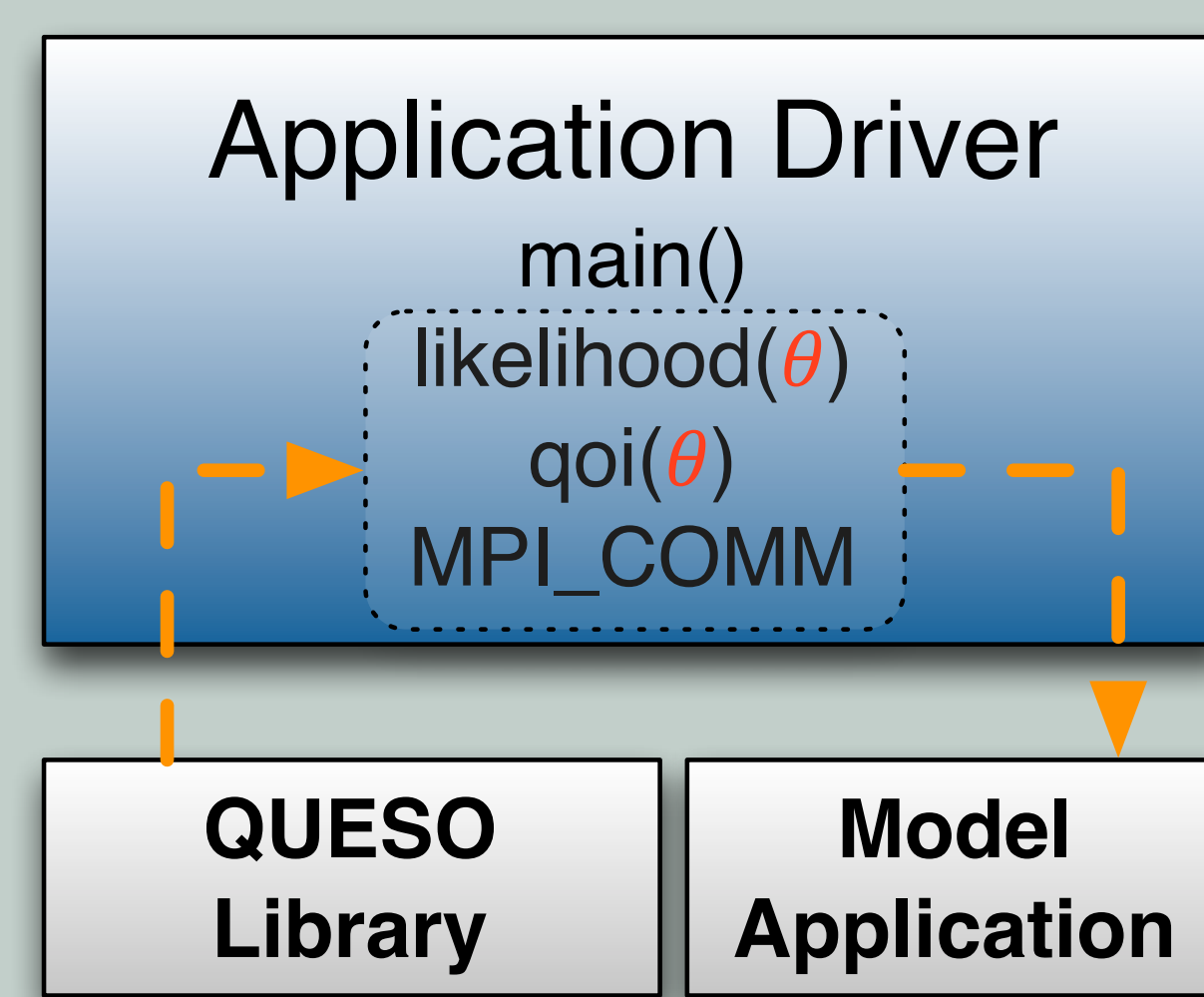
