Uncertainty Quantification Workshop for Joint GN&C and L&D Face-to-Face @ MSFC

Ignorance vs. Variability

Considerations for Uncertainty Quantification Analyses

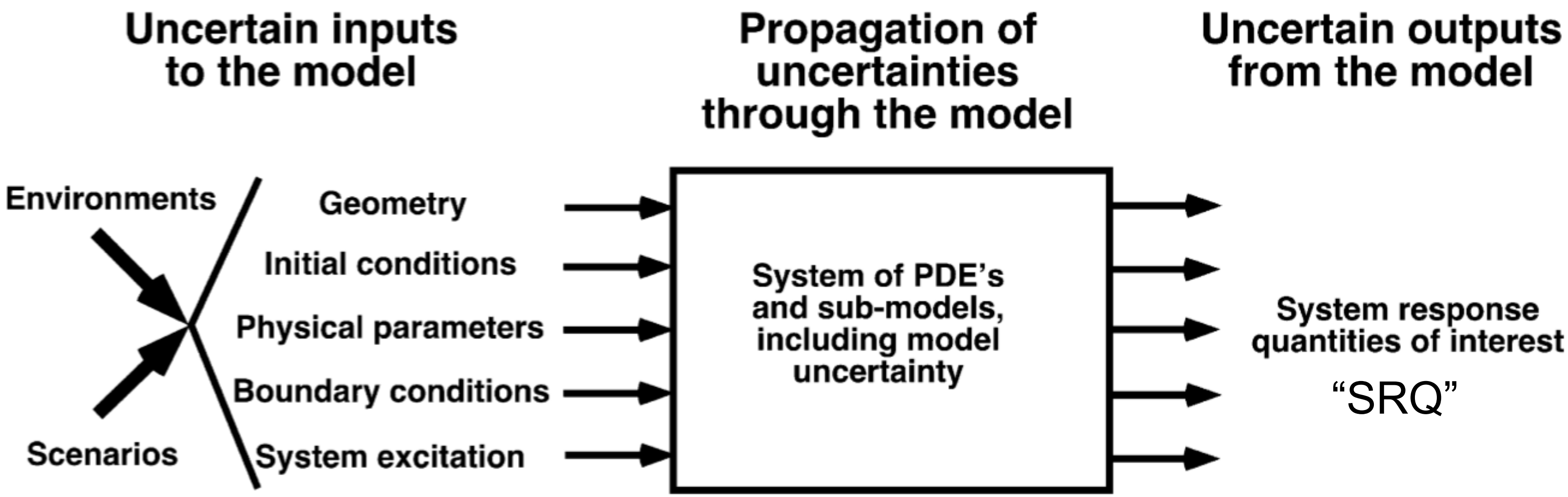
Erik S. Bailey, EDL GNC Systems Group, Jet Propulsion Laboratory, California Institute of Technology
Lee Peterson, Mechanical Division Staff, Jet Propulsion Laboratory, California Institute of Technology
April 18, 2018

Your modeling is probably more uncertain than you think it is.

- Today, we're going to focus on how to include **ignorance** in Uncertainty Quantification (UQ) analyses, and potential consequences of ignoring input ignorance
  - There are other topics we'll touch on today that warrant their own sessions, and I propose that we schedule them!
- We typically treat every input variable to a simulation as an independent random variable
- **Variability** is not the same as **Ignorance**
  - If we don't even know the bounding values of an input, we have a bigger problem!
- Treating everything as variability and excluding ignorance effects *obfuscates information* in Uncertainty Quantification
- Good news! There are ways to combine both and uncover/communicate more robust uncertainty.

Uncertainty Quantification (UQ) Components

Typical problem setup & workflow



"Inputs we vary to study system behavior/performance"

Classification types:
Aleatory (variability)
Epistemic (ignorance)

"Model of system of interest & execution machinery"

"Outputs customers care about for performance, margin assessment, etc."

When we say ‘Monte Carlo’…

(What we tend to do for UQ)

- We tend to vary our inputs only as:
  - Uniform Distributions (note: mean = median)
  - Gaussian Distributions (note: mean = median)
- The above distributions are mathematically convenient, and are emphasized when teaching Probability Theory
- Very rarely do we:
  - Use a different Distribution
  - Correlate inputs to our Monte Carlo Analyses
  - Ask ourselves: “do we REALLY know what that distribution is?”
    - If you don’t have measured data & statistical analysis to back it up, the answer is “no” and it should be treated as ignorance (an epistemic input)
  - Use more advanced sampling techniques to reduce computational load and results crowding the E[x] region
    - We rely on number of runs to ensure we properly populate the tails.

Typical Input Deck for a ‘Monte Carlo’

Example taken from Europa Lander DDL Simulation input deck

Units

Distribution Type

Classification/ Organization

Parameter Name/ID

Parameterization Values

Assoc. Frame

Revision Control Information

Category	Assembly	Parameter	Units	Lower Bound Value	Nominal Value	Upper Bound Value	Distribution	Coordinate Frame	Last Update	Source	Comment
Mass_Properties	Lander_Stowed	Mass	kg	279.424	377.6	475.776	Uniform	not applicable	1/12/17	Brant Cook	Lander properties in DS Configuration
Mass_Properties	Lander_Stowed	Xcg	mm		0		not applicable	LNDR Frame	1/12/17	Brant Cook	pos_error_xy_mag and pos_error_xy_az define relevant dispersions
Mass_Properties	Lander_Stowed	Ycg	mm		0		not applicable	LNDR Frame	1/12/17	Brant Cook	pos_error_xy_mag and pos_error_xy_az define relevant dispersions
Mass_Properties	Lander_Stowed	Zcg	mm	160.00	180.00	200.00	Uniform	LNDR Frame	1/12/17	Brant Cook	
Mass_Properties	Lander_Stowed	pos_error_xy_mag	mm	0	0	5	Uniform	LNDR Frame	1/12/17	Brant Cook	defines magnitude of CG position error in xy plane
Mass_Properties	Lander_Stowed	pos_error_xy_az	deg	0	0	360	Uniform	LNDR Frame	1/12/17	Brant Cook	defines orientation/clocking of interface position error in xy plane
Mass_Properties	Lander_Stowed	Ixx	kg*mm^2	40074636.36	54154914.0	68235191.64	Uniform	LNDR Frame	1/12/17	Brant Cook	
Mass_Properties	Lander_Stowed	Iyy	kg*mm^2	36386791.6	49171340.0	61955888.4	Uniform	LNDR Frame	1/12/17	Brant Cook	
Mass_Properties	Lander_Stowed	Izz	kg*mm^2	64797834.34	87564641.0	110331447.7	Uniform	LNDR Frame	1/12/17	Brant Cook	
Mass_Properties	Lander_Deployed	Mass	kg	279.424	377.6	475.776	Uniform	not applicable	1/12/17	Brant Cook	Lander properties in open (petals fully deployed) SkyCrane Configuration
Mass_Properties	Lander_Deployed	Xcg	mm		0		not applicable	LNDR Frame	1/12/17	Brant Cook	pos_error_xy_mag and pos_error_xy_az define relevant dispersions
Mass_Properties	Lander_Deployed	Ycg	mm		0		not applicable	LNDR Frame	1/12/17	Brant Cook	pos_error_xy_mag and pos_error_xy_az define relevant dispersions
Mass_Properties	Lander_Deployed	Zcg	mm	209.00	229.00	249.00	Uniform	LNDR Frame	1/12/17	Brant Cook	
Mass_Properties	Lander_Deployed	pos_error_xy_mag	mm	0	0	5	Uniform	LNDR Frame	1/12/17	Brant Cook	defines magnitude of CG position error in xy plane
Mass_Properties	Lander_Deployed	pos_error_xy_az	deg	0	0	360	Uniform	LNDR Frame	1/12/17	Brant Cook	defines orientation/clocking of interface position error in xy plane
Mass_Properties	Lander_Deployed	Ixx	kg*mm^2	53721005.22	72595953.0	91470900.78	Uniform	LNDR Frame	1/12/17	Brant Cook	
Mass_Properties	Lander_Deployed	Iyy	kg*mm^2	50033129.38	67612337.0	85191544.62	Uniform	LNDR Frame	1/12/17	Brant Cook	
Mass_Properties	Lander_Deployed	Izz	kg*mm^2	71013829.9	95964635.0	120915440.1	Uniform	LNDR Frame	1/12/17	Brant Cook	
Mass_Properties	Descent_Stage	Mass	kg	252.636	341.4	430.164	Uniform	not applicable	1/12/17	Brant Cook	Dry Descent Stage properties (dry = no fuel, no helium)
Mass_Properties	Descent_Stage	Xcg	mm		0		not applicable	DS Frame	1/12/17	Brant Cook	with 0 kg was able to balance to within 0.89mm…zeroed out for this exercise, pos_error_xy_mag and pos_error_xy_az define relevant dispersions
Mass_Properties	Descent_Stage	Ycg	mm		0		not applicable	DS Frame	1/12/17	Brant Cook	with 6.5 kg was able to balance to within 0.015mm…zeroed out for this exercise, pos_error_xy_mag and pos_error_xy_az define relevant dispersions
Mass_Properties	Descent_Stage	Zcg	mm	157.45	177.45	197.45	Uniform	DS Frame	1/12/17	Brant Cook	
Mass_Properties	Descent_Stage	pos_error_xy_mag	mm	0	0	5	Uniform	DS Frame	1/12/17	Brant Cook	defines magnitude of CG position error in xy plane
Mass_Properties	Descent_Stage	pos_error_xy_az	deg	0	0	360	Uniform	DS Frame	1/12/17	Brant Cook	defines orientation/clocking of interface position error in xy plane

Other things we do:

- Randomly sample from a set of files by file name (usually Uniform weighting)
  - Atmospheric quantities, Terrain quantities, etc.

Example: Consequence of Treating Ignorance as Variability

Background on Uncertainty Types

Seminal paper: Ferson & Ginzburg, 1996

- If we have an uncertain input, and we only know an interval, it is typical (and incorrect!) to assign a uniform distribution and include it in sampled set of inputs for running the model
- Ferson & Ginzburg show with a simple multiplication of two variables that the result is misleading/wrong using that machinery

$$A = [0.2, 0.4] = \{\, A \in \mathbb{R} \,|\, 0.2 \le A \le 0.4 \,\}$$

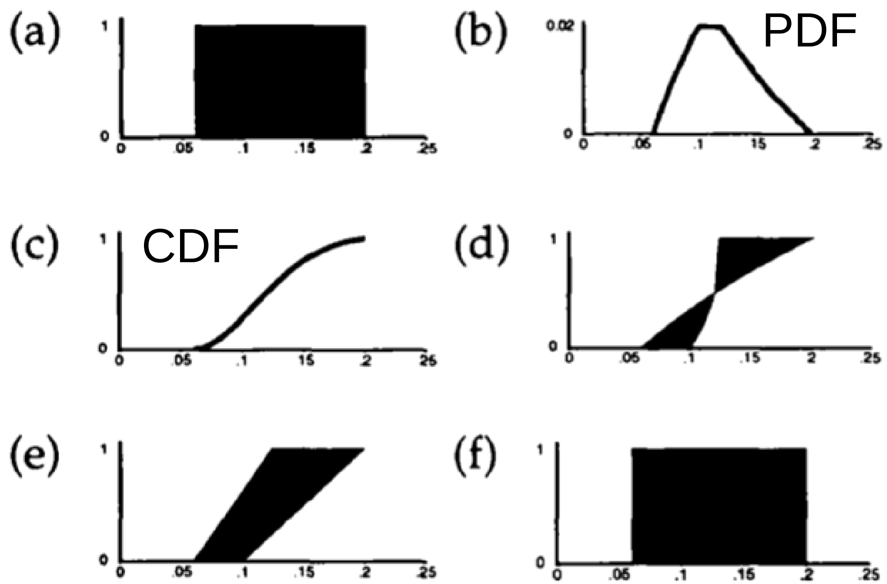
$$B = [0.3, 0.5] = \{\, B \in \mathbb{R} \,|\, 0.3 \le A \le 0.5 \,\}$$

$$AB = ?$$

- Common sense tells us AB = [0.06, 0.2], but…

Differing results under differing assumptions

Source: Ferson & Ginzburg



Interval guaranteed to contain product AB (intuition)

(b) and (c) are result of applying simple “Monte Carlo“ machinery.

