National Aeronautics and Space Administration

Transformational Tools and Technologies (T^3) Project

Design Under Uncertainty for Conceptual Aircraft Design Leveraging Analytical Gradients

AIAA SciTech
9 January, 2025

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

Innovative solutions through foundational research and cross-cutting tools

Design Under Uncertainty for Conceptual Aircraft Design Leveraging Analytical Gradients

Overview

Uncertainty quantification (UQ) in a Multidisciplinary Design Analysis and Optimization (MDAO) framework is necessary to evaluate aircraft at the conceptual design level

- Lack of historical data available for unconventional concepts
- Guide future investments to reduce uncertainty
- Inform certification by analysis
- Enables **system level view** of uncertainty for aircraft and its operation

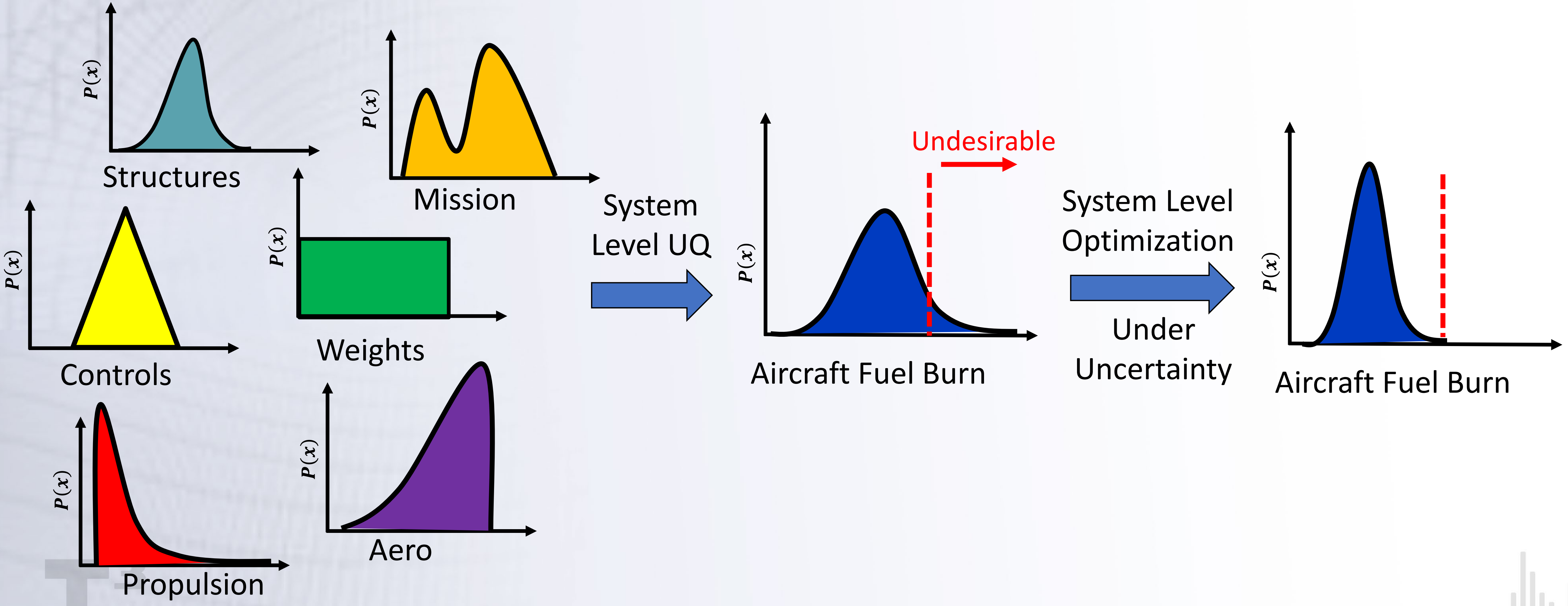
Design under uncertainty in a Multidisciplinary Design Optimization (MDO) framework enables design of highly coupled concepts and ability to mitigate against potential negative impacts of uncertainties early on in the design process

Extend previously developed methods for design under uncertainty with analytical gradients to conceptual aircraft design

To satisfy the US aviation climate action plan's aggressive timelines, "unconventional" may become "everyday"

Design Under Uncertainty for Conceptual Aircraft Design Leveraging Analytical Gradients

System Level Design Under Uncertainty



Tools and Methods for Design Under Uncertainty

OpenMDAO:

