Surrogate Based Optimization

Charles F. Jekel

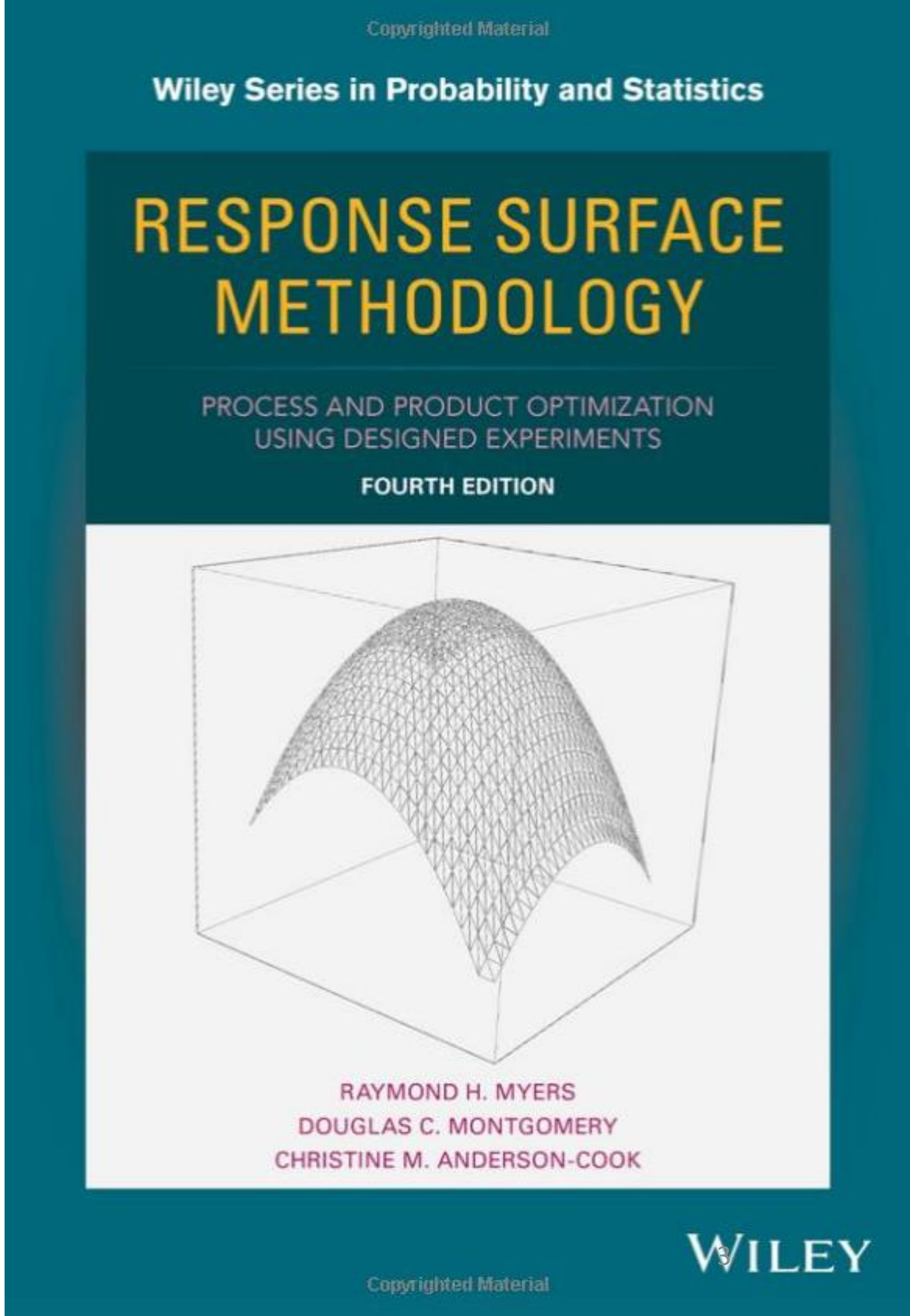
10/15/2020

Contents

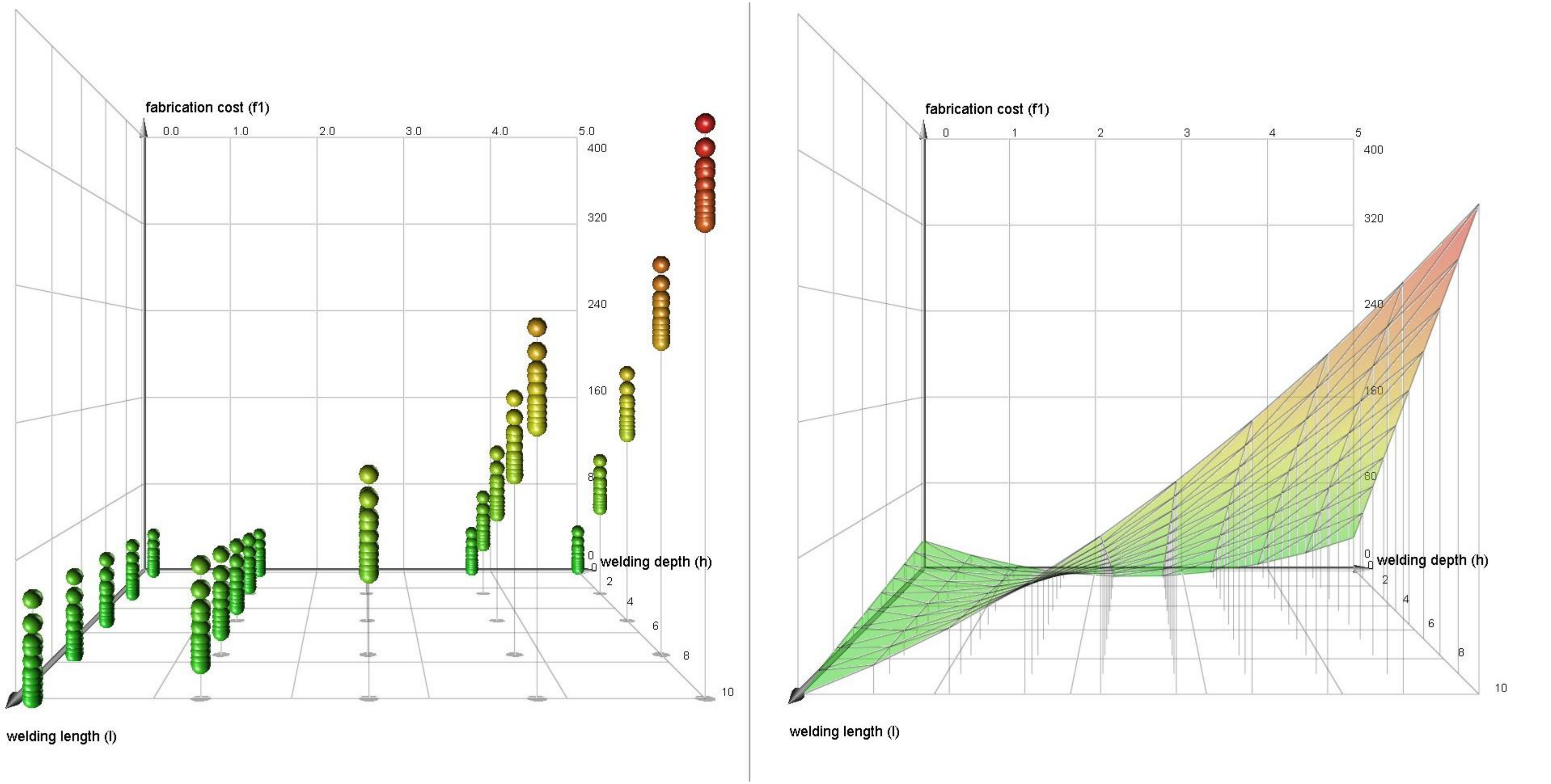
- Response Surface Methodology
- Basic surrogate based optimization algorithms
- Specific optimization algorithms
  - Successive Response Surface Method (SRSM)
  - Efficient Global Optimization (EGO)
- Example Shape Optimization

Response Surface Mythology

- Collection of algorithms and techniques related to optimization experimental or manufacturing process
- Origins in the 1950s
- In a nutshell
  - An experiment or process, where you don’t understand the full physical problem
  - You control the variables as best as possible, but you don’t fully understand the physics, you get highly variable outcomes
  - Goal to optimize the process for an outcome
  - RSM it the toolset for these problems



Response Surface Methodology (RSM) Fitting smooth surfaces (mostly polynomials) to highly variable data



Why mention Response Surface Mythology (RSM)

- Maybe more relevant to optimizing experimentation rather than computer simulations
- Origins of surrogate based optimization of computer simulations
- Pool of literature on
  - Sampling techniques
  - Regression models
  - Statistics of regression models

RSM may be useful in some applications, and the approach is in many ways the opposite of Surrogate Based Optimization

Surrogate Based Optimization [1]

- Focus on the outputs of computer simulations
- Accurate, high-fidelity simulations are expensive to run, and may not provide gradient information
- Desire to optimize these models
- Surrogate model is used as a cheap representation of a fidelity model

[Insert image from your expensive physical simulation that generates pretty pictures here]

[1] Nestor V. Queipo, Raphael T. Haftka, Wei Shyy, Tushar Goel, Rajkumar Vaidyanathan, P. Kevin Tucker, Surrogate-based analysis and optimization, Progress in Aerospace Sciences, Volume 41, Issue 1, 2005, Pages 1-28, ISSN 0376-0421, https://doi.org/10.1016/j.paerosci.2005.02.001.

