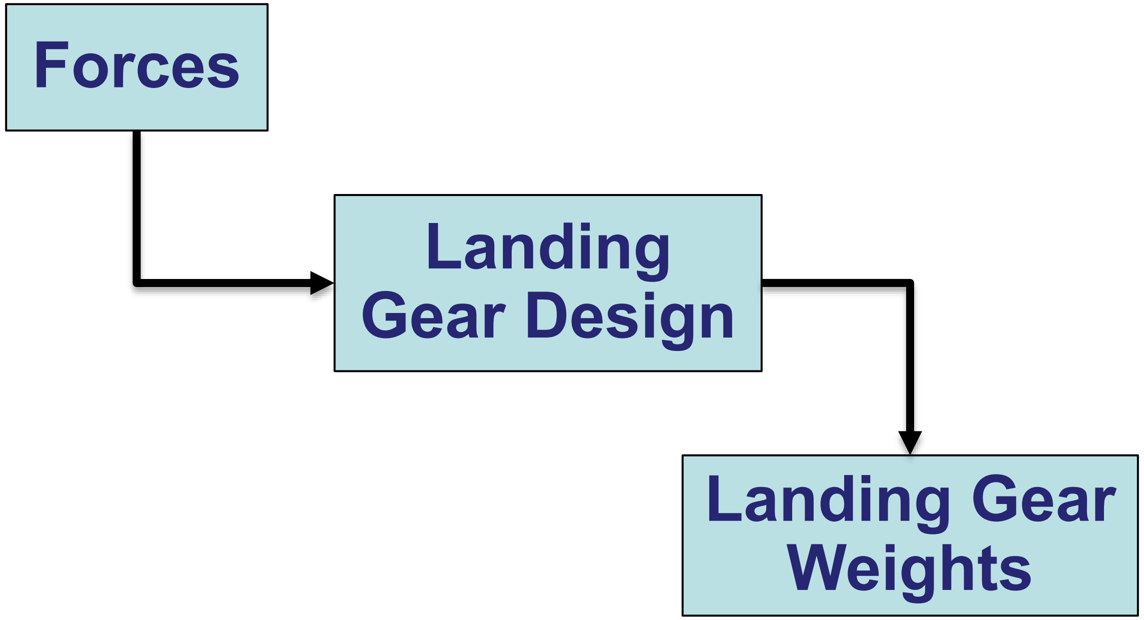
$$\frac{\partial C_L}{\partial \alpha} \approx \frac{C_L(\alpha + \Delta\alpha) - C_L(\alpha)}{\Delta\alpha}$$

*Requires underlying analysis code to produce analytical derivatives which is not trivial and can require significant upfront development costs

Why Do We Care About Analytical Derivatives?

OpenMDAO's (OM) "Secret Sauce": MAUD (Modular Analysis and Unified Derivatives)

- If the designer can supply partial derivatives from their subsystem, OM can do the heavy lifting



All other subsystem partial derivatives

$R = f(X)$

R is function of design variables only

$$\frac{\partial Weight}{\partial Forces} \cdots \frac{\partial R_i}{\partial X_i} \Rightarrow \boxed{\text{OM}} \Rightarrow \frac{dR}{dX}$$