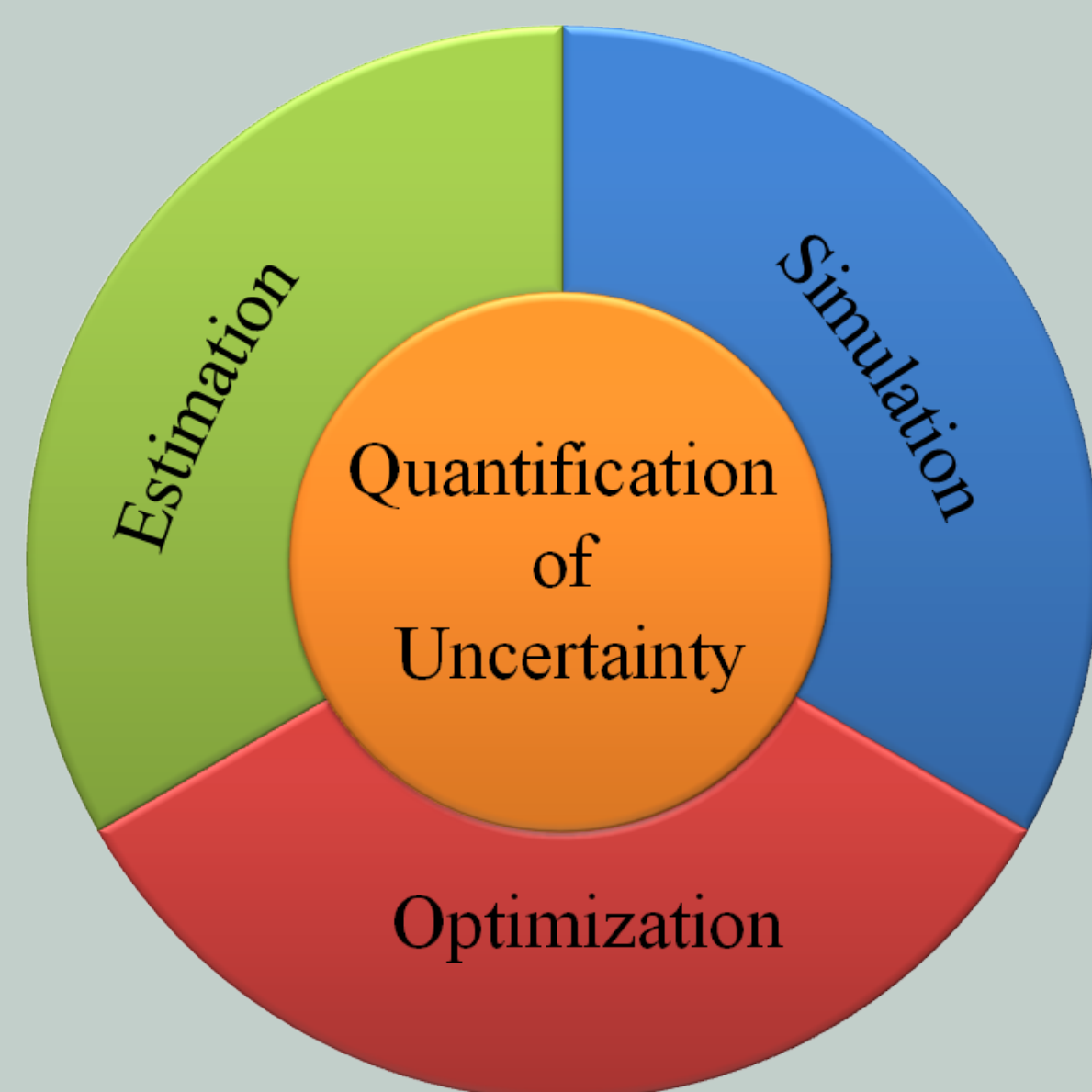