- Open source Python software 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 for generalized nonintrusive point-collocation PCE
- External wrapper and “Black Box” ability, integrated with OpenMDAO
- https://github.com/nasa/UQPCE/

Aviary

- Open source Python software for NASA’s next generation conceptual aircraft design tool
- Built on top of OpenMDAO with modular subsystems
- Core subsystems enable traceability to legacy design tools, FLOPS and GASP
  https://github.com/OpenMDAO/Aviary

Case Study Problem Formulation

To demonstrate the use of analytical gradients in design under uncertainty, an optimization of a single aisle subsonic transport aircraft was used as a case study

Uncertainty in mission requirements driving a design under uncertainty problem

Initial geometry and configuration derived from a NASA internal 154 pax concept vehicle, the N3CC

OpenVSP's Python API along with a geometry wrapper developed by the Model-Based Systems Analysis and Engineering (MBSA&E) team was used for the optimization

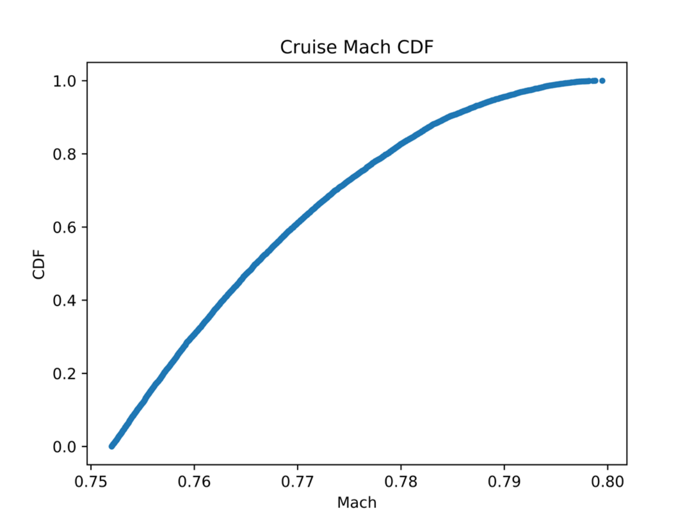
N3CC OpenVSP Model

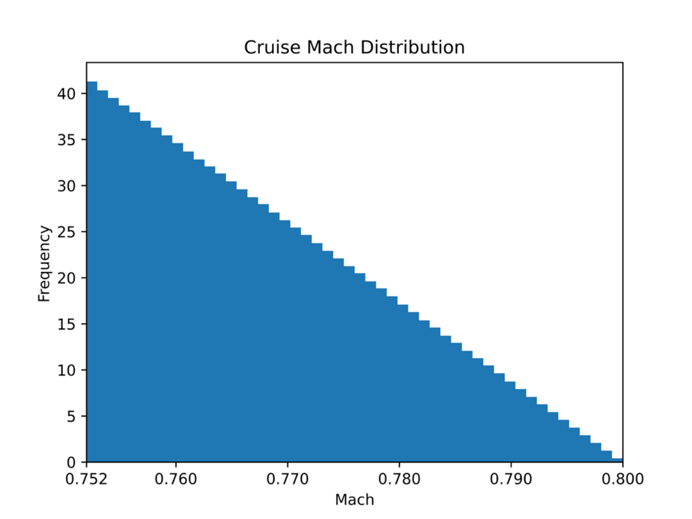
Case Study Problem Formulation

Uncertainty in mission requirements for a subsonic, commercial transport driving a design under uncertainty problem

- 3000 nmi mission three times daily
- Uncertainty in the amount of time for each mission
- Gram-Schmidt orthogonalization used to model the triangular distribution for the PCE model

Maps generated by the Great Circle Mapper - copyright © Karl L. Swartz.





- Assuming 60 min aircraft turn time gives 430 min for each mission (24 hr = 1440 min)
- Aircraft turn time uncertainty requires missions to be flown faster to stay on schedule
- Publicly available data on differences in scheduled vs actual aircraft turn times formed basis for uncertain distributions
- Cruise time range from 430 min (nominal) to 404 min (fastest)
- Converted mission time constraint to Mach number in cruise for ease of optimization

More details in manuscript

Case Study Problem Formulation

Uncertainty in mission requirements for a subsonic, commercial transport driving a design under uncertainty problem

- 3000 nmi mission three times daily
- Uncertainty in pax weight and load factor
  - How much does each pax weigh?
  - How many bags, what do they weigh?
  - How full is the plane?