CDF of (b)

Smallest region for (c) Assuming correlation Between -1 to +1

Smallest region for (c) Not restricting correlation

Smallest region to contain (c) given no other info. (same as (a))

Illustrates that ignorance should be treated differently than variability to produce a **region** of CDFs instead of a single CDF, which can be radically different under different assumptions.

If we do not know anything about A and B other than the interval, we need machinery and analytical techniques that produce (f)

“Aleatory” vs. “Epistemic” uncertainty

Source: Roy & Oberkampf 2011

- Aleatory uncertainty:

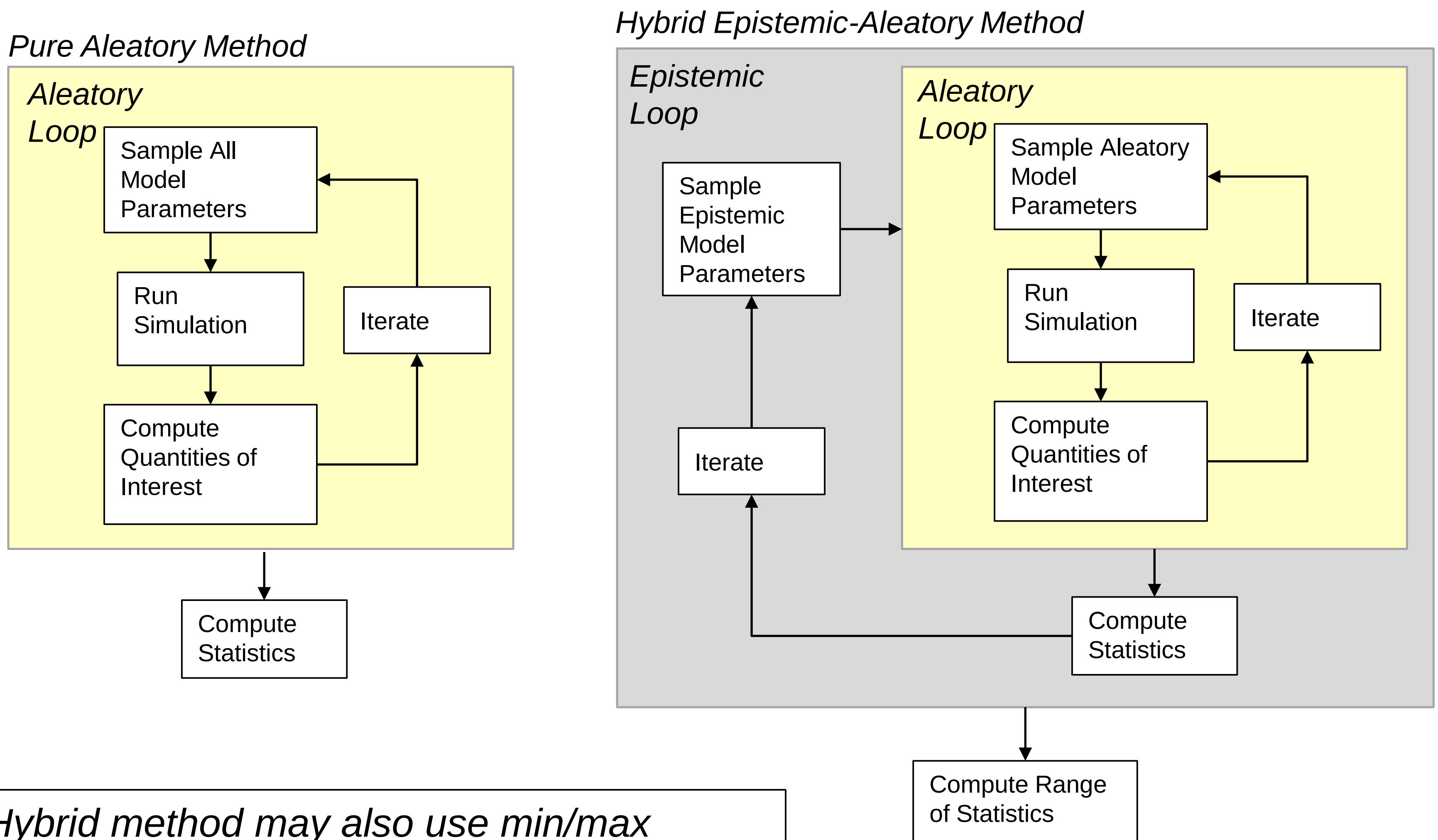
  Aleatory uncertainty (also called irreducible uncertainty, stochastic uncertainty, or variability) is uncertainty due to inherent variation or randomness and can occur among members of a population or due to spatial or temporal variations. Aleatory uncertainty is generally characterized by either a probability density function (PDF) or a cumulative distribution function (CDF).

- Epistemic uncertainty:

  Epistemic uncertainty (also called reducible uncertainty or ignorance uncertainty) is uncertainty that arises due to a lack of knowledge on the part of the analyst, or team of analysts, conducting the modeling and simulation…We will represent epistemic uncertainty as an interval-valued quantity, meaning that the true (but unknown) value can be any value over the range of the interval, with no likelihood or belief that any value is more true than any other value.

So how do we include Epistemic inputs (ignorance)?

Hybrid Epistemic-Aleatory UQ



Hybrid method may also use min/max optimization to search for range of statistics.

A more complete/generic input deck

Unique ID

Classification/ Organization

Parameter Name

Units

Assoc. Frame

Undispersed Nominal Value

Uncertainty Type

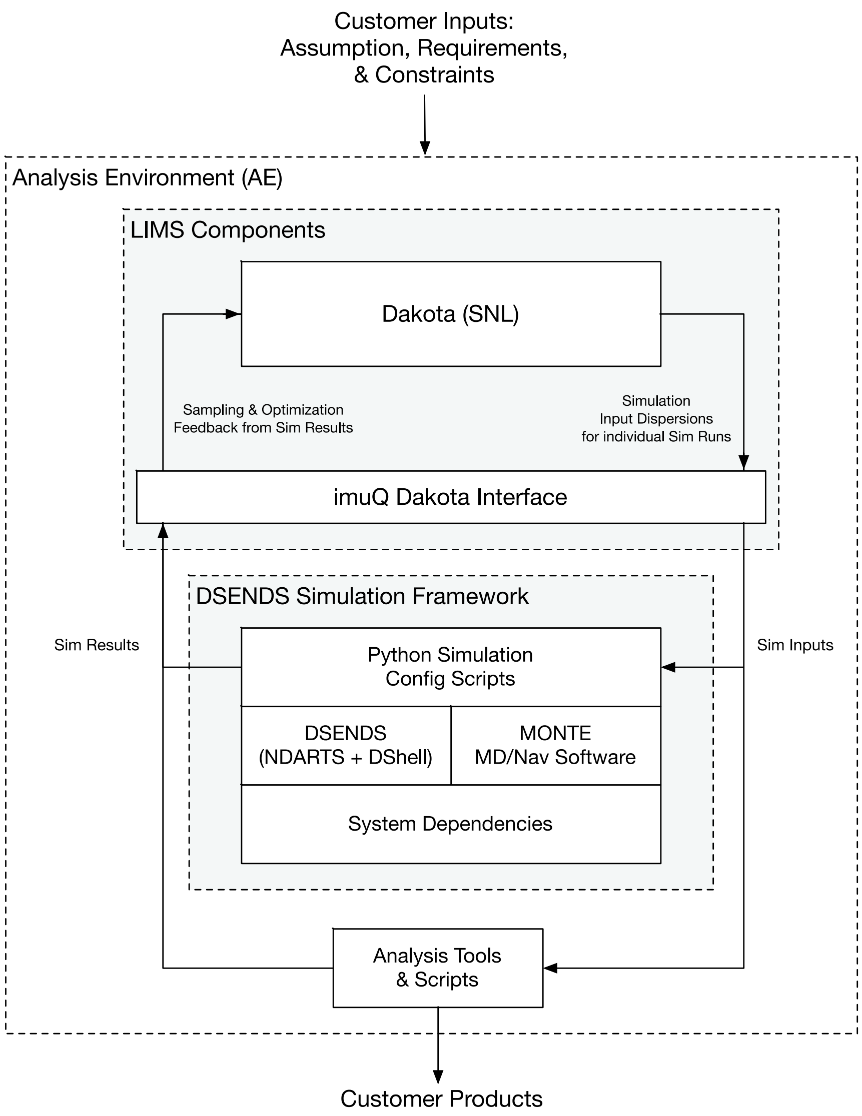
Parameterization values (Depend on Uncertainty Type)

Revision Control Information

ID	Category	Assembly	Parameter	Units	Frame	Nominal	Uncertainty Type	P1	P2	P3	P4	P5	P6	Last Update	Source	Comments
0	Mass_Properties	Lander_Deployed	Mass	kg	NA	377.6	Epistemic	279.424	475.776					2/15/18	B. Cook	Lander properties in open (legs fully deployed) SkyCrane configuration
1	Mass_Properties	Lander_Deployed	Xcg	mm	LNDR	0	Circ_Uniform	2	5					2/15/18	B. Cook	correlated to Circ_Uniform result with Ycg
2	Mass_Properties	Lander_Deployed	Ycg	mm	LNDR	0	Circ_Uniform	1	5					2/15/18	B. Cook	correlated to Circ_Uniform result with Xcg
3	Mass_Properties	Lander_Deployed	Zcg	mm	LNDR	229	Epistemic	209	249					2/15/18	B. Cook	
4	PPPCS	DescentEngine1	thrustunitvector_i	NA	DS	-0.5	Constant							12/9/16	Brant Cook	vector given is direction of resultant thrust
5	PPPCS	DescentEngine1	thrustunitvector_j	NA	DS	0	Constant							12/9/16	Brant Cook	vector given is direction of resultant thrust
6	PPPCS	DescentEngine1	thrustunitvector_k	NA	DS	-0.866	Constant							12/9/16	Brant Cook	vector given is direction of resultant thrust
7	PPPCS	DescentEngine1	location_x	mm	DS	790.85	Circ_Uniform	8	2					12/9/16	Brant Cook	location of nozzle throat
8	PPPCS	DescentEngine1	location_y	mm	DS	591.03	Circ_Uniform	7	2					12/9/16	Brant Cook	location of nozzle throat
9	PPPCS	DescentEngine1	location_z	mm	DS	519.42	Uniform	515.42	523.42					12/9/16	Brant Cook	location of nozzle throat