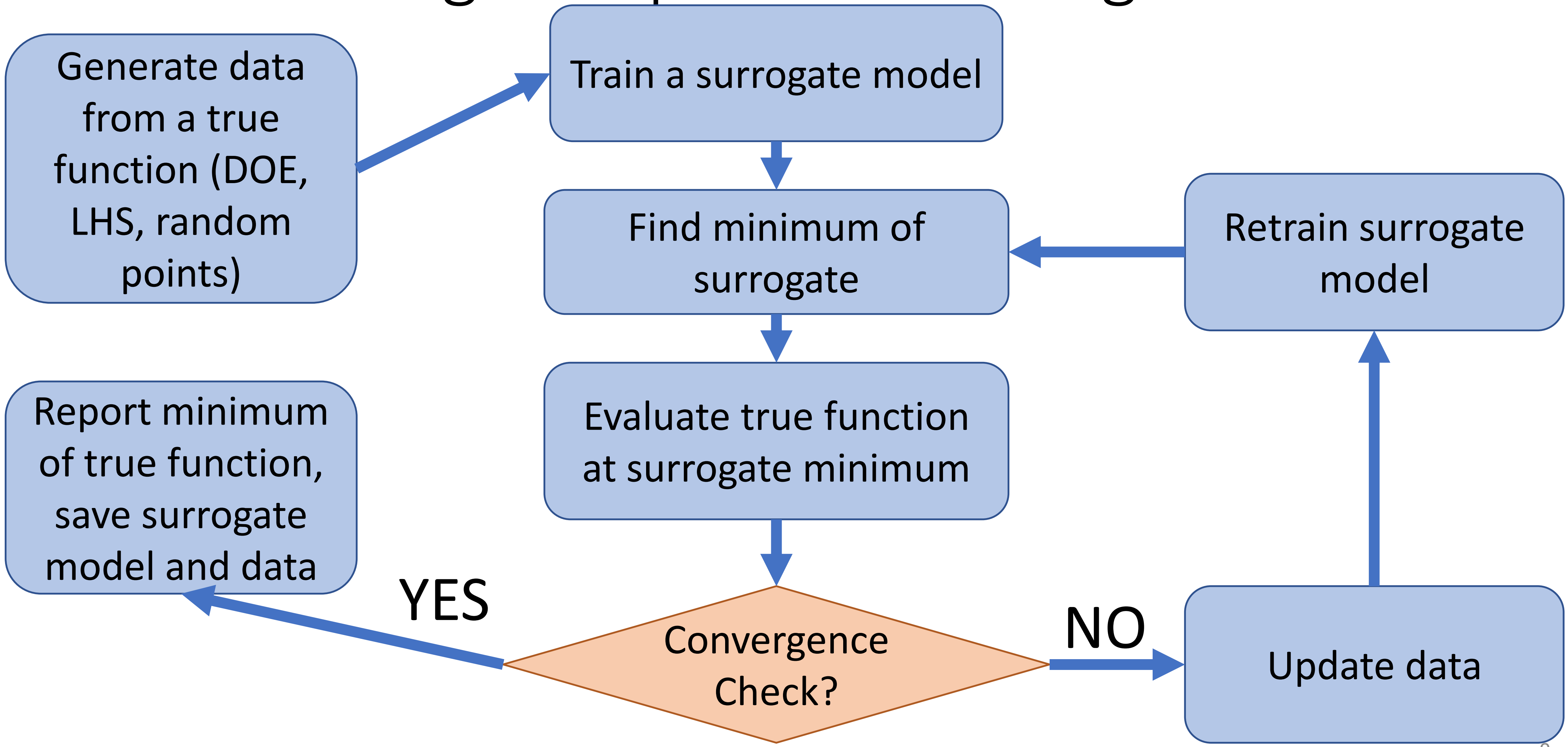
When **not to** Surrogate Based Optimization

- Cheap analytical functions
- Cost of fitting and evaluating surrogate model is more expensive than function evaluation
- Cheap derivative information