Total passenger payload of aircraft designed for 154 pax:
μ = 30400 lb, σ = 5500 lb

- CDC data for US population for pax weight distribution
  - Men μ = 199.8, σ = 43.7 lb
  - Women μ = 170.8, σ = 46.5 lb
  - Children [20, 180] lb
- Assume pax are 49% women, 46% men, 5% children
- Assume 40% of pax travel with carry on luggage only, 55% check a bag, 5% check two bag
- Assume carry on luggage: μ = 20 lb, σ = 5 lb; checked luggage: μ = 40 lb, σ = 10 lb
- Average monthly load factors from 1/1/2000 – 6/1/2024 for US flights (removing 9/11 and Covid impact months)

More details in manuscript

Design Under Uncertainty for Conceptual Aircraft Design Leveraging Analytical Gradients

Case Study Problem Formulation

Three optimizations with the same design variables, but different objective functions:

- **Mean fuel burn**
- **Upper bound of the 95% confidence interval (CI) for fuel burn**
- **Deterministic (no uncertainty)**

Input	Range
Takeoff Gross Weight (TOGW)	[80,000 , 200,000] lb
Wing Sweep	[10 , 45] deg
Wing Span	[100 , 160] ft

Design Variables

	Variable or Function	Size	Discipline
minimize	Mean or Upper Bound of the 95% CI Fuel Burn	1	
with respect to	Wing Sweep	1	Aerodynamics
	Wing Span	1	
	Design Gross Mass	1	Weights
	Summary Gross Mass	12	
	Climb Duration	12	Trajectory
	Cruise Duration	12	
	Descent Duration	12	
	Mass	552	
	Distance	540	
subject to	Mass Residuals and Constraints	36	Weights
	Throttle Constraints	720	Propulsion
	Range Residual	12	Trajectory
	Pseudospectral Constraints	1,080	

Problem Formulation

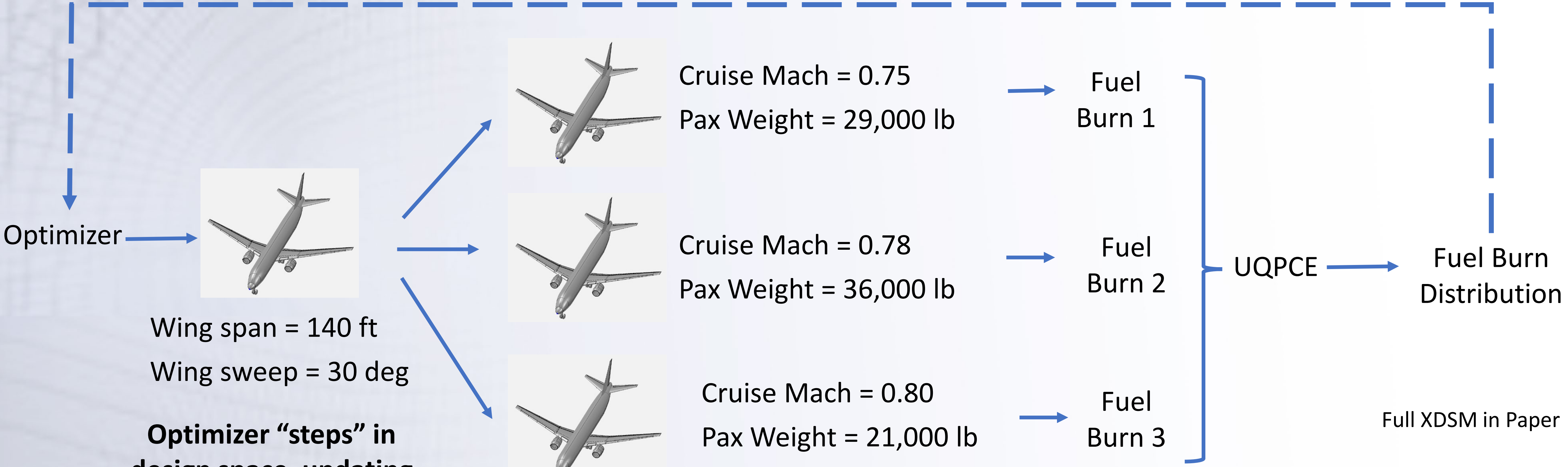
Input	Distribution		
Cruise Mach	Triangular	Lower Bound: 0.752	Upper Bound: 0.80
Passenger Payload Weight, lb	Gaussian	$\mu = 30,450$	$\sigma = 5500$