DSENDS + LIMS Analysis Environment @ JPL

- Group 3436 historically uses DSENDS & MONTE to generate Monte-Carlo data for analysis of EDL & ProxOps
- J. Benito & C. Noyes began exploring the use of “Dakota” from Sandia National Labs as part of the MAV task for 6x
- Inspired by Benito & Noyes’ work, Group 3436 is now hooking up DSENDS to the imuQ and Dakota from L. Peterson LIMS packagein 35x to enable interrelated UQ/SA/Optimization consistent with the wider scientific computing V&V community
  - The construct on previous slide is one of many this machinery can do.
- **Customers don’t know what to ask for (yet)… hence these talks!**



Example

A dynamics example: 1D projectile in gravity

Simple model to demonstrate methodology w/ credible results

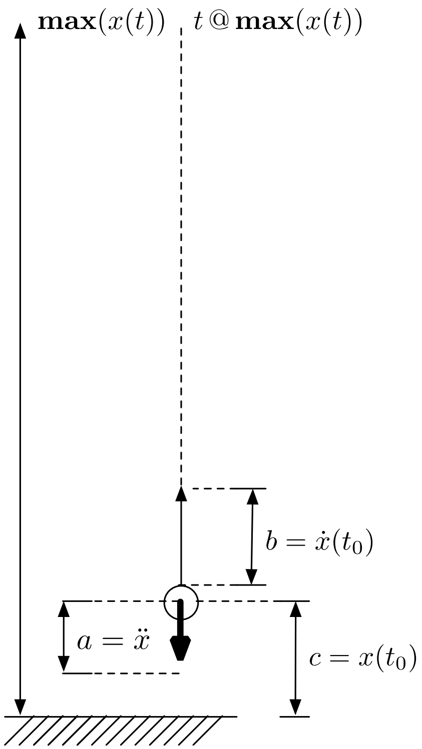
- Simple problem: 1D acceleration field on particle with initial conditions (a vertical projectile)
- Physics: $\ddot{x} = -a \qquad \dot{x}(t_0) = b \qquad x(t_0) = c$

$$x(t) = -\frac{1}{2}at^2 + bt + c$$

- a, b, and c are input parameters to be dispersed.
- System Response Quantities (SRQs) we care about:

max(**x**(t)) and t @ max(**x**(t))

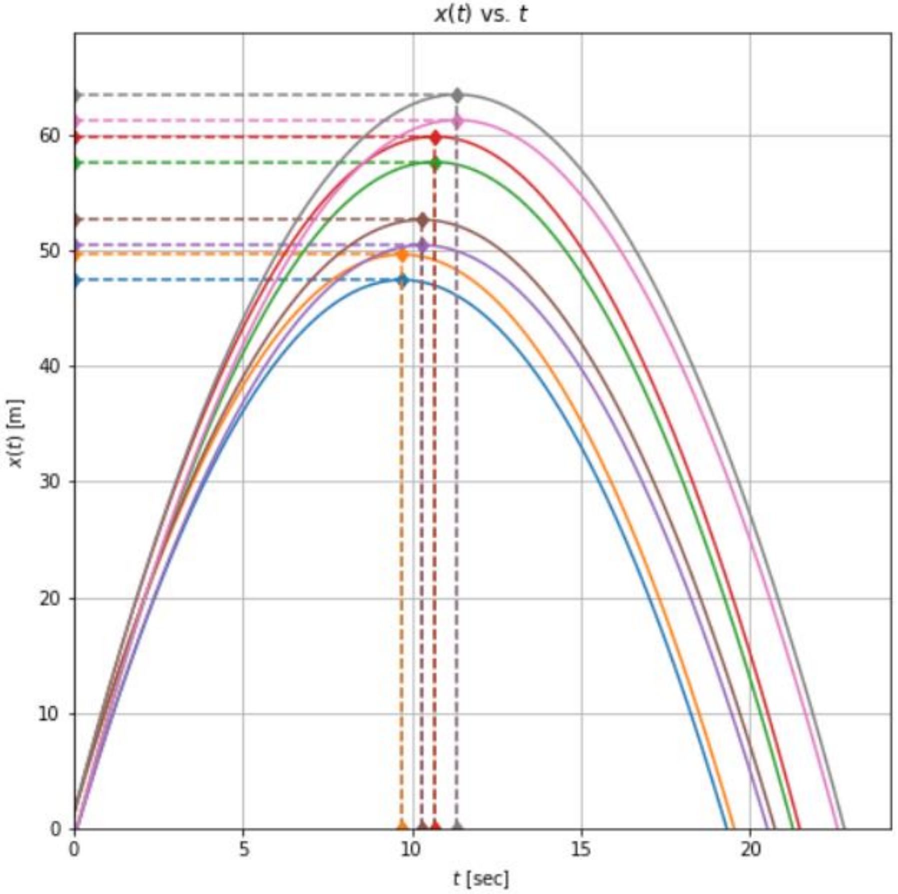
- Traditionally, we do "monte carlo", assign all parameters a distribution and "roll the dice" to get a performance.



SRQs illustrated w/ interval corner cases

This problem has easy sanity check for range of SRQ results

Expected range
of heights
(SRQ #1)



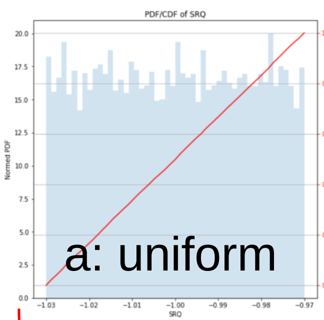
Expected range
of time to max height
(SRQ #2)

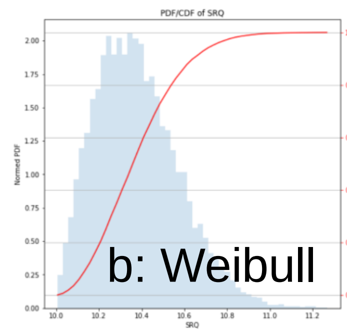
max altitude range is 47.44 to 63.47 [m]
time to max altitude range is 9.71 to 11.34 [s]

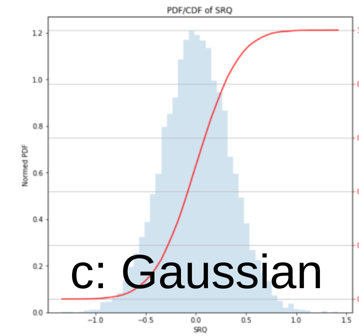
Traditional “Monte Carlo” results

treating a, b, and c as parameters with known variability

Inputs

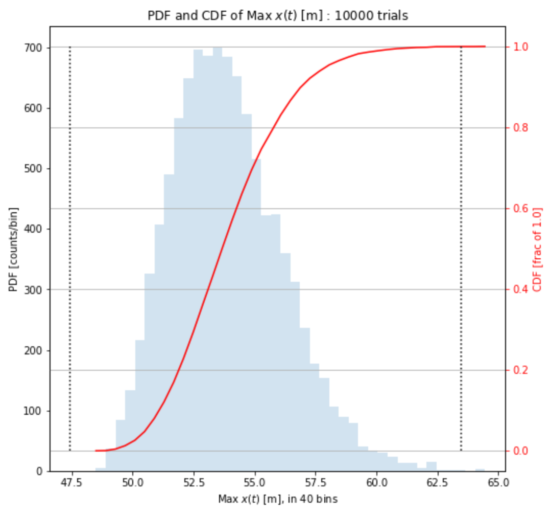




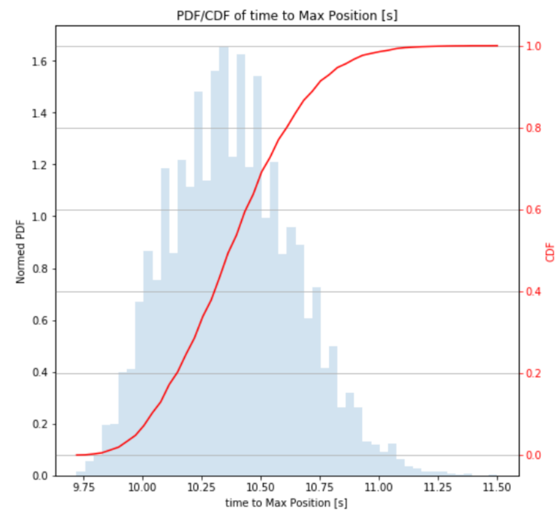


Outputs

max(x)



t @ max(x)



ONE CDF per SRQ

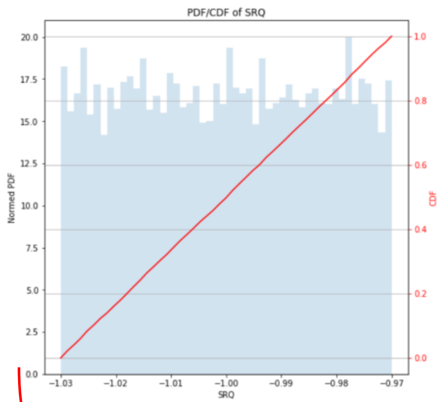
Regarding Monte Carlo

- MC is adequate only if all inputs are truly aleatory
  - Needs appropriate sampling to capture assessment percentiles

- What to do if there isn't a high confidence in input distributions
  - Treat inputs w/ ignorance as epistemic: sample epistemic inputs across their expected range, and vary ALL OTHER ALEATORY INPUTS at EVERY PERMUTATION of epistemic samples (mini MC at each epistemic parameter permutation)
  - This essentially combines "worst case" analysis and uncertainty quantification to produce a range of CDFs for System Response Quantities (SRQs)
  - Can discover where worst cases are not at the limits of the range of epistemic inputs

What if: b doesn't have any data to back up PDF?