Introduction

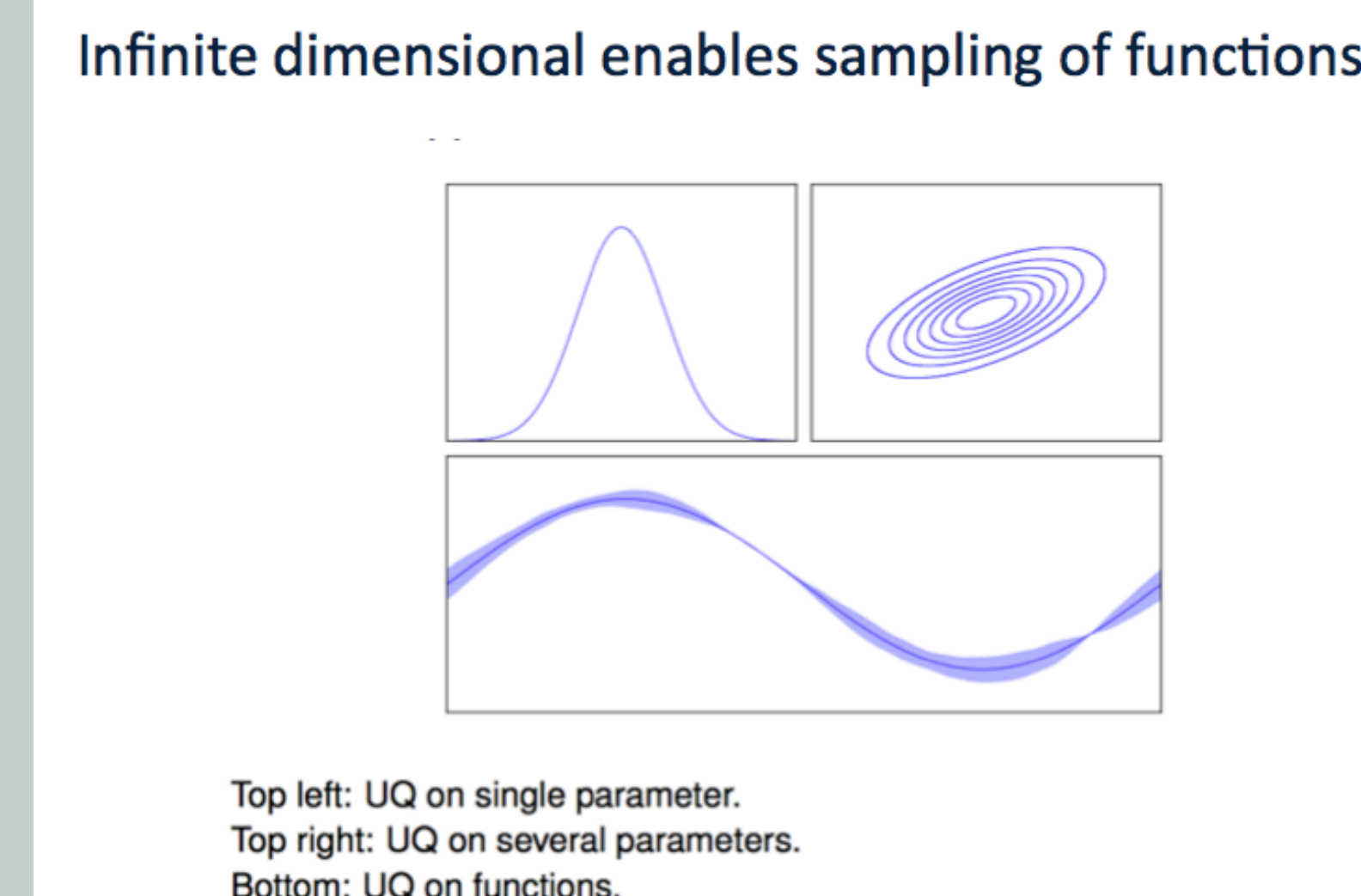
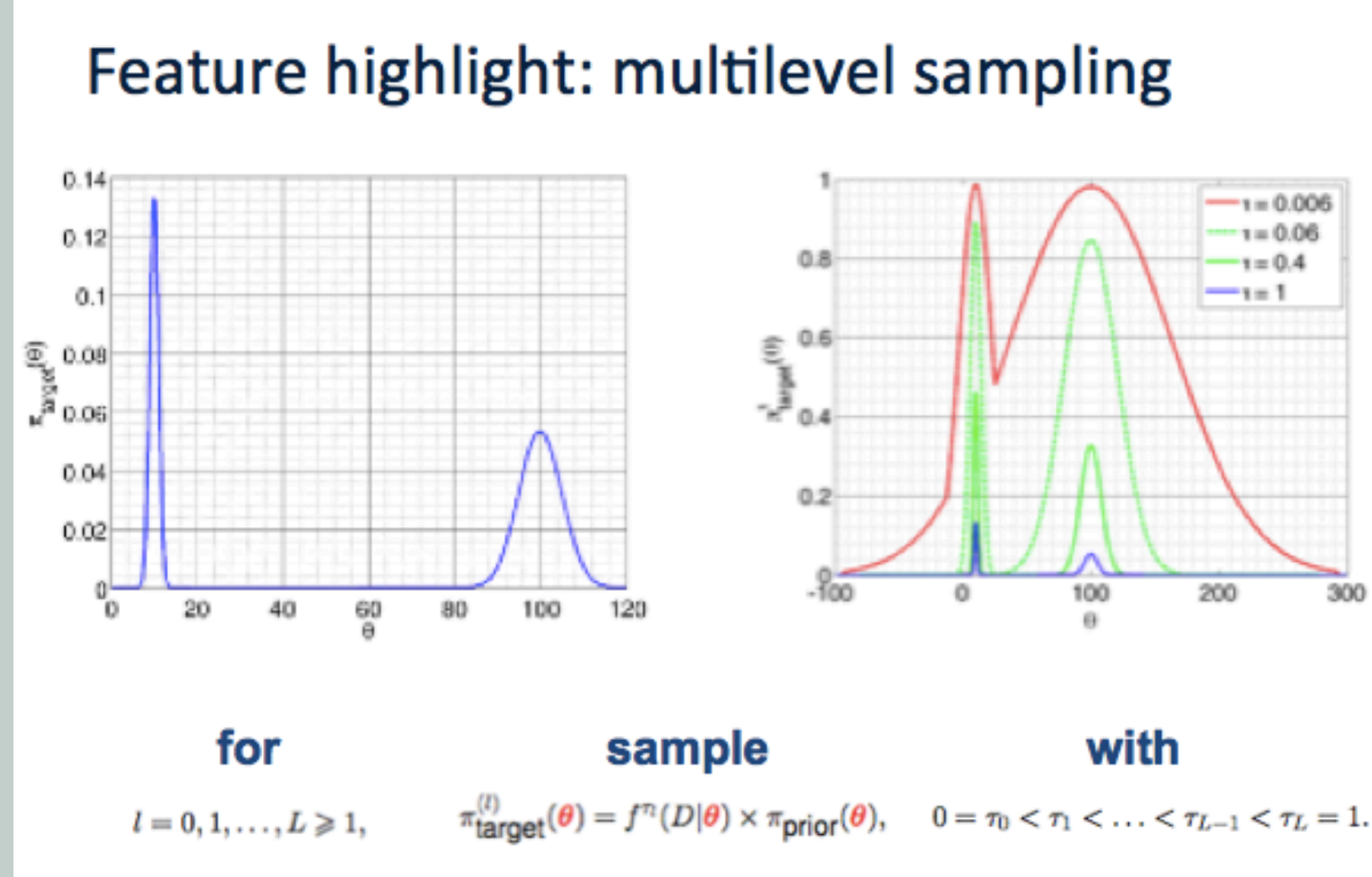
The QUESO library is a collection of parallel statistical algorithms and object-oriented programming constructs supporting research into the uncertainty quantification of mathematical models.