Uncertain Parameters

Mechanics of UQPCE & OpenMDAO (OM) Integration

Case study application for conceptual aircraft design under uncertainty

See previous work for details: [AIAA 2024-0173]



Optimizer "steps" in design space, updating design variables

Corresponding "step" in uncertainty space necessary

UQPCE takes information from the sampled uncertainty space and returns a distribution and accompanying partial derivatives to OM

Mechanics of UQPCE & Aviary Integration

To perform design under uncertainty in Aviary, some feature development was required

- Multi-mission capability enables optimization across different missions
- Off-design analysis capability enables optimized vehicle to be analyzed at non-optimized conditions

Optimizer

Mission 1 → Fuel Burn 1

Mission 2 → Fuel Burn 2

Mission 3 → Fuel Burn 3

Total Fuel Burn

Multi-mission optimization

Deterministic Optimized Design

Mission 1 → Fuel Burn 1

Mission 2 → Fuel Burn 2

Mission 3 → Fuel Burn 3

UQPCE → Fuel Burn Distribution

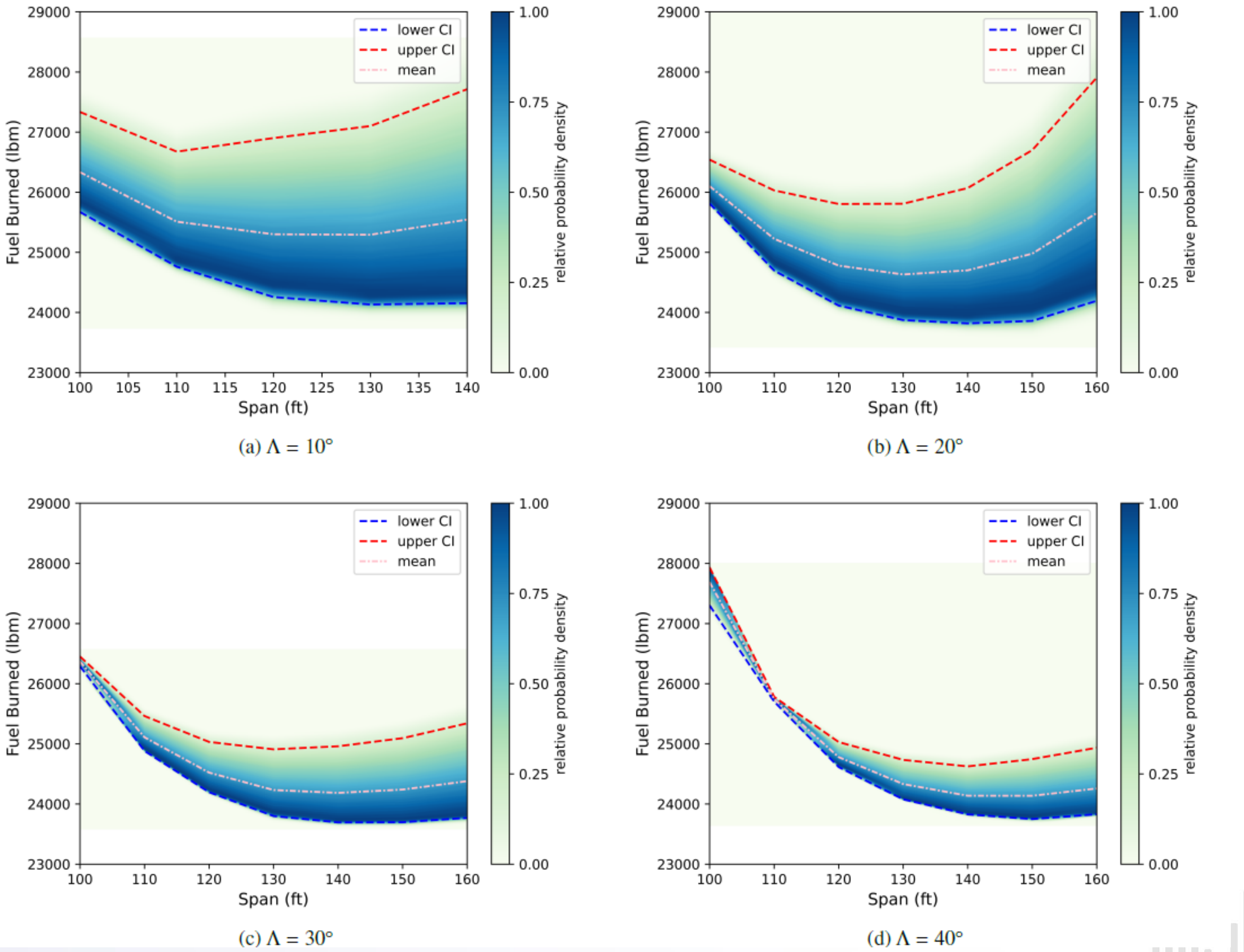
Off-Design Analysis

Post-Optimality Uncertainty Quantification (UQ after the fact)