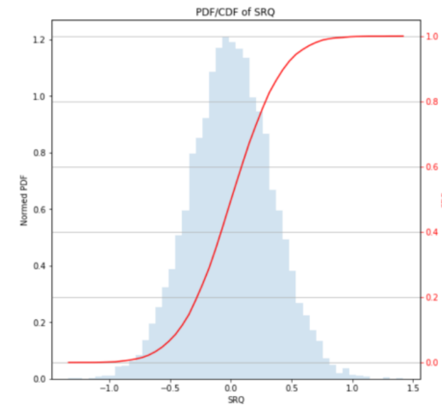
a: uniform



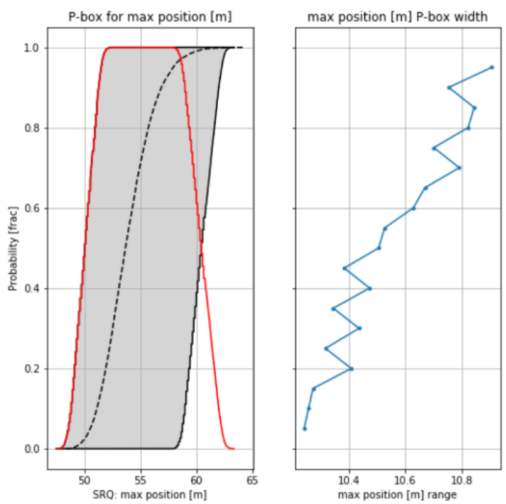
b: Epistemic

$$b_{min} \leq b \leq b_{max}$$

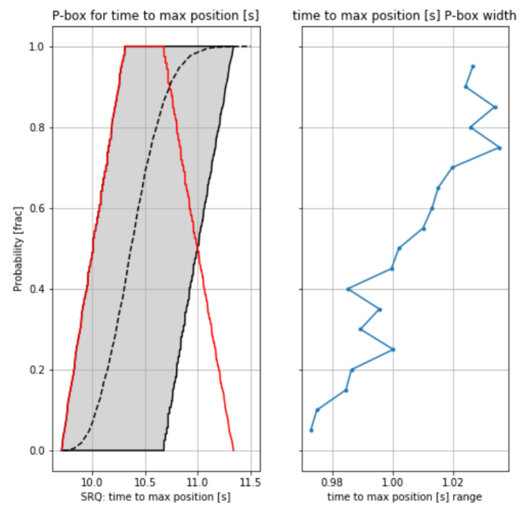
c: Gaussian



max(x)



t @ max(x)



Produces a RANGE OF CDFs for each SRQ: a.k.a. "p-boxes"

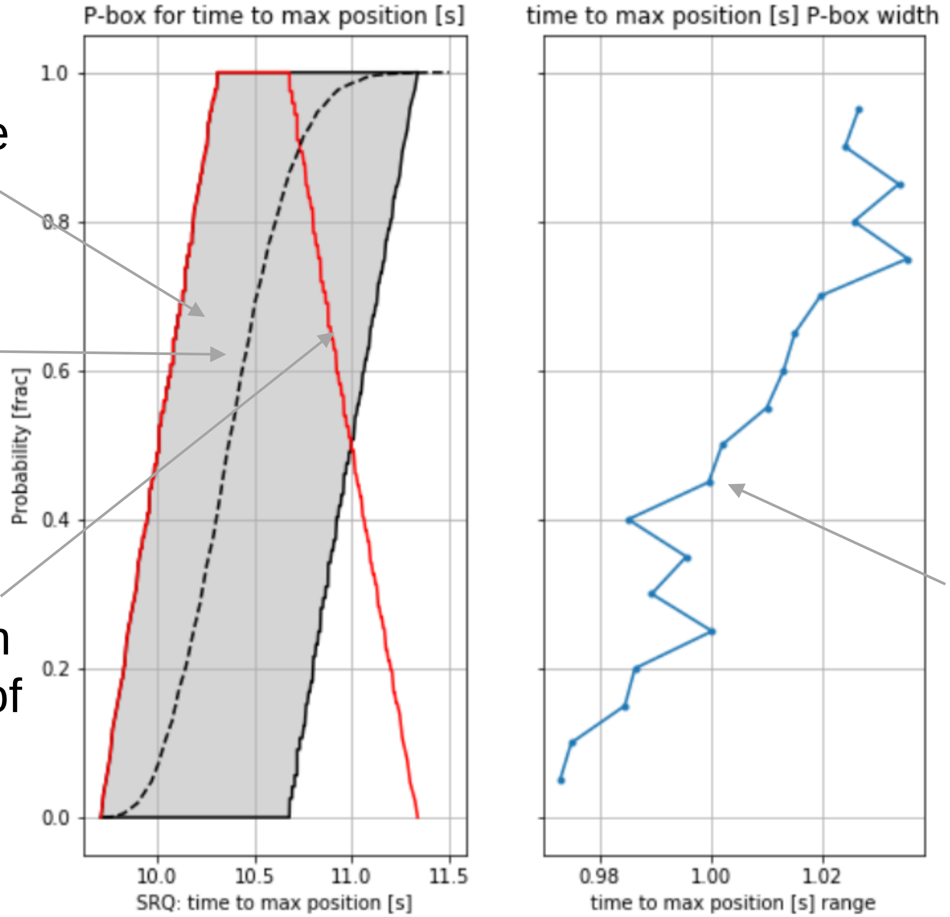
Interpreting a P-Box (a.k.a. “horsetail”) plot

SRQ: t @ max(x)

Region where
Actual CDF may lie
(grey area)

All Aleatory result
For comparison
(dashed line)

Possible range of
Percentiles for given
SRQ value (height of
Shaded area)



Possible range SRQ
Values for a given
Percentile (width of
Shaded area)

“Being honest” means including ignorance appropriately in UQ analysis

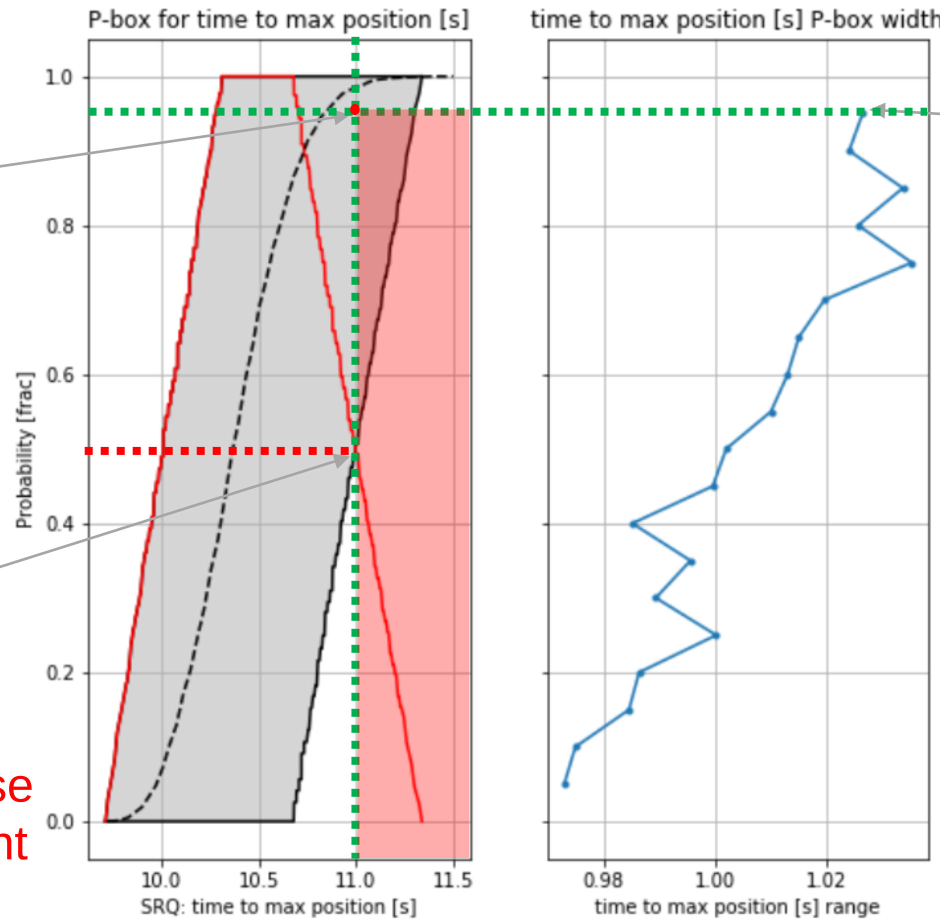
- We can now answer questions like:
  - “What is the range of SRQ values correspond to a particular percentile?”
  - “What range of percentiles correspond to a particular SRQ value?”
- Requirements can now be assessed relative to ignorance in the input parameters (where there is not a valid statistical model).
- **Absence of a statistical model for an input does not justify a “uniform” distribution – this may not be conservative**.
  - Worst case SRQ can be at a small part, even a single value, of the epistemic range(s)
  - Worst case SRQ may not be at limits of epistemic variable range

Comparing against a Requirement/Limit/etc.

SRQ: t @ max(x)

Requirement:
SRQ < 11.0 sec
@ P95

Percentile range
(uncertainty) at
Specified Metric:
P50-P100 (50%!)

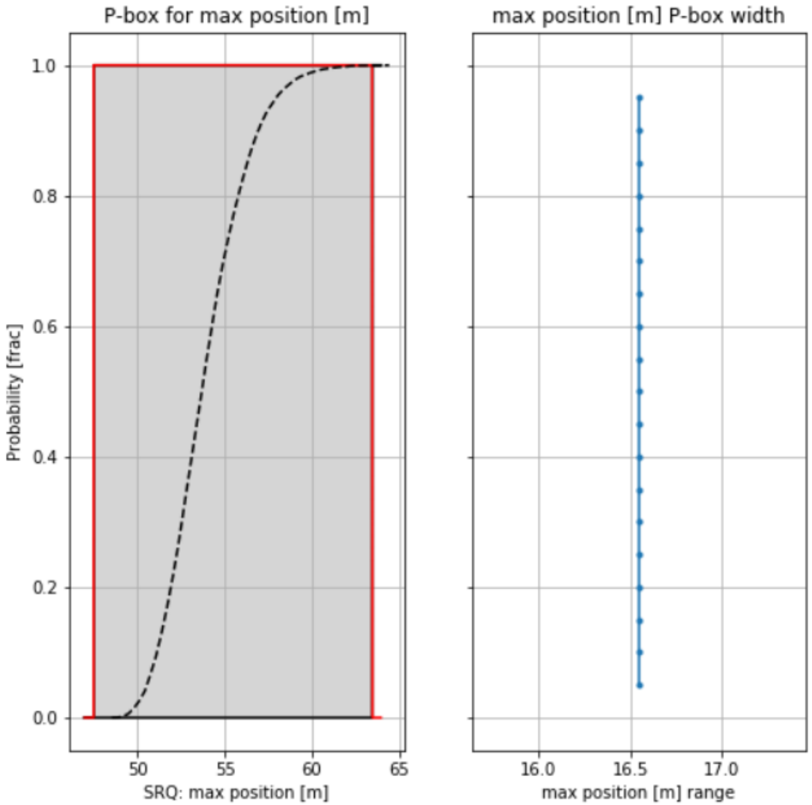


SRQ interval
at P95: 1.025 s

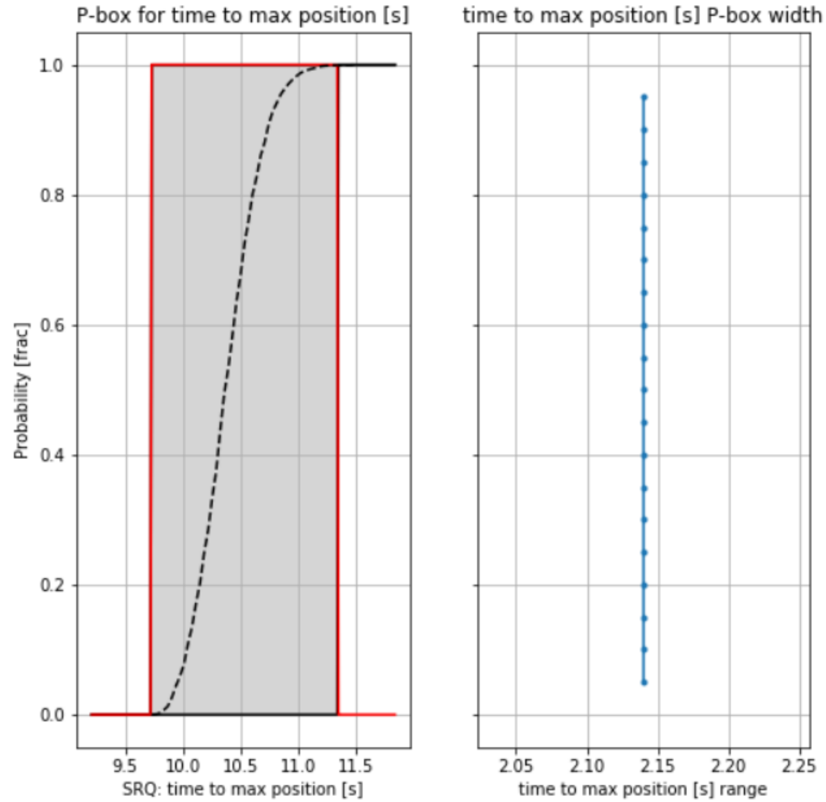
Note: All aleatory
analysis in this case
passes requirement

All Epistemic? Does it match intuition?