- Quantification of Uncertainty for Estimation, Simulation and Optimization
- Available at <https://github.com/libqueso/queso>
- Development began in 2008 under the PSAAP program
- Since 2011, QUESO development has been part of the SCIDac3 program as part of the QUEST center
- QUESO is used by Dakota to solve the inverse problem
- QUESO has traditionally focused on the inverse problem but has forward propagation capabilities as well



Why use QUESO?

- Other solutions available: R, PyMC, emcee, MICA, etc
- QUESO solves the same problem but has significantly more CS&E capabilities
 - Has been designed to be used with large forward problems
 - Has been used with over 10k cores
 - Support for finite and infinite dimensional problems
 - Can sample multimodal distributions
 - Can leverage Dakota for forward propagation (Dakota can use QUESO for the inverse problem)
 - Emulation capabilities being developed



QUEST Impact on QUESO

- PAST
 - Documentation and testing
 - Trilinos integration
 - Dakota integration
 - New example problems
- PRESENT
 - GPMSA
 - More Dakota integration
 - Software quality and usability improvements
 - User community development
 - Infinite dimensional UQ
- FUTURE
 - Further emulation development
 - Continued software engineering improvement
 - Additional options for vector/matrix classes to increase user base
- Opportunity to be adopted as THE community code for uncertainty quantification

HPC Relevance

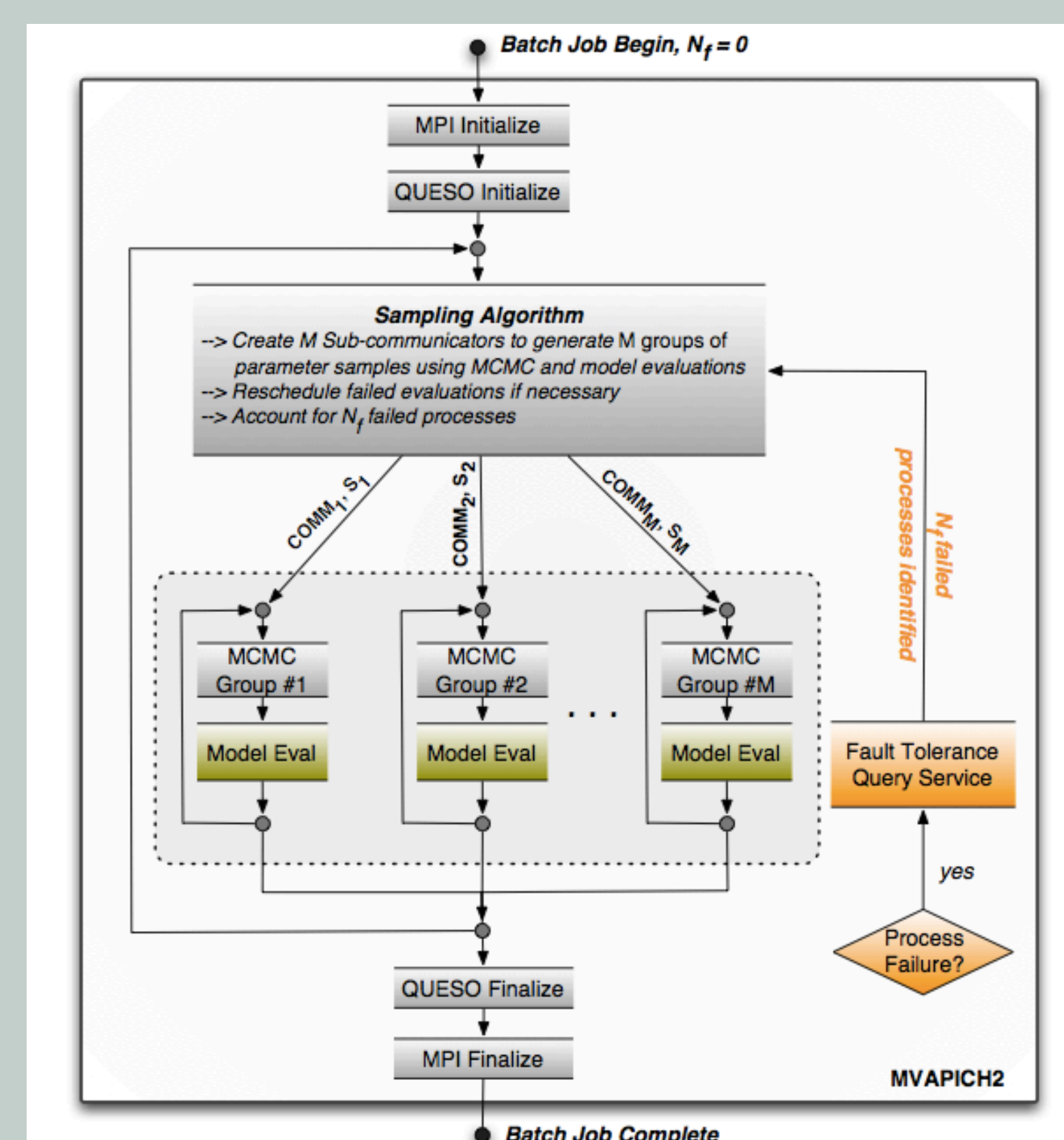
- Parallel Sampling
- Automatic load-balancing for homogenous systems
- Future load-balancing for heterogeneous systems
- Fault tolerance