Case Study Baseline Uncertainty Analysis

Design space exploration with uncertainty analysis natively inside OpenMDAO

- Parameter sweeps with uncertainty for traditional systems analysis
- Visualization of the uncertainty space prior to design under uncertainty
- Example from the case study showing fuel burn for fixed wing sweep values while varying wing span
- Depending on objective formulation, optimal design should fall between sweep = [30, 40] deg and span = [135, 145] ft

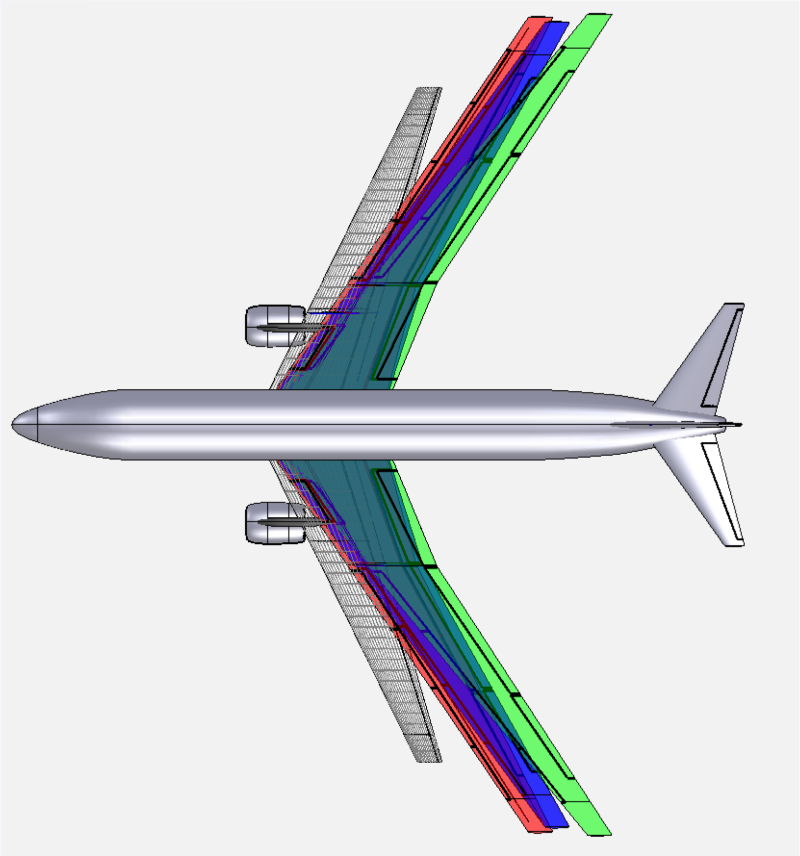


Case Study Results

Design variable outputs from optimal solutions differ based on the objective function

- Three optimizations with different objectives: mean, upper confidence interval (CI) and post-optimal UQ deterministic fuel burned

- All three optimizations add more sweep and span benefiting from the favorable aerodynamics in the empirical aerodynamic models without explicit structural or aeroelastic constraints