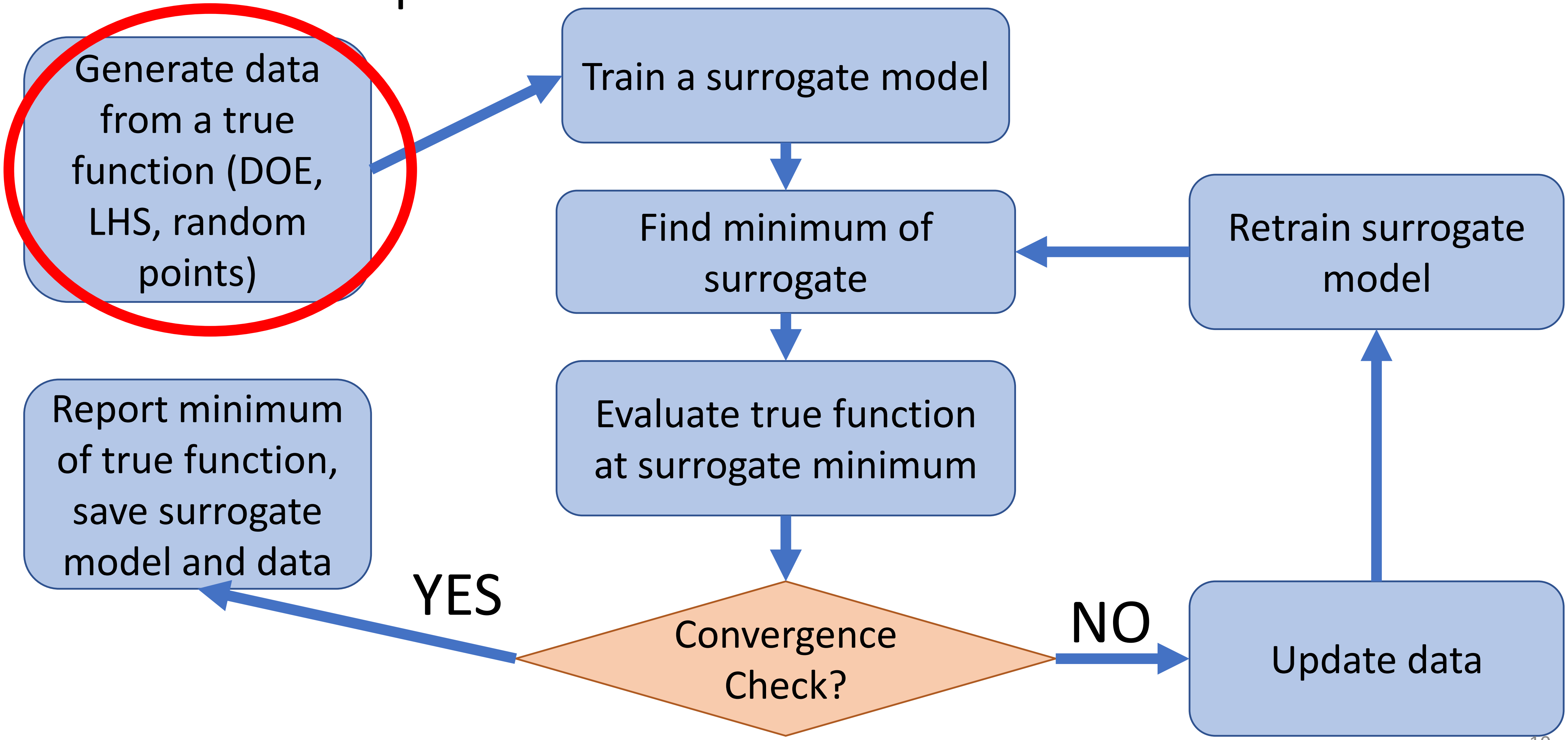
When **to** Surrogate Based Optimization

- Expensive functions
- Cost of fitting and evaluating surrogate model is much cheaper than a function evaluation
- Lack of cheap derivative information
- Intention to re-use or re-run many future similar optimizations
- Gaining more insight into optimization problem
- You have no idea where an optimization should start