Sponsors

- 2008 – 2014: DOE NNSA, PSAAP Program
- 2010 – 2011: DOE SNL-NM, Peridynamics Program
- 2010 – 2012: KAUST, AEA2 Program
- 2011 – 2013: AFOSR, RTC, DDDAS Program
- 2011 – 2015: DOE SC, SciDAC3 Program
- 2012 – 2013: DOE LANL and ORNL, CASL Project
- 2012 – 2014: KAUST, AEA3 Program

Acknowledgements

- E. Prudencio: original author of QUESO
- D. McDougall, K. Schulz, K. Estacio-Hiroms and the rest of the PECOS development team
- DOE laboratories: B. Adams (SNL), M. Eldred (SNL), J. Gattiker (LANL), D. Higdon (LANL), L. P. Swiler (SNL), B. Williams (LANL).

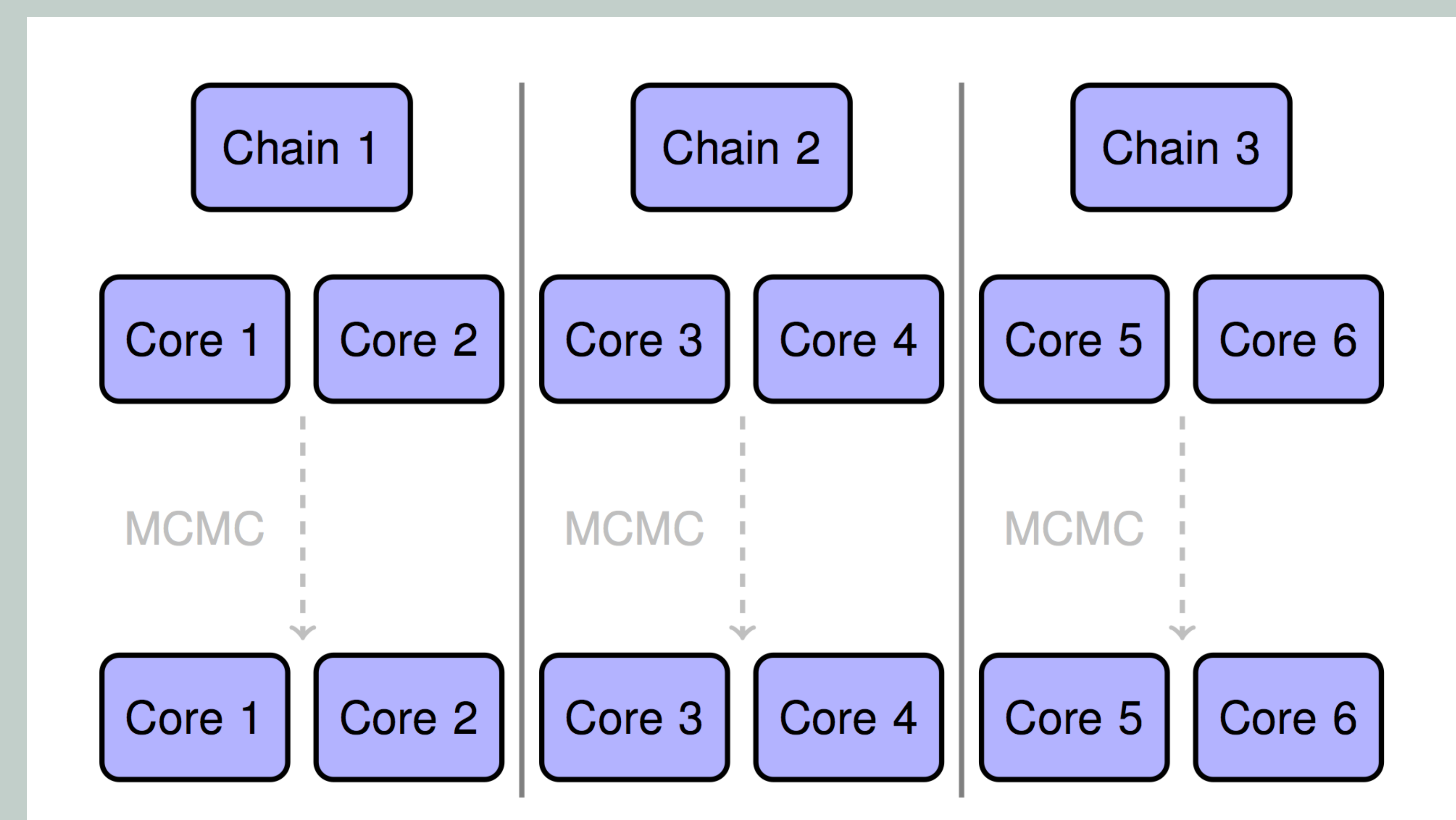


E. E. Prudencio and K. W. Schulz. The parallel C++ statistical library QUESO: Quantification of Uncertainty for Estimation, Simulation and Optimization. In M. Alexander et al., editors, Euro-Par 2011 Workshops, Part I, volume 7155 of Lecture Notes in Computer Science, pages 398-407. Springer-Verlag, Berlin Heidelberg, 2012.