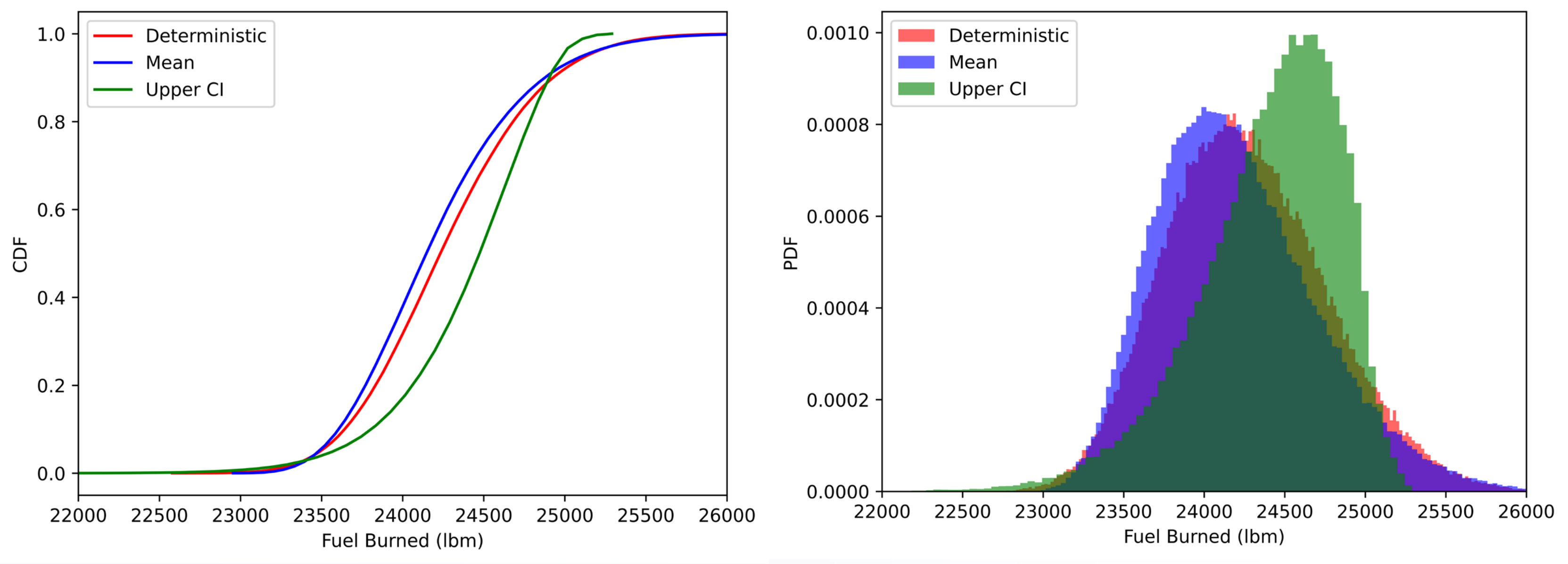


Objective	Wing Sweep	Wing Span	Operating Empty Weight
Deterministic Fuel Burn	34.0 deg	143.3 ft	80669 lb
Mean Fuel Burn	35.9 deg	141.6 ft	82318 lb
Upper Bound of the 95% CI Fuel Burn	39.1 deg	144.4 ft	83780 lb

Design Under Uncertainty for Conceptual Aircraft Design Leveraging Analytical Gradients

Case Study Results

Approach with analytical gradients enables direct manipulation of output distributions



- Three optimizations with different objectives: mean, upper confidence interval (CI) and post-optimal UQ deterministic fuel burned

- When optimizing the mean (blue curve), peak values in CDF shift left in comparison to post-optimal UQ deterministic (red curve)

Objective	Mean	CI
Deterministic Fuel Burn	24,262	[23,370 , 25,302] lb
Mean Fuel Burn	24,201	[23,383, 25,332] lb
Upper Bound of the 95% CI Fuel Burn	24,404	[23,367, 25,046] lb

- When optimizing upper CI (green curve), the tail of the CDF shifts to the left in comparison to other optimizations

Summary, Lessons Learned, and Future Work

Demonstrated gradient-based MDO with analytical derivatives for conceptual aircraft under uncertainty with UQPCE, Aviary and OpenMDAO

- The optimizer was able to manipulate the output distribution when given uncertain information
- Over 1100 variables and 1800 constraints
- Non “standard” distributions
- Mission requirements with uncertainty driving a design under uncertainty problem
- Computational cost for design under uncertainty was 13.6x deterministic optimization
- UQPCE libraries released open source

Monolithic optimization is not well suited for design under uncertainty with PCE

- The optimizer will solve the problem given
- Variance-based objective functions require sub-problem optimality

Future Work

- Parallelization for computational performance
- Derivatives through sub-problems
- Swappable atmosphere, incorporating winds
- Expand to other subsystems – EAP, Cryo etc.