SRQ 1: max height



SRQ 2: time to max height



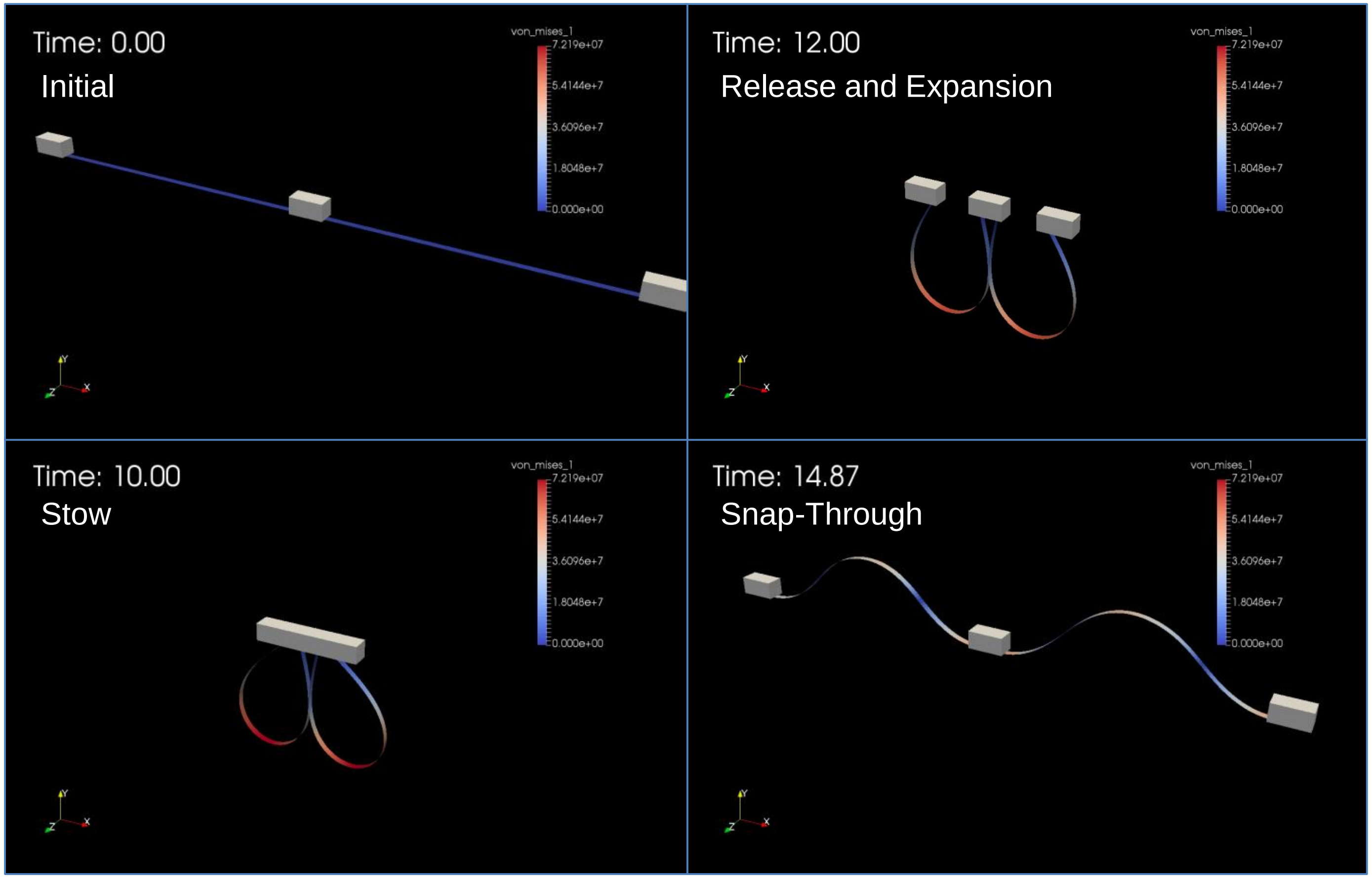
max altitude range is 47.44 to 63.47 [m]

time to max altitude range is 9.71 to 11.34 [s]

Matches corner-case intuition check.

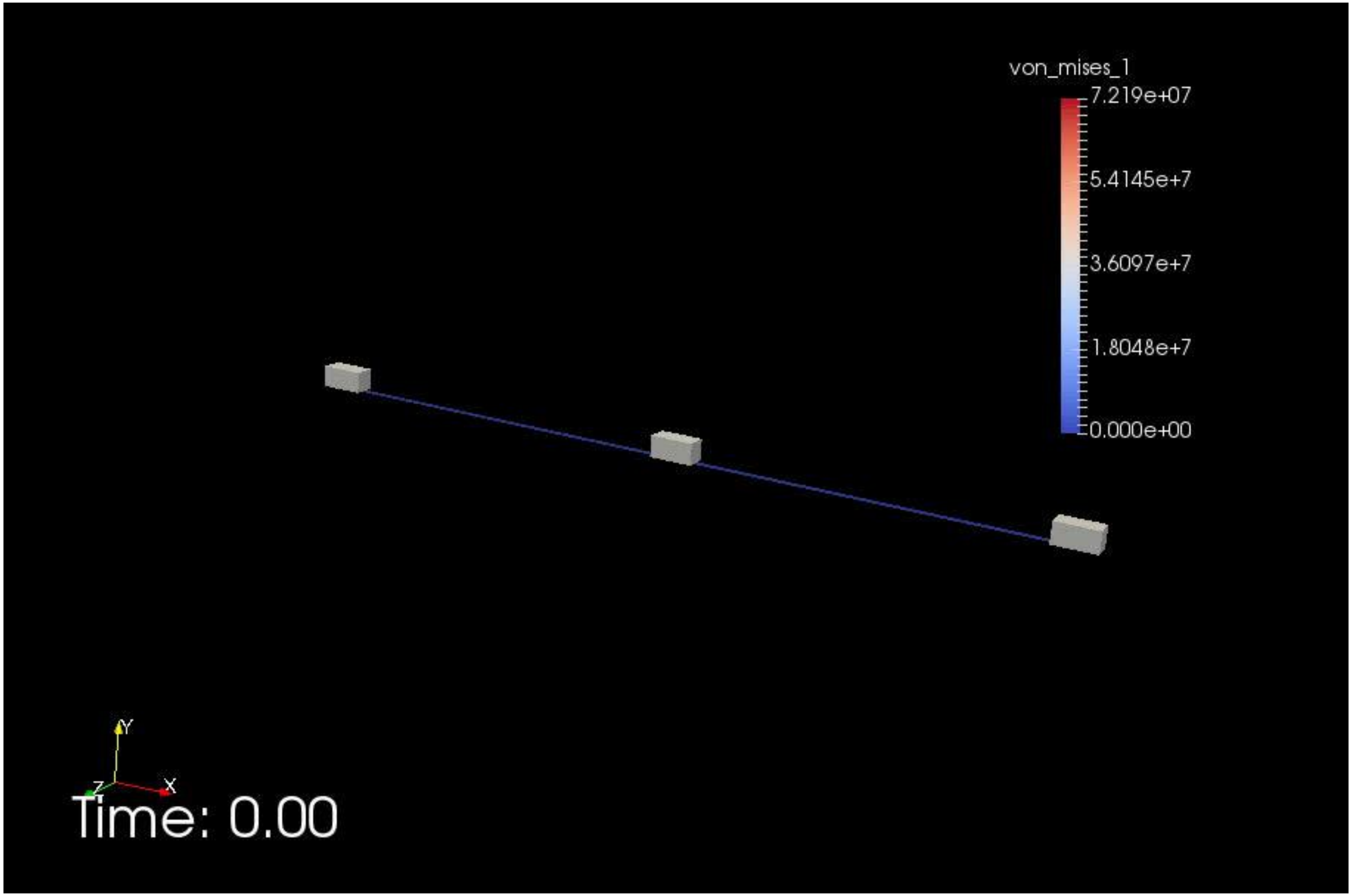
Dynamic snatch load example (L. Peterson)

Two-Strap Benchmark Problem



- Proved to be most challenging benchmark (buckling, instability, soft contact)