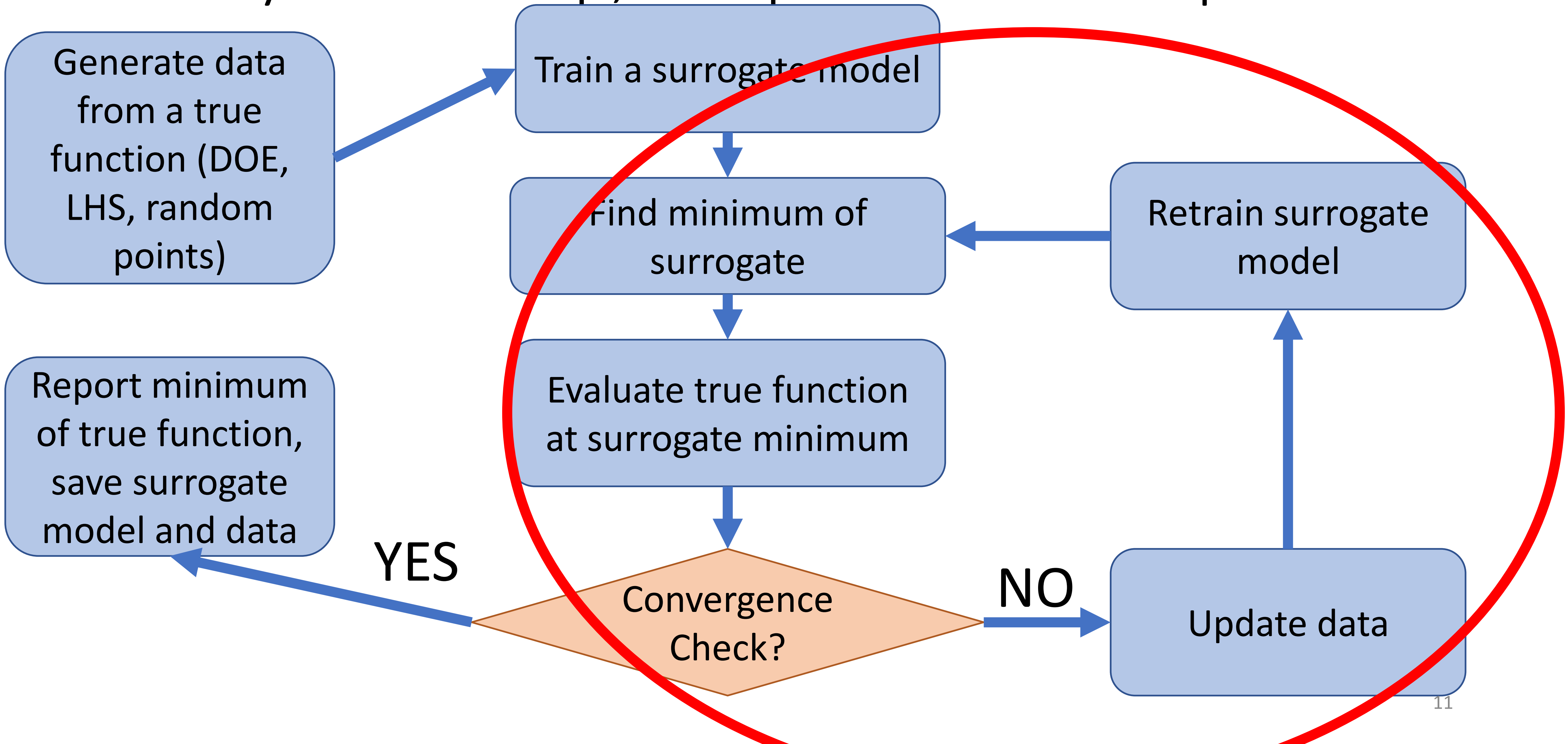
Basic Surrogate Optimization Algorithm



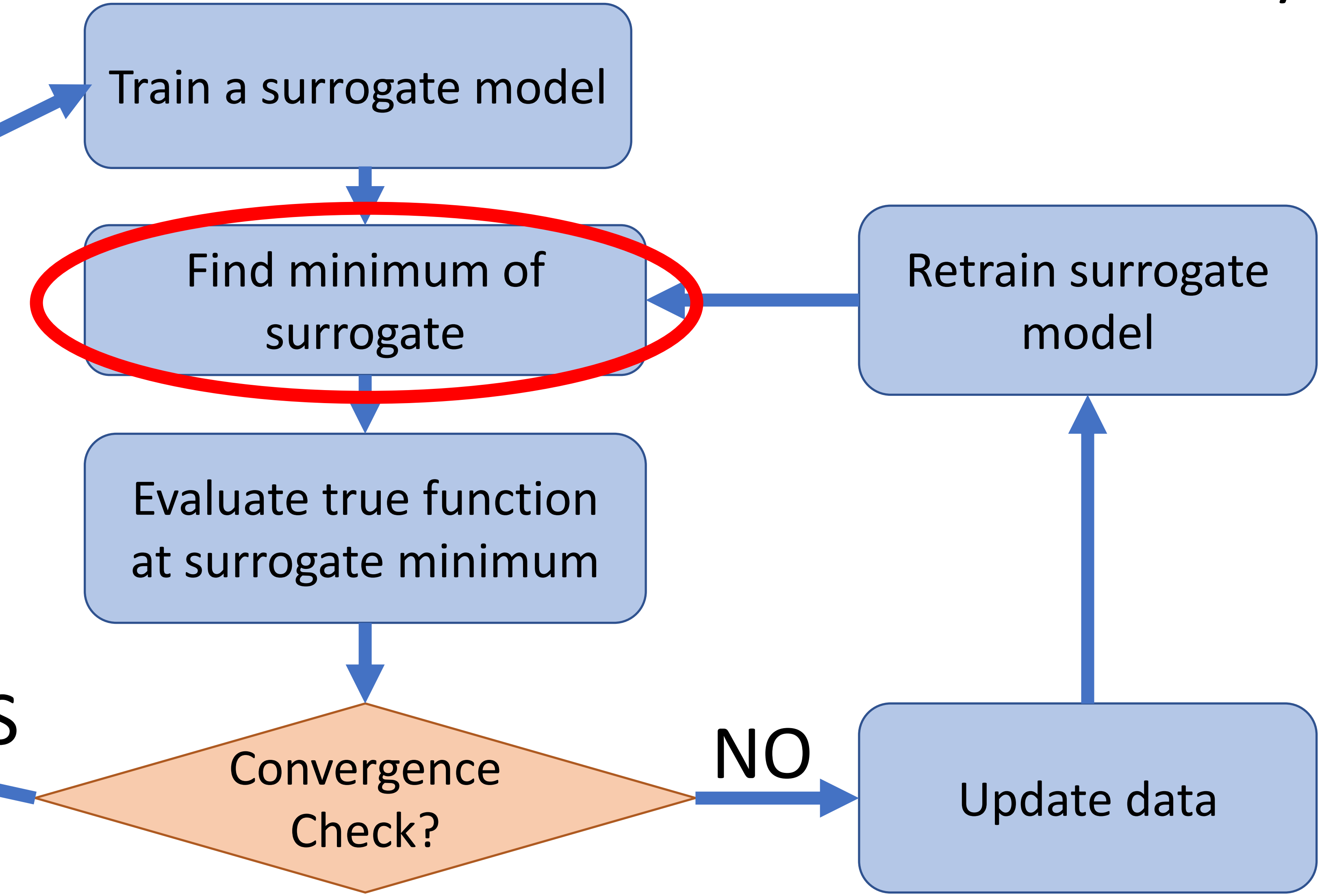
Trivial to parallelize