National Aeronautics and Space Administration

Transformational Tools and Technologies (T^3) Project

Design Under Uncertainty for Conceptual Aircraft Design Leveraging Analytical Gradients

SciTech 2025
9 January, 2025

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

Innovative solutions through foundational research and cross-cutting tools

www.nasa.gov

Backup

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

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

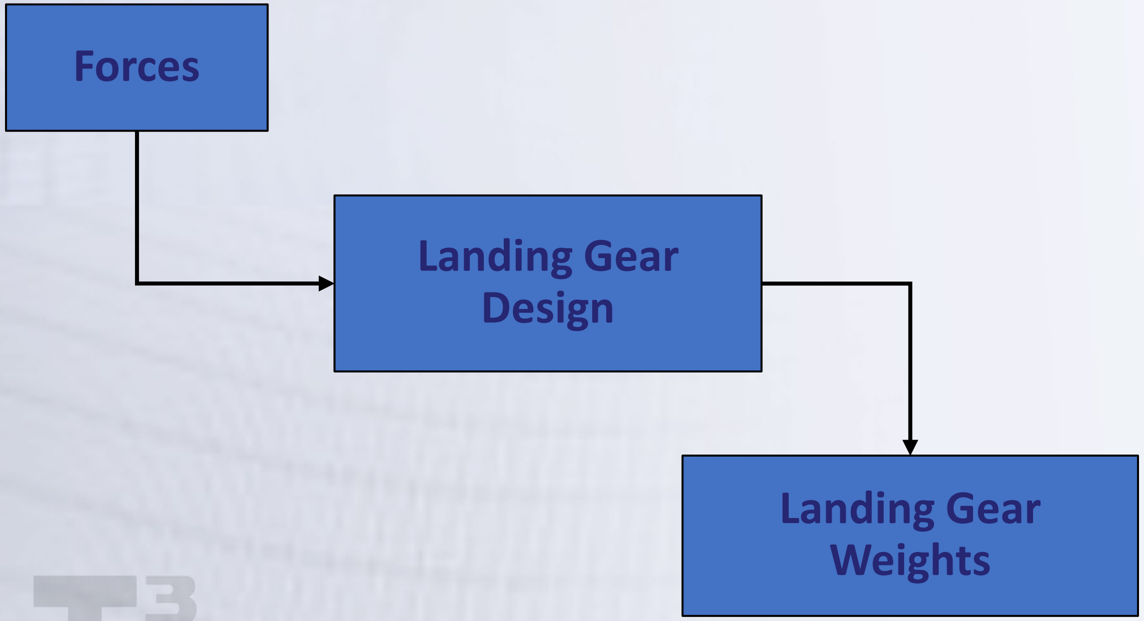
*Requires underlying analysis code to produce analytical derivatives which is not trivial and can require significant upfront development costs

Towards System Level Optimization Under Uncertainty

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

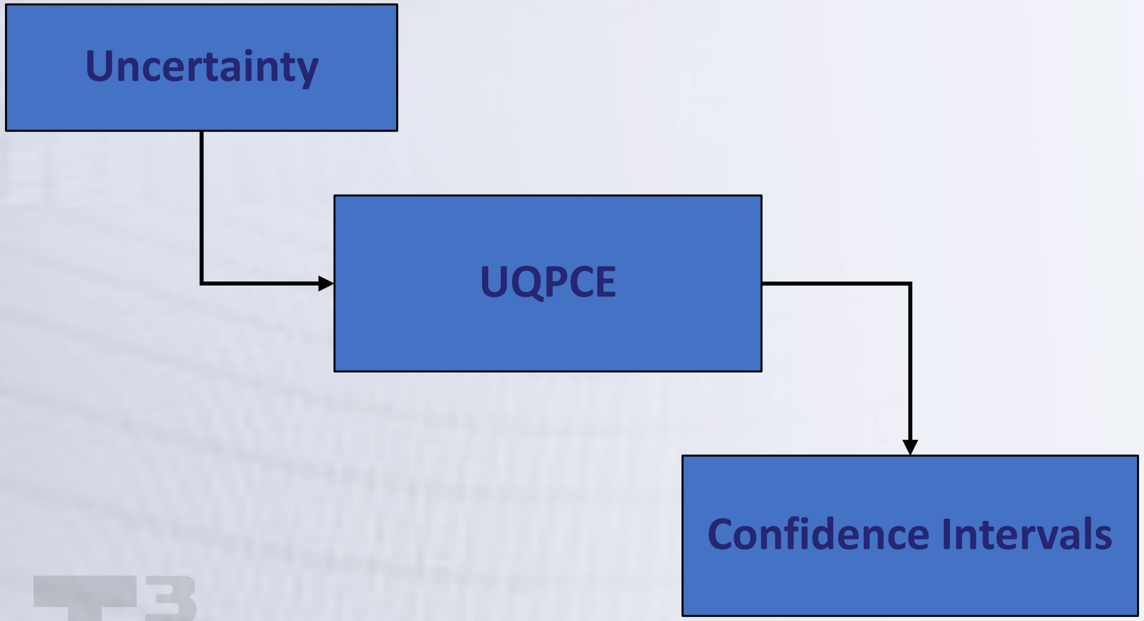
How Do We Get Partial Derivatives for Confidence Intervals?