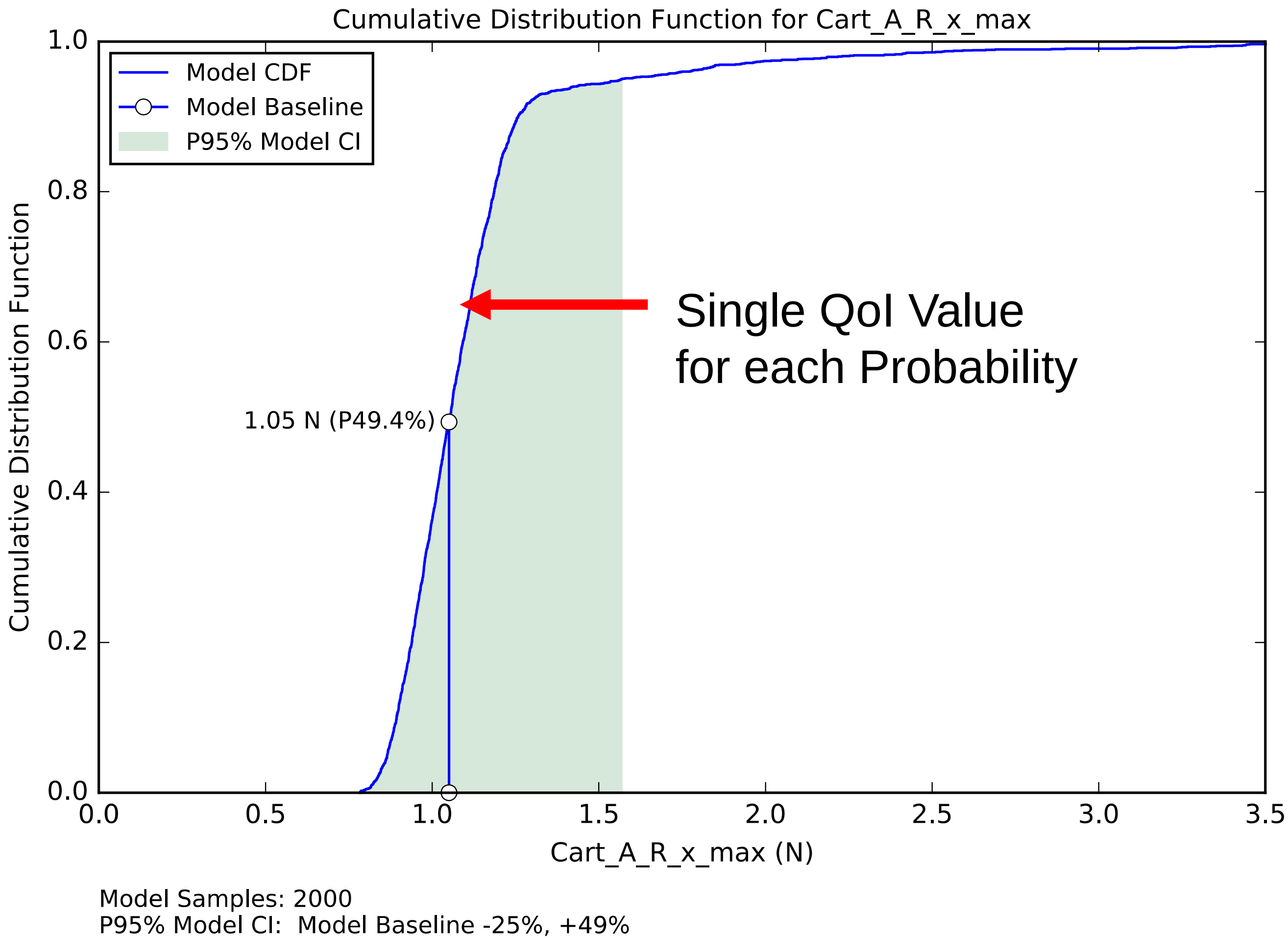
Two-Strap Benchmark Animation



Two-Strap Benchmark UQ Implementation

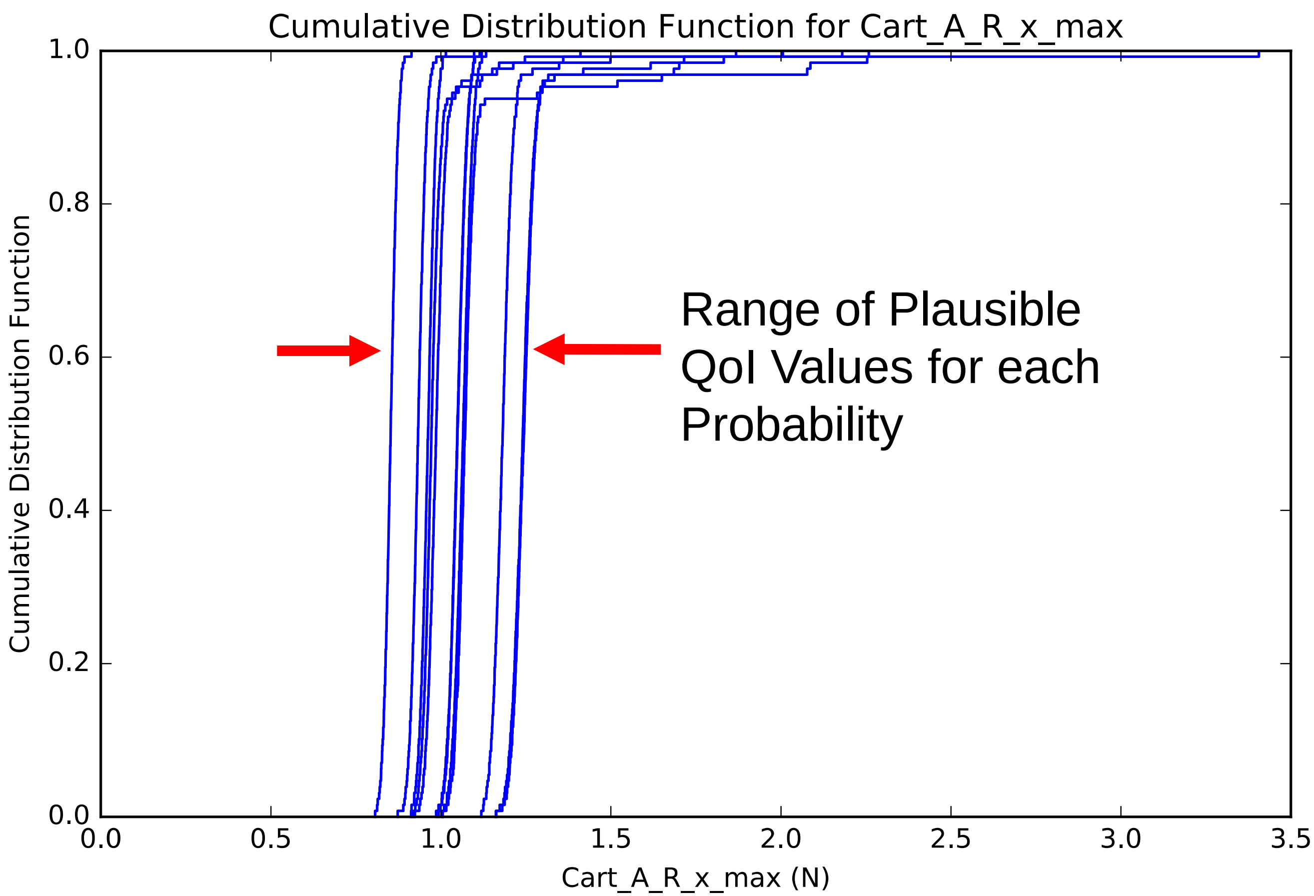
- Quantities of Interest (QoI/SRQ):
  - X and Y reaction forces on the left-most cart
- Epistemic Variabilities (3):
  - Strap-Strap Friction Coefficient
    - Interval-valued 0.1 to 0.8
  - Strap Material Young's Modulus
    - Interval-valued -20% to +20%
  - Strap Thickness
    - Interval-valued -20% to +20%
- Aleatory Variabilities (5):
  - Cart Material Properties
    - Gaussian, 3% standard deviation, +/-30% bound
  - Strap Density and Poisson's Ratio
    - Gaussian, 3% standard deviation, +/-30% bound

Two-Strap Monte Carlo UQ Results (Rx) *LHS (2000)*



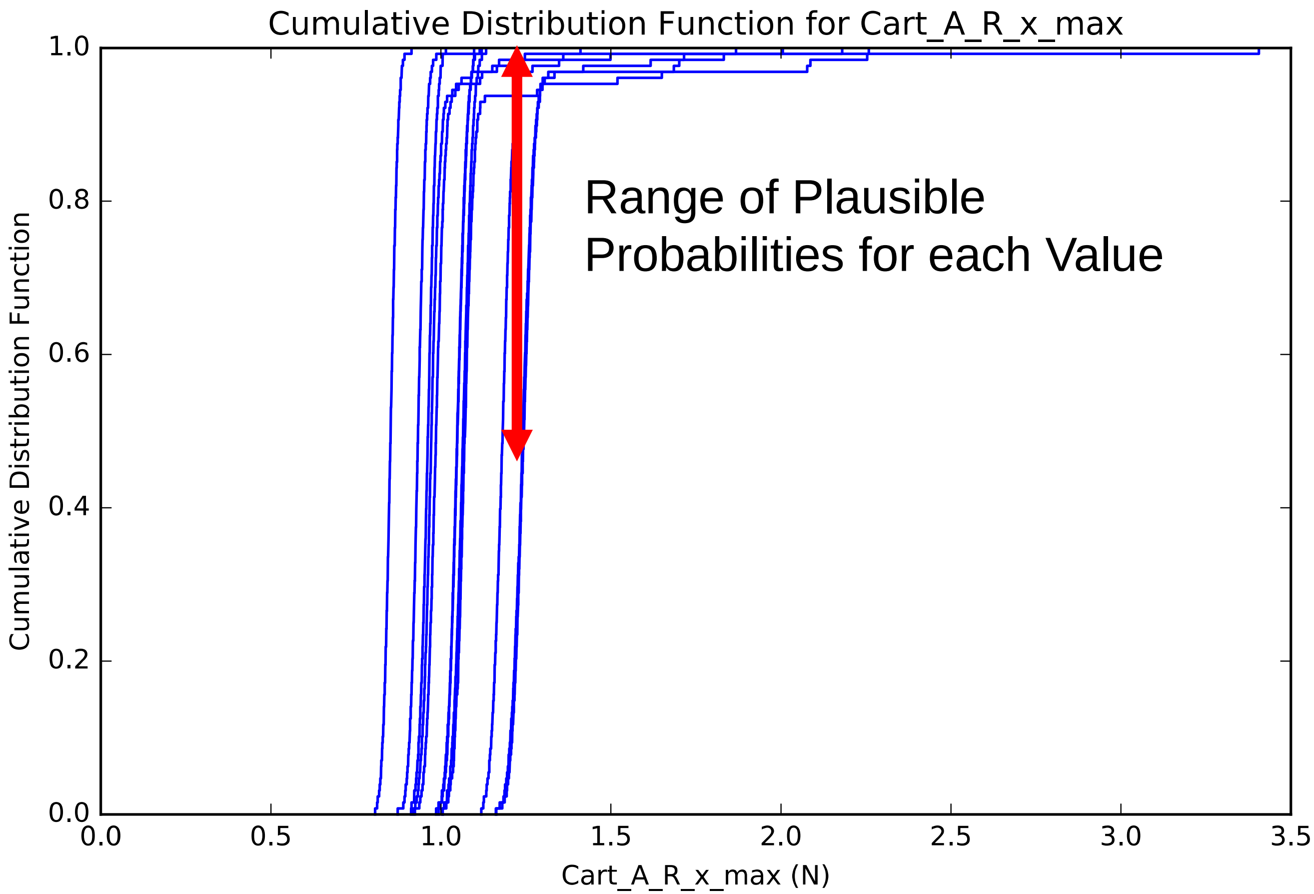
CDF Plot: Probability Model QoI is < x

Two-Strap Epistemic-Aleatory UQ Results (Rx)
IVP: LHS (12) x LHS (128)