Recent Changes in QUESO

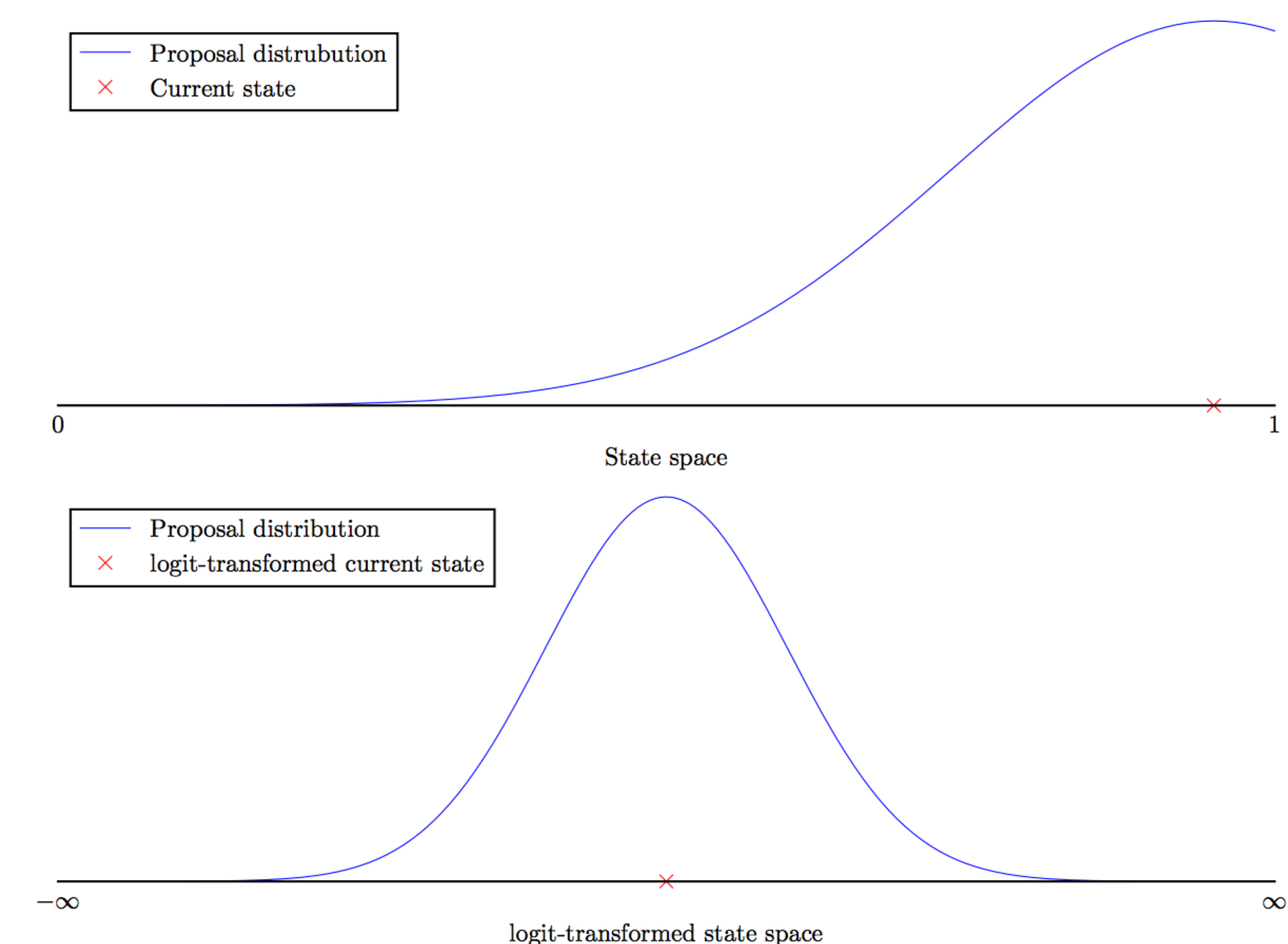
- New swanky website: <http://libqueso.com>
- Version 0.51.0
 - Add canned likelihood for scalar GPMSA use-case a la Higdon et al
 - Add a logit-transformed transition kernel for more efficient proposals
 - Adding Jeffreys distribution as an available prior distribution
 - Adding likelihood value caching to ML sampler
- Version 0.52.0
 - Add canned Gaussian likelihoods for different full/diagonal/scalar/block diagonal covariance matrices
- Above: 428 commits
- DAKOTA integration. Ongoing work with Laura Swiler, Brian Adams, Mike Eldred, Brian Williams
- Version 0.53.0
 - Add linear interpolation surrogates
 - Refactor input options processing
 - Refactor existing queso errors and asserts
 - Add new error checking macros
 - Add basic scoped pointer wrappers
 - Better error message reporting for bad sample covariance matrices

Parallel Chains in QUESO



Increased Sample Acceptance Rate via Logit Transformation

Performance improvement with $\text{logit}(x) = \log(1/(1-x))$
This maps $(0, 1) \rightarrow (-\infty, \infty)$. We never propose states out of bounds:



QUESO featured in the Springer UQ Handbook

QUESO chapter co-authored with Damon McDougall, Nicholas Malaya and Robert Moser

- Formulates the types of problems QUESO can solve
 - Example for showcasing how the user interacts with the API in general
 - Showcases a specific example (ball-drop problem)
 - Showcases an infinite-dimensional UQ example
 - Showcases extensibility through custom priors (for, perhaps, hierarchical Bayesian problems)
- Extended documentation effort!