Mostly serial loop, but parallelization possible

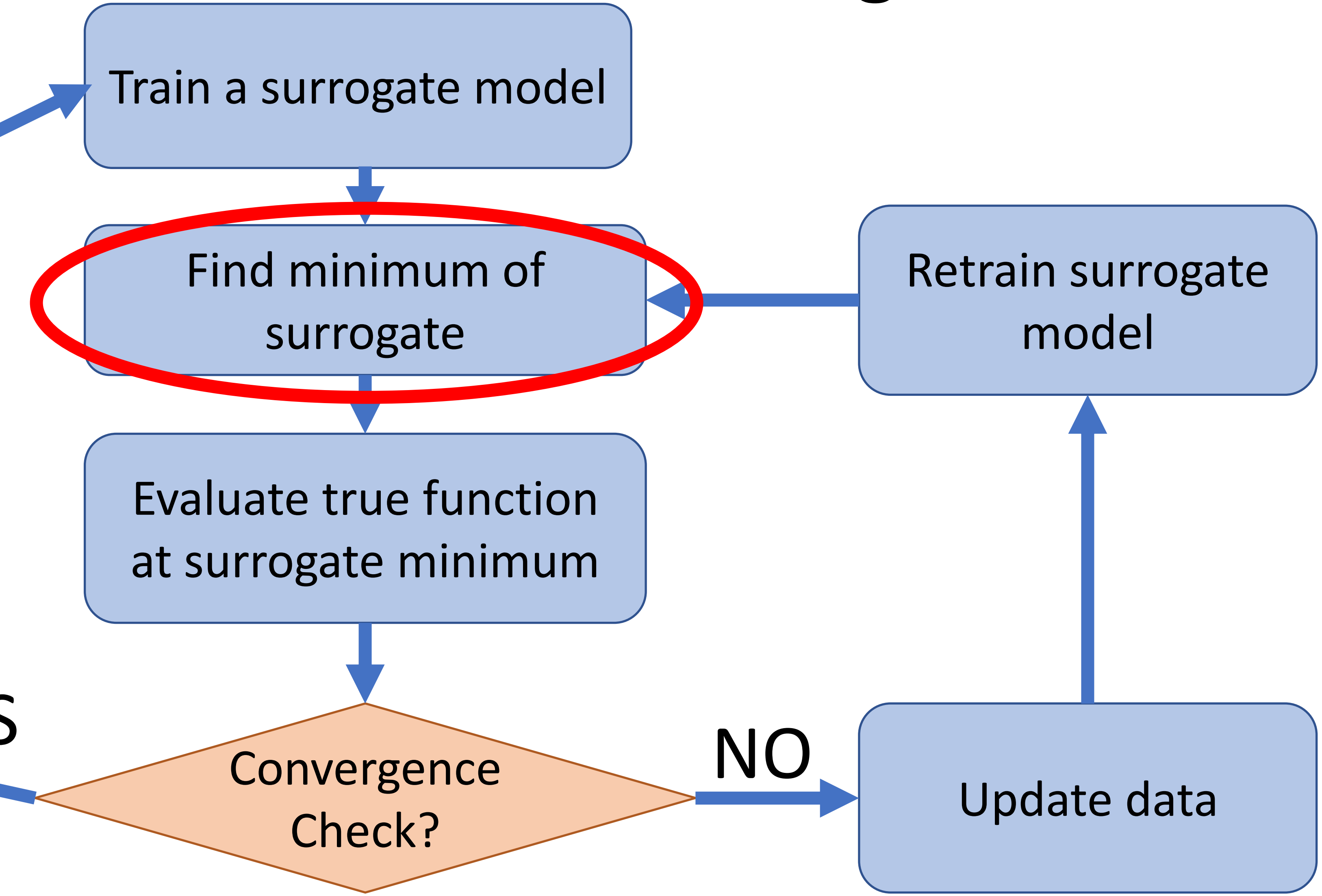


How to find the minimum of your surrogate?