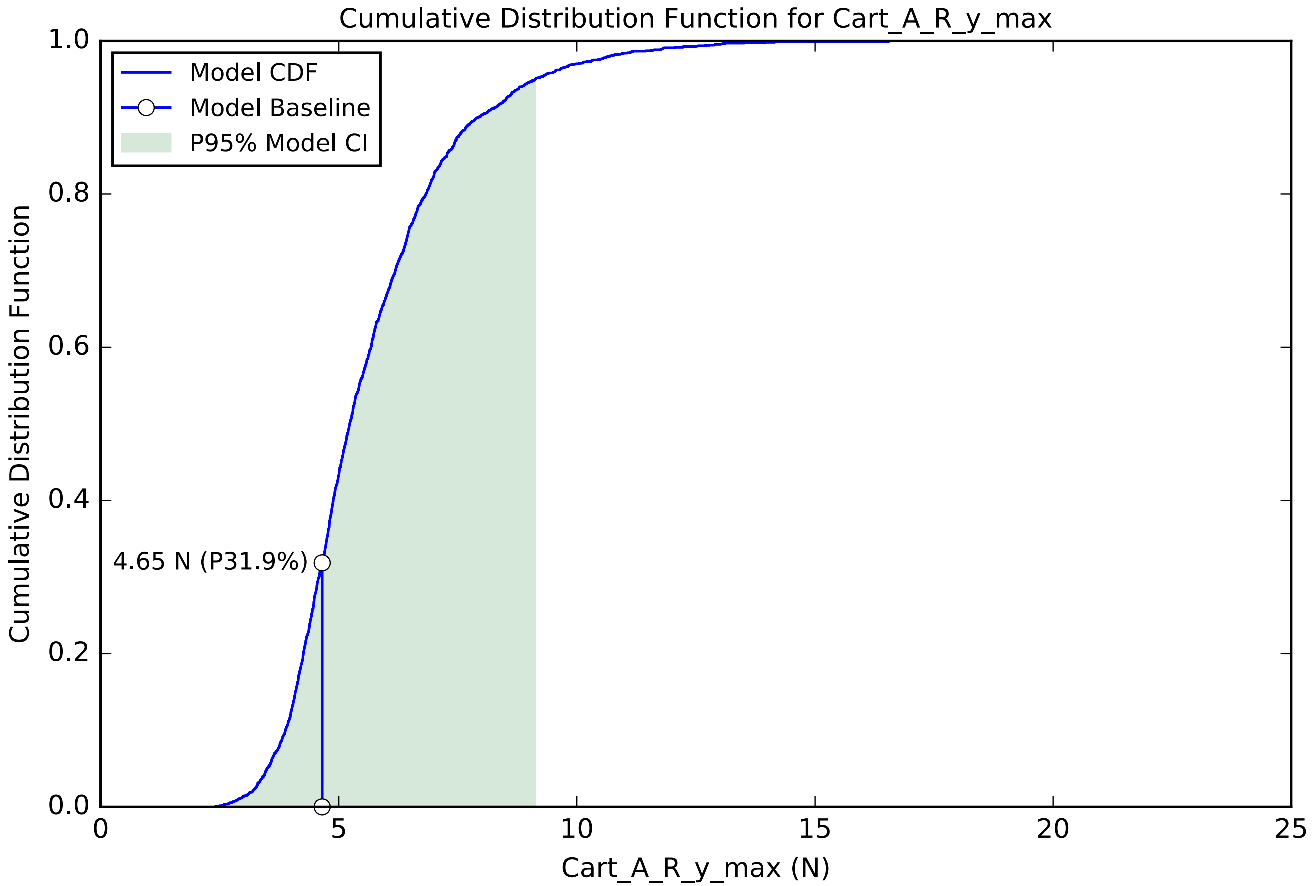
"Horsetail Plot: CDF for each outer loop (i.e. given set of epistemic values)

Two-Strap Epistemic-Aleatory UQ Results (Rx)
IVP: LHS (12) x LHS (128)



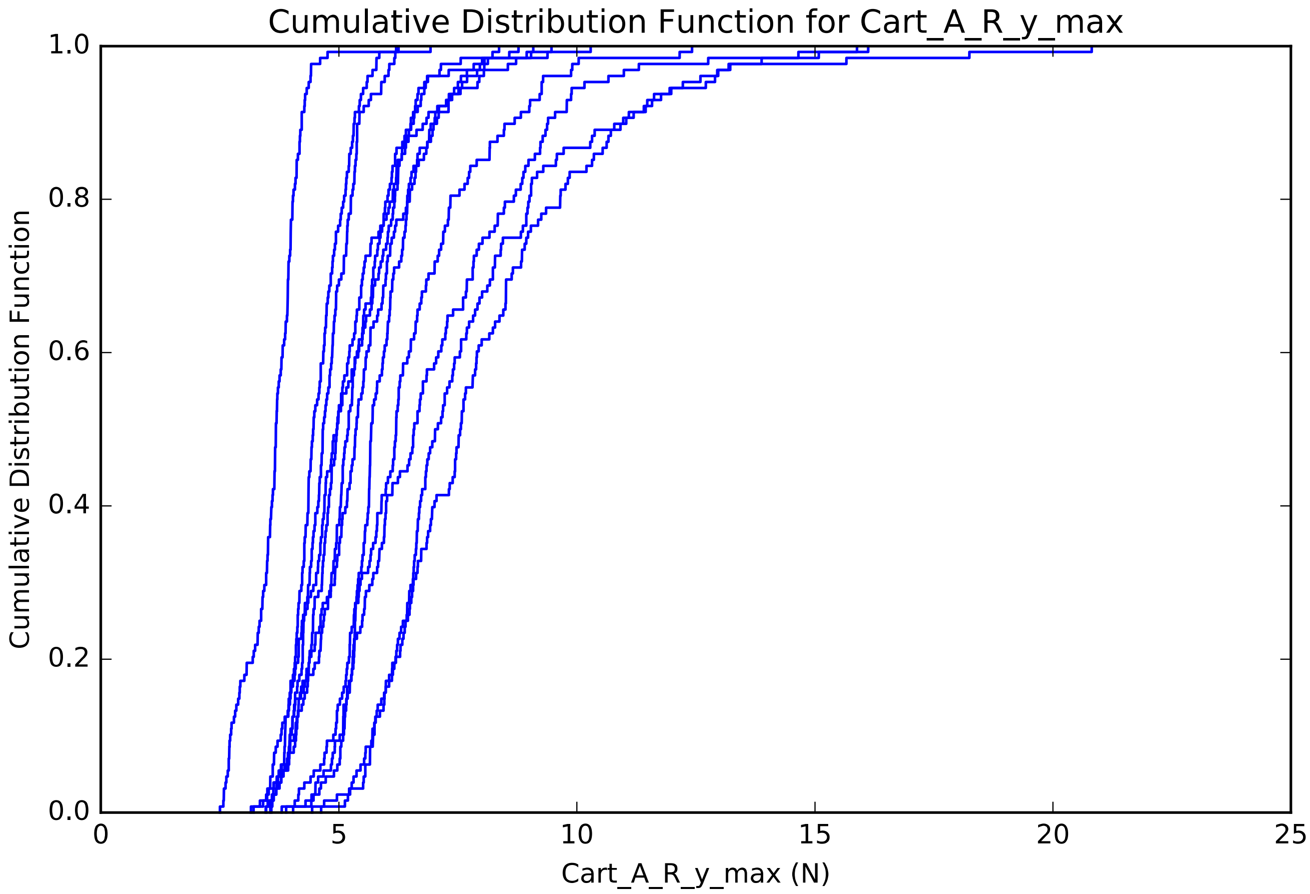
"Horsetail Plot: CDF for each outer loop (i.e. given set of epistemic values)

Two-Strap Traditional Monte Carlo UQ Results (Ry) *LHS (2000)*



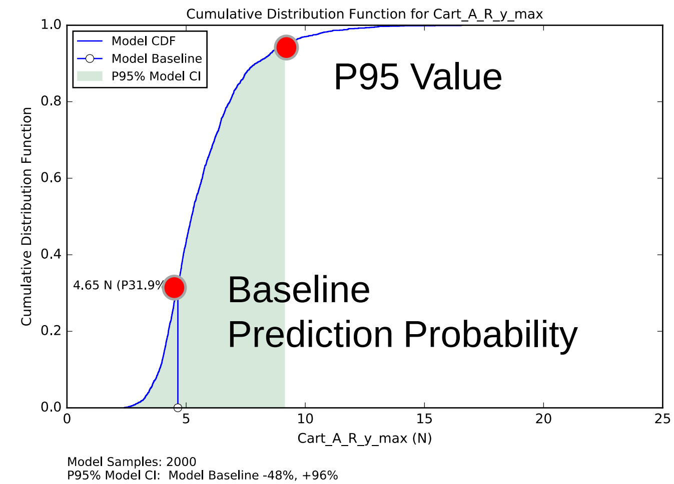
Model Samples: 2000
P95% Model CI: Model Baseline -48%, +96%

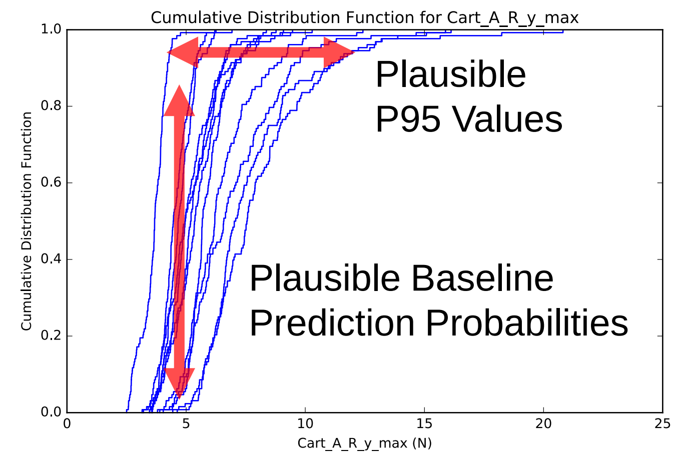
Two-Strap Epistemic-Aleatory UQ Results (Ry)
IVP: LHS (12) x LHS (128)



Comparison of Two-Strap Monte Carlo and Hybrid Epistemic-Aleatory UQ Results

- Baseline Model Prediction
  - Monte Carlo: P31
  - Hybrid: ~P0 to P100
- P95 Prediction
  - Monte Carlo: +96%
  - Hybrid: -6% to +167%
- Implications
  - Design margin would be effected by the (unknown) values of the epistemic variables
    - Opportunity for more robust designs
    - <u>and/or</u>
    - Need additional testing (calibration)