Landing gear subsystem contribution to system level analysis

Total Derivatives

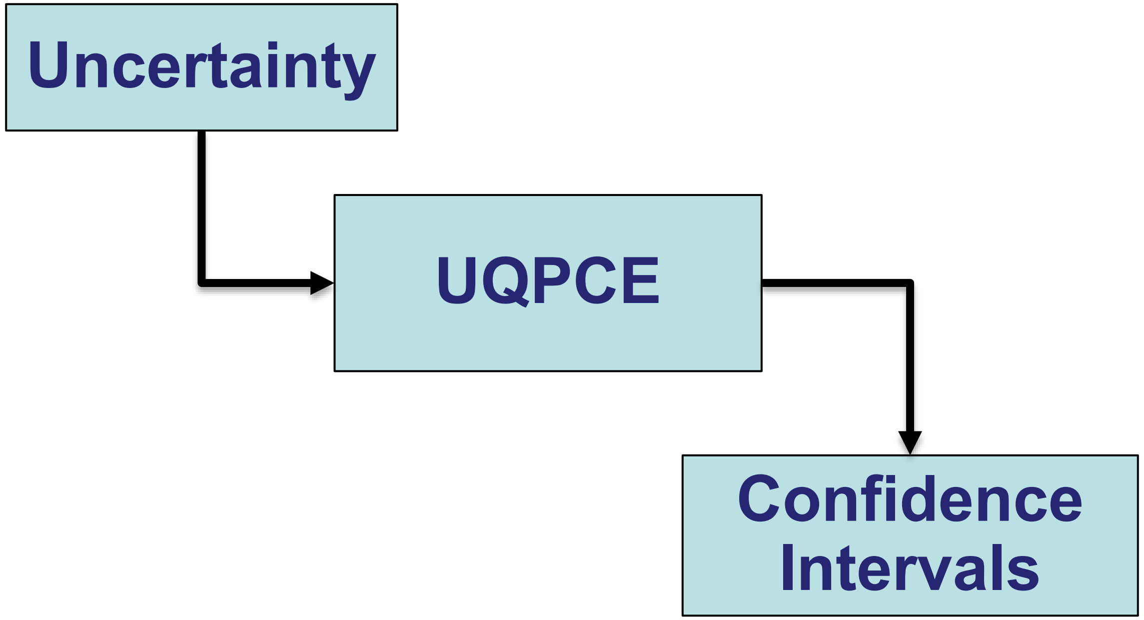
How Do We Get Partial Derivatives for Confidence Intervals?

We need partial derivatives with uncertainty included for OM to estimate total derivatives

- Confidence intervals (CI) require a binning procedure to estimate
- Binning is non-differentiable
- CI’s enable analysis/design with epistemic uncertainties

$$R = f(X, U)$$

R is now a function of design variables and uncertain variables



$$\frac{\partial CI}{\partial X} \cdots \frac{\partial R_i}{\partial X_i} \Rightarrow \boxed{\text{OM}} \Rightarrow \frac{dR}{dX}$$

Partial derivative of confidence interval with respect to inputs (Oversimplification)

Total Derivatives

How Do We Get Partial Derivatives for Confidence Intervals?

How can we make a confidence interval differentiable? Take some inspiration from the machine learning world, hyperbolic tangent activation function:

$$f(\vec{x}, z, \omega) = \sum_{i=1}^{n} 1 - \frac{1 + \tanh\left(\frac{\vec{x} - z}{\omega}\right)}{2}$$

Continuous counting function that provides an approximate count of the number of elements in $\vec{x}$ that are less than or equal to z. ω controls how sharp the transition from 1 – 0 occurs.

We can now form an expression for a residual based on a specific significance level

$$\mathcal{R}_z(\vec{x}, z, \omega) = \sum_{i=1}^{n} f_i(\vec{x}, z, \omega) - \left(1 - \frac{a}{2}\right) n = \sum_{i=1}^{n} f_i(\vec{x}, z, \omega) - 0.975n$$

$$(a = 0.05)$$

How Do We Get Partial Derivatives for Confidence Intervals?

With an expression for the residual, a Newton solver can be used for convergence

$$\mathcal{R}_z(\vec{x}, z, \omega) = \sum_{i=1}^{n} f_i(\vec{x}, z, \omega) - 0.975n$$