- Use your favorite NLP routine
  - (BFGS, SQP, MMA)
- Use metaheuristics
  - (Genetic algorithm, partial swarm, differential evolution)
- Monte Carlo or pure random search
- **Be very thorough!**

Minimum of surrogate occurs at existing data!?



- The minimum of an interpolation surrogate can often be at an existing data point!
- Observed this behavior with Radial Basis Functions
- **Need more data!**

Other issues with the Basic algorithm

- How do you get convergence guarantees?
- Minimizing surrogate does not improve surrogate model
- How to parallelization serial loop
- Dealing with constraints?
- Dealing with more than one objective?

Get guarantees on local convergence

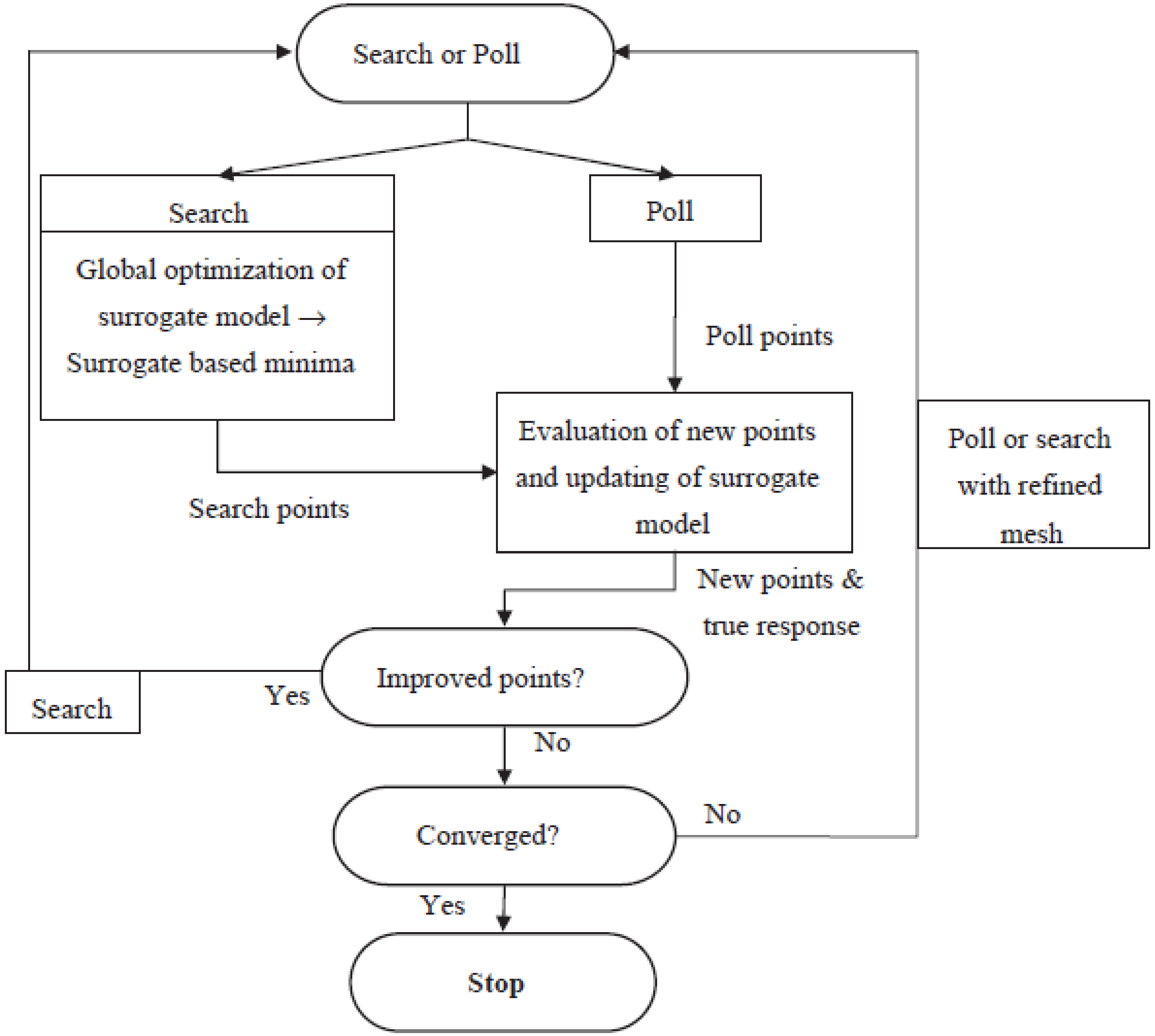


Fig. 8. The basic SMF algorithm.