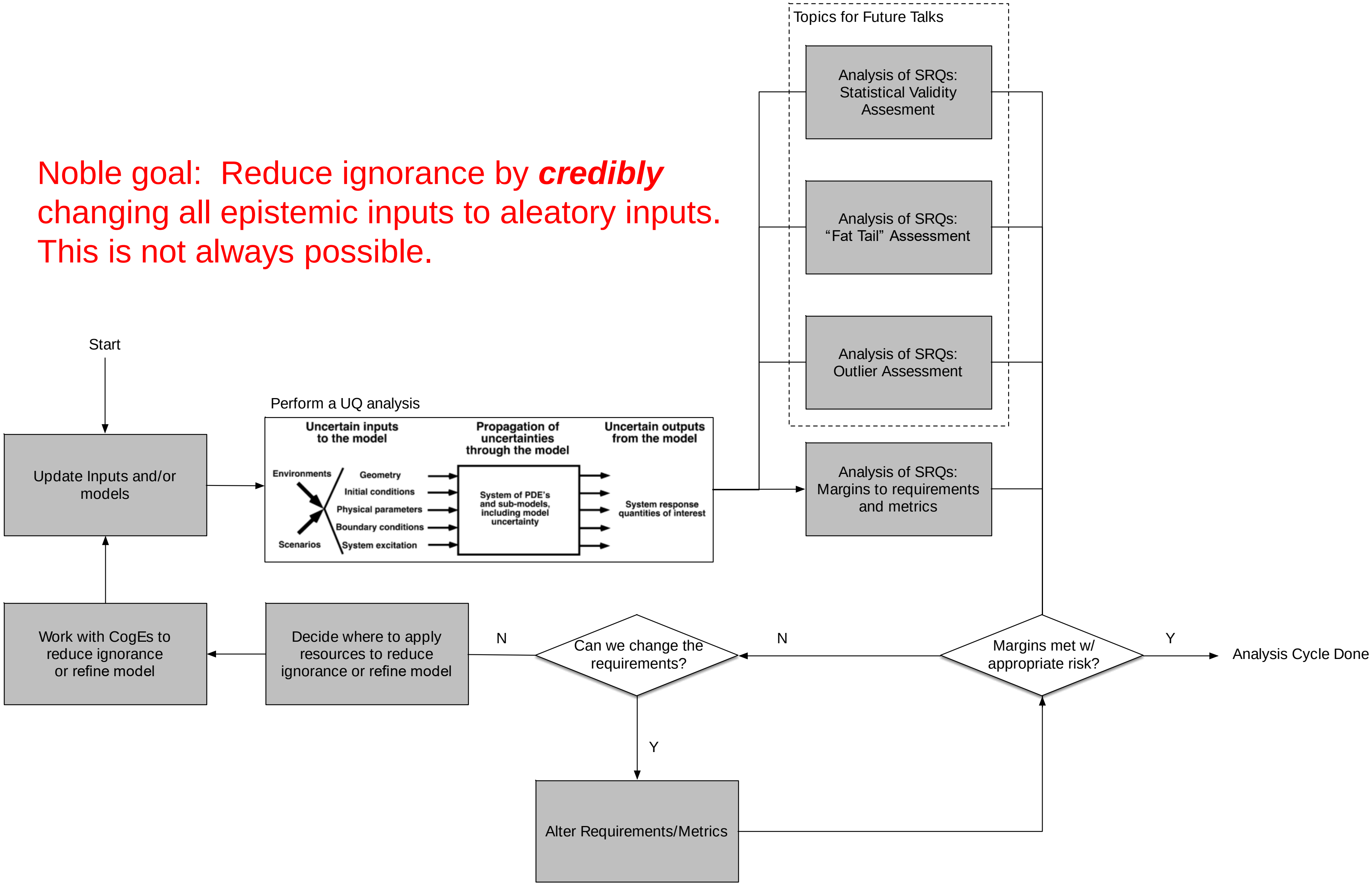




What the cycle of ignorance reduction looks like with distinction of ignorance & variability

The UQ Design/Analysis Process

Noble goal: Reduce ignorance by ***credibly*** changing all epistemic inputs to aleatory inputs. This is not always possible.



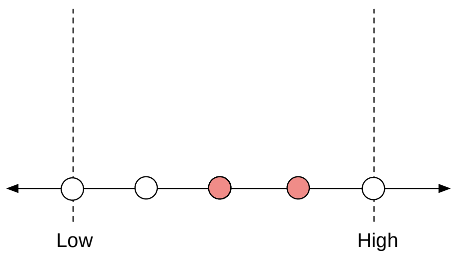
The Input Uncertainty Improvement Process

Inputs

Outputs

Cycle 1

1



Input Parameter X = Total Ignorance

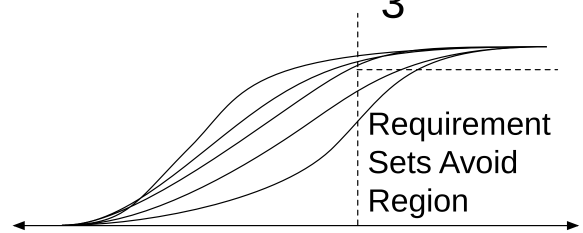
2

Hybrid Aleatory/Epistemic UQ
Creates CDFs of SRQs

Post-Processing ID's problem areas

4

3



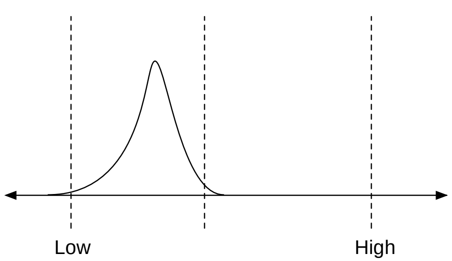
Output SRQ = violating requirement w/ X Epistemic

5

Tests performed to get
information about true
variability of input

NEW KNOWLEDGE

Cycle 2



Input Parameter X = Sampled Variability Model

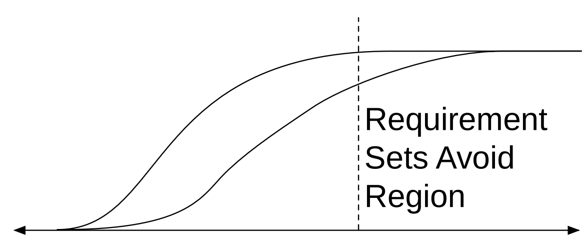
6

Hybrid Aleatory/Epistemic UQ
Creates CDFs of SRQs

Post-Processing ID's problem areas

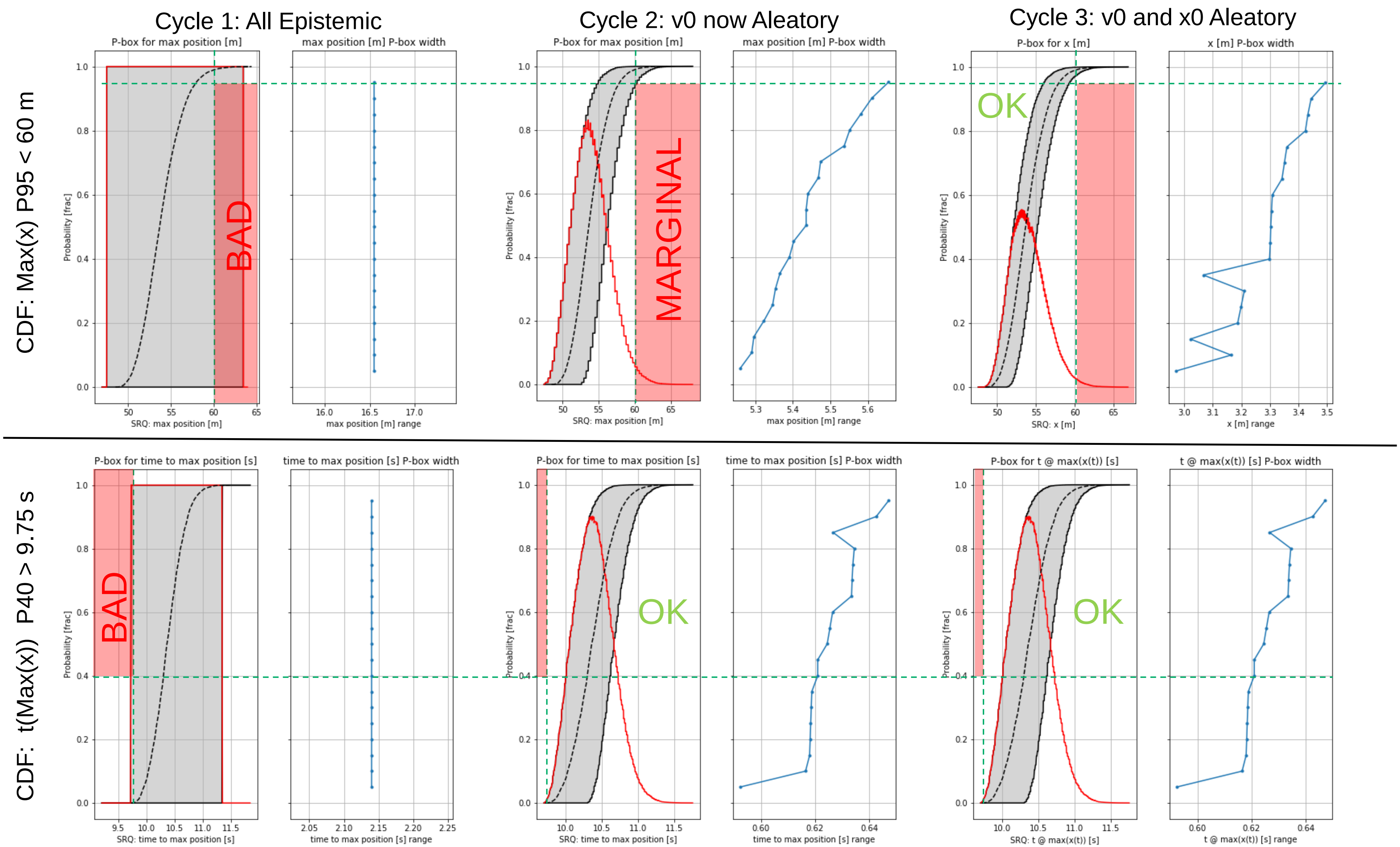
8

7



Food for thought: What are consequences of getting right-hand tail wrong (too short/thin) in Cycle 2 input model?

Results Progression Example (1DoF Problem)



Note that on previous slide, an all-variability (no ignorance) treatment passes requirements for all analytical cycles.

Relating to GN&C…

In simple 1D trajectory example:

- **What if one were to design a trigger to best capture when the projectile is at it's maximum height?**
- Techniques described here allow the analyst to combine effects of ignorance and variability for robustness assessment.
  - Could be used to justify simplification of trigger design
  - Could prove simple design isn't accurate enough given ignorance and or variability
- Because we're being honest about what we know and what we don't know, we can make decisions based on our lack of knowledge
  - Make design decisions based on more accurate representation of uncertainty ranges in SRQs
  - Invest in more testing to turn epistemic into aleatory

In the strap deployment example:

- **What if there is a closed-loop controller sensitive to IMU measurements of the forces seen from the strap deployment?**
  - The CDF variability of dynamics can now be included in robustness assessment of control scheme and decisions can be tied directly to ignorance and/or variability
  - Knowledge of dynamic event uncertainty could prevent poor decisions form being made
- Case in point: Schiaparelli failure due to IMU saturation
  - The dynamics ranges and associated modeling assumptions led analysts to believe that the high rates they occasionally saw in EDL simulations were “low-likelihood outliers” therefore saturation events were unlikely.
  - Had state-of-the-art ignorance/variability inclusion in their UQ been used, would they have thought they had adequate margins?

Recommendations

- We need to treat ignorance and variability separately in UQ
- Just because the “hammer” of Bayesian Monte Carlo machinery exists, don’t force ignorance to look like variability (a “nail”) just to use the machinery in hand – doing so obfuscates useful information for decision making purposes.
- The larger V&V community (ASME, DoE, DoD) has proven and recommended techniques of looking at these problems and techniques for managing computational burdens (advanced sampling methods, etc.)
  - See: Sandia National Labs’ DAKOTA tool as an example
- We should avail ourselves of tools/techniques/expertise to ensure that we can communicate UQ results clearly

References:

- [1] S. Ferson and L. R. Ginzburg. *Different methods are needed to propagate ignorance and variability*. Reliability Engineering and System Safety, 54:133–144, 1996.
- [2] W. L. Oberkampf and C. Roy. *Verification and Validation in Scientific Computing*. Cabridge University Press, first edition, 2010.
- [3] W. L. Oberkampf and C. Roy. *A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing*. In Comput. Methods Appl. Mech. Engrg., volume 200, pages 2131– 2144. Elsevier, 2011.
- [4] L. Peterson and M. Mobrem. *A Comparison of Uncertainty Quantification Methods on Benchmark Problems for Space Deployable Structures*. AIAA SciTech Conference, 2018.

Ongoing work in JPL Group 3436

- Adoption of imuQ + DAKOTA to include epistemic variability and multi-looped UQ analyses
- Development of Monte-Carlo Processing (MCP) to facilitate the knowledge/risk assessment process
- Modernization of simulation deployment methods to enable use of JPL CAE-provided Amazon GovCloud computing to deal with increased computation required to reduce time-to-solution
- Using non-commercial tools to ensure scalability that can track with parallel computation needs for analysis

Thank you