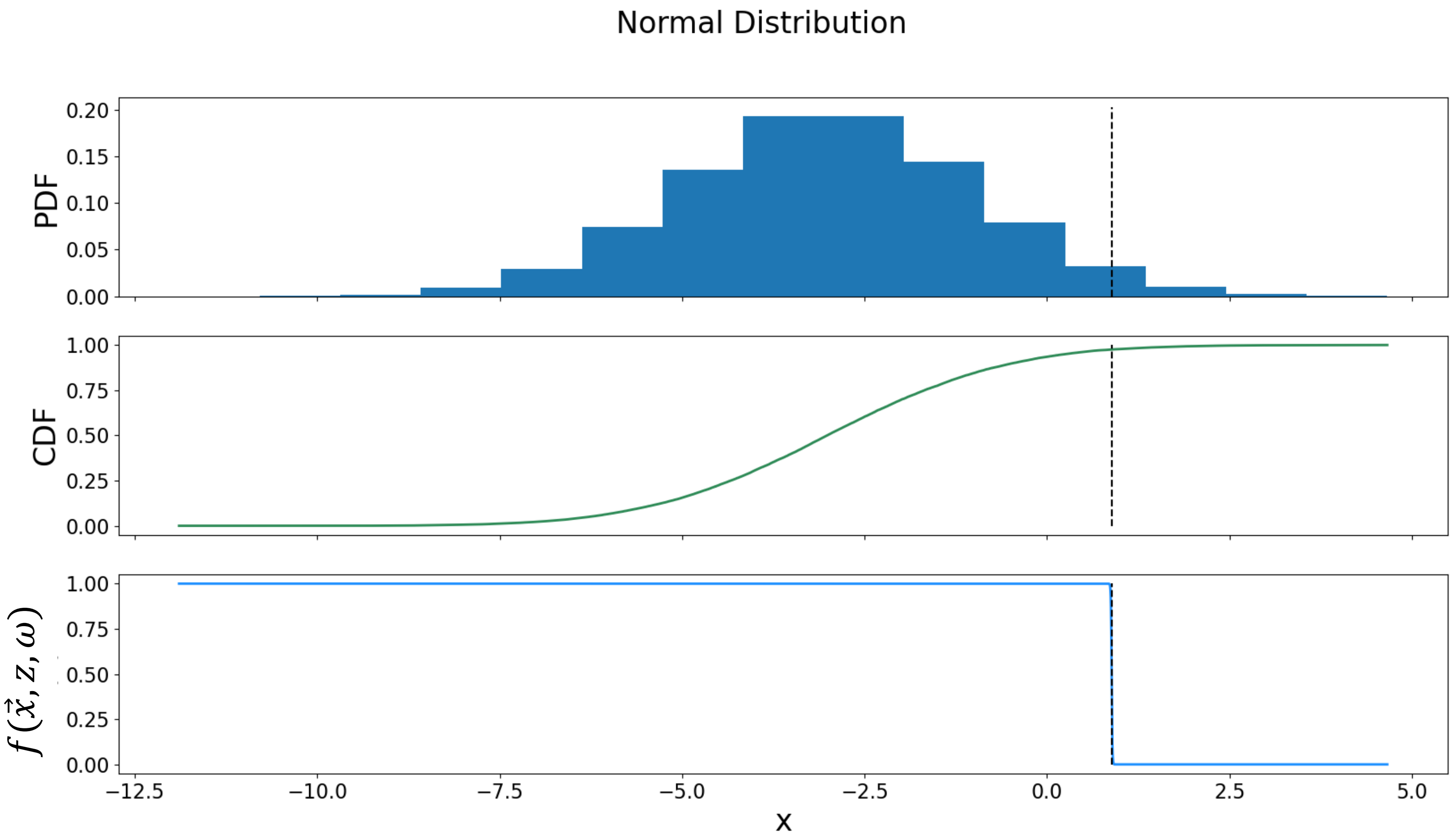
Given an initial guess of the 95% confidence interval based on a normal distribution:

$$z_{guess} = \mu + 2\sigma$$

The implicit function theorem can be applied to this process to get the sensitivity of $z\ wrt\ x$. Combined with other derivatives in the computational chain, OpenMDAO can obtain the total derivative of uncertainty metrics with respect to the design variables of the problem.

This process decouples the computational cost from the number of design variables

How Do We Get Partial Derivatives for Confidence Intervals?