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

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

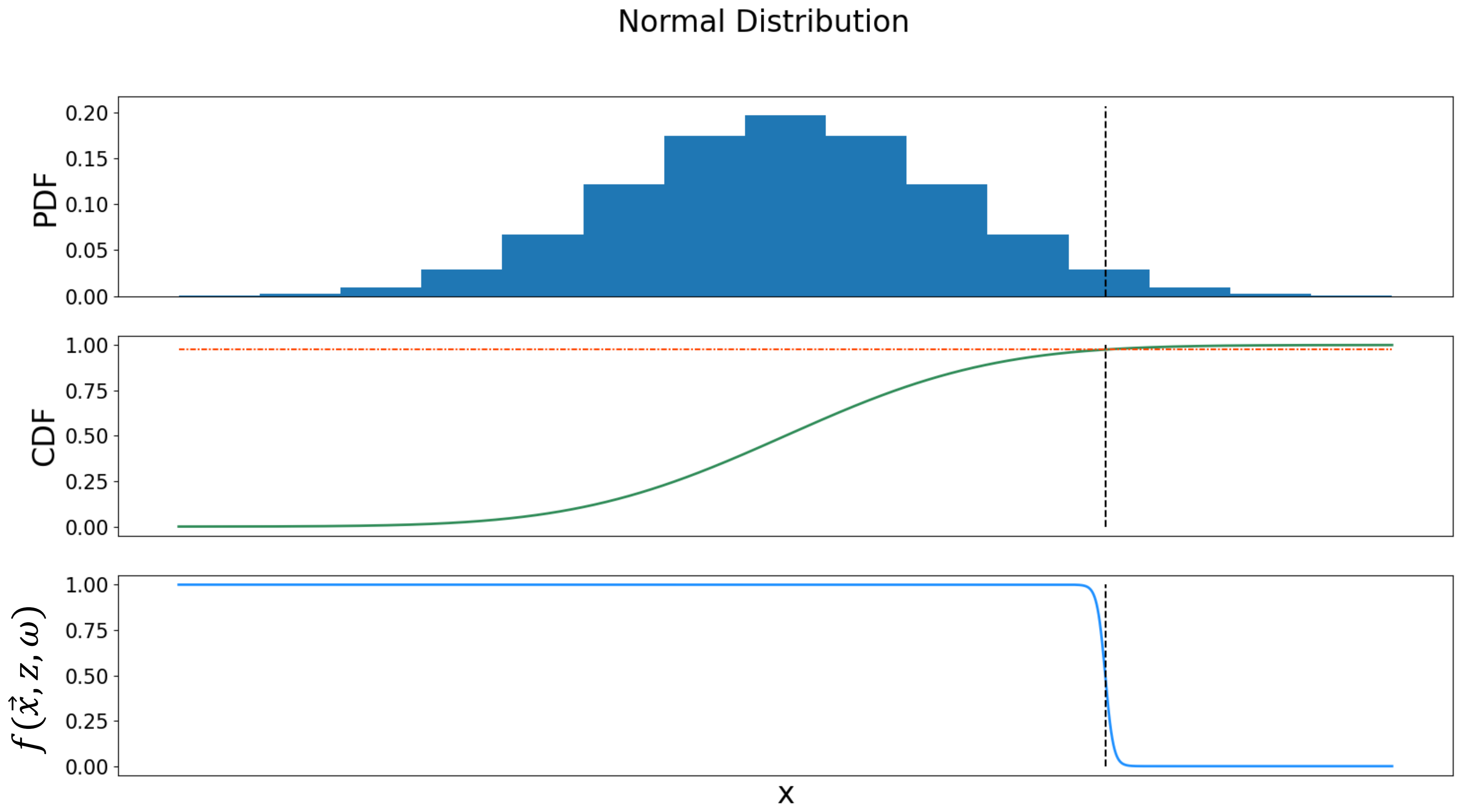
Given an initial guess of the 95% confidence interval based on a normal distribution:

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

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	$C = n_{pce} n_{dv}$