Probability Density Function (PDF)

Cumulative Distribution Function (CDF)

Activation Function

Computational Costs

Computational savings scale with number of design variables, n_{dv} and number of function calls to build the PCE model, n_{pce}. Per-iteration function call computational costs:

Function calls using finite difference, n_{fd}	$n_{fd} = n_{pce}(n_{dv} + 1)$
Function calls using analytical derivatives , n_{an}	$n_{an} = n_{pce}$
Computational savings, C on per-iteration basis	$C = n_{pce} n_{dv}$

Demonstration Problem

Input	Range
t/c	[0.1, 0.15]
P_{minx}	[0.3, 0.5]
C	[0.01, 0.043]
λ	[0.1, 1]
Λ	[0, 40] (deg)
b	[20, 50] (m)
c_r	[5, 7](m)

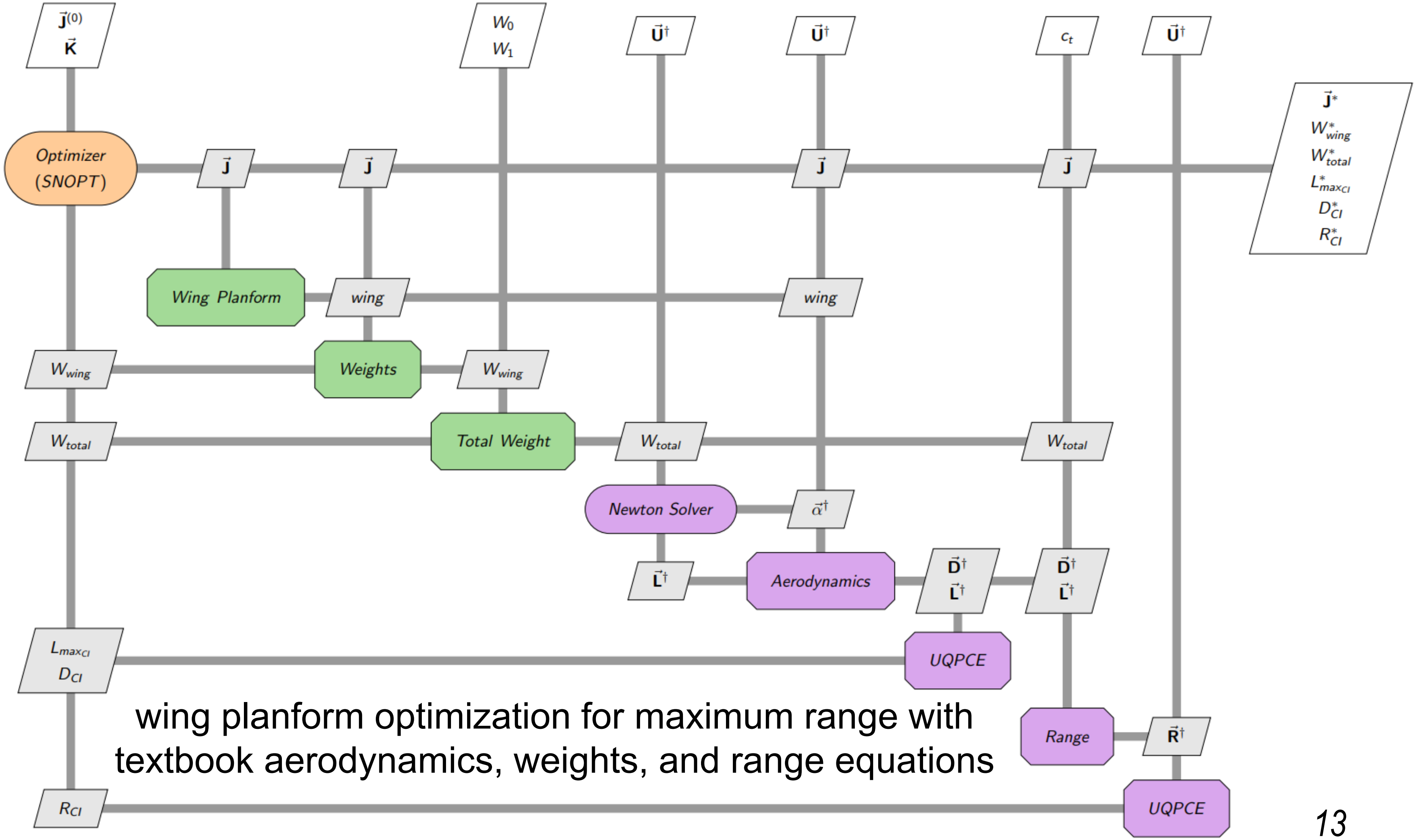
design parameters

Input	Interval
$\%_L$	$[0, 25\%]$
$\sigma_{C_{D_0}}$	$[4, 6]$
κ_α	$[0.85, 0.95]$
S_{csw}	$[0.05, 0.2]$

uncertain parameters (epistemic)

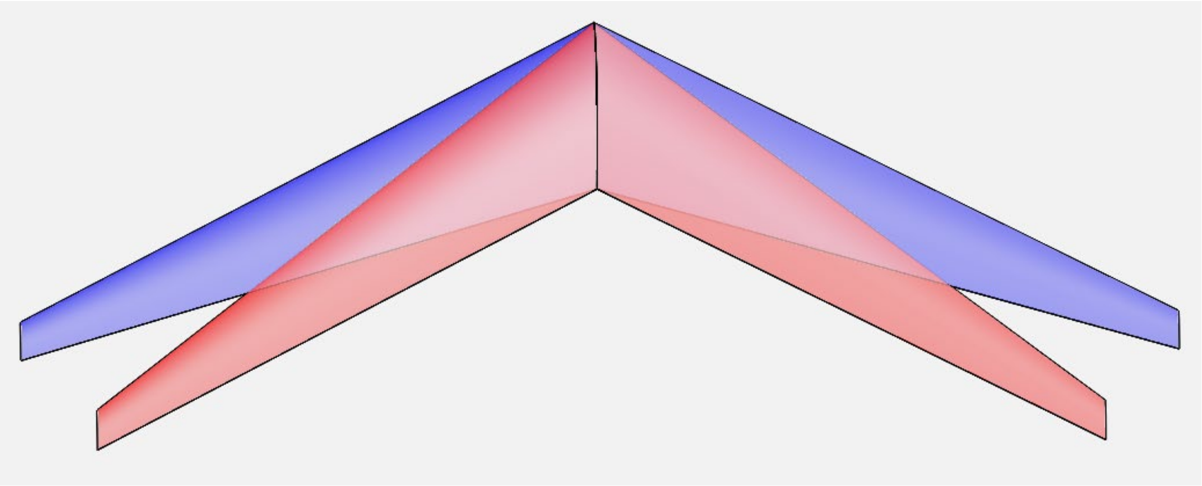
Input	Distribution	Mean	Std. Dev.
σ_{C_L}	Gaussian	$2x10^{-5}$	$3x10^{-6}$
M	Gaussian	0.72	0.02
c_t	Gaussian	0.45	0.045 (1/hr)

uncertain parameters (aleatory)



wing planform optimization for maximum range with textbook aerodynamics, weights, and range equations

Demonstration Problem



Comparison of optimal wing planform designs (deterministic in blue, uncertain in red)

Variable	range (Uncertain)	range (Deterministic)
t/c	0.1	0.118
P_{minx}	0.312	0.3
C	0.043	0.043
λ	0.196	0.185
Λ	30.59 (deg)	24.32 (deg)
c_r	5 (m)	5 (m)
b	45.62 (m)	46.00 (m)
W_{weight}	9291.4 (mean, kg)	8338.0(kg)

Variable	$range_{CI}$ (Uncertain)	$range$ (Deterministic)
Mean	3679.6 (nmi)	3723.1 (nmi)
Confidence Interval	[2581.2, 5414.1] (nmi)	[2342.4, 5653.9] (nmi)

Function calls for finite difference optimization: 30,952

Function calls for analytical derivative optimization: 3,816

Computational costs reduced by 87.7%

10.2% increase in lower confidence interval for range with 1.2% decrease in mean value

Conclusions and Future Work

Developed a framework for design under uncertainty with analytical derivatives

- Native OpenMDAO integration
- Physics/Application agnostic
- Any combination of uncertain and/or deterministic constraints and objectives
- Surrogate model outputs available for custom objective functions
- Demonstrated significant computational savings over finite difference and complex step

Future Work

- Open source release code to public repository
- Integrate with Model Based Systems Analysis and Engineering (MBSAE)
- Integrate with the conceptual aircraft design tool Aviary
- Apply codes and methods to higher fidelity analysis
- Incorporate other methods (multifidelity, adaptive sampling etc.)
- Collaborate with other researchers

Acknowledgements

This work was funded by the Transformational Tools and Technologies Project (TTT) within the Transformative Aeronautics Program (TACP)

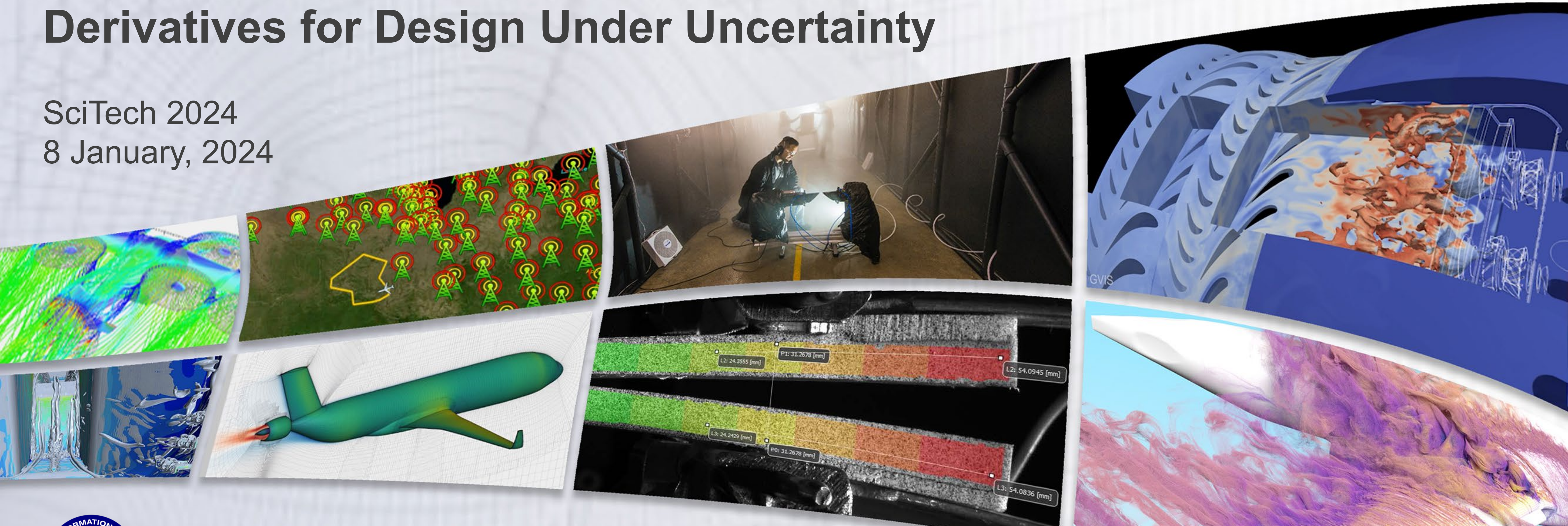
Previous support for UQPCE was provided by the Commercial Supersonic Technology Project (CST)

National Aeronautics and Space Administration

Transformational Tools and Technologies (T^3) Project

End-to-End Uncertainty Quantification with Analytical Derivatives for Design Under Uncertainty

SciTech 2024
8 January, 2024



Ben Phillips: Tech Lead MDAO, NASA LaRC
Joanna Schmidt: NASA LaRC
Rob Falck: NASA GRC
Eliot Aretskin-Hariton: NASA GRC

www.nasa.gov