- Fig 8 from [1]
- Poll phase looks at extending from best solution
- Small steps in all major directions
- Poll can be thought of as central difference steps
- Poll answers whether we have local optimum
- Search phase minimizes surrogate model

Successive Response Surface Method (SRSM)

- Local optimization algorithm
- **Domain reduction technique**
- This algorithm is based on Response Surface Methodology (RSM)
- 20 year commercial usage in LS-OPT for optimizing expensive dynamic finite element analysis [2]

[2] Stander, N. and Craig, K.J. (2002), "On the robustness of a simple domain reduction scheme for simulation-based optimization", *Engineering Computations*, Vol. 19 No. 4, pp. 431-450. https://doi.org/10.1108/02644400210430190

Domain reduction techniques

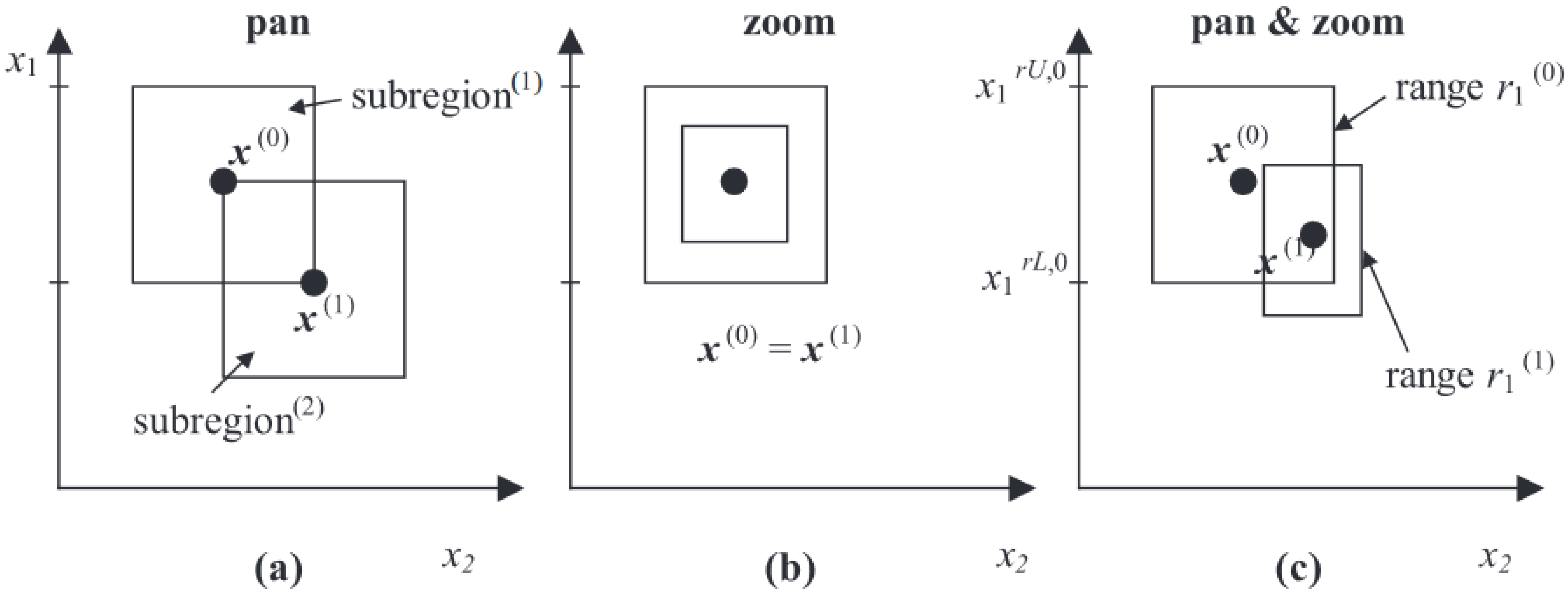


Figure 1.
Adaptation of subregion in SRSM: (a) pure panning, (b) pure zooming and (c) a combination of panning and zooming

- Figure from [2]
- Reduce the boundary of search space
- Successive iterations reduce the design domain

D-Optimal vs Latin Hypercube (LHS) DOEs

- Figure from [3]
- Design of Experiments (DOE)
- D-Optimal
  - Variance optimal design
  - Samples placed on boundaries
  - Model specific
- LHS: Space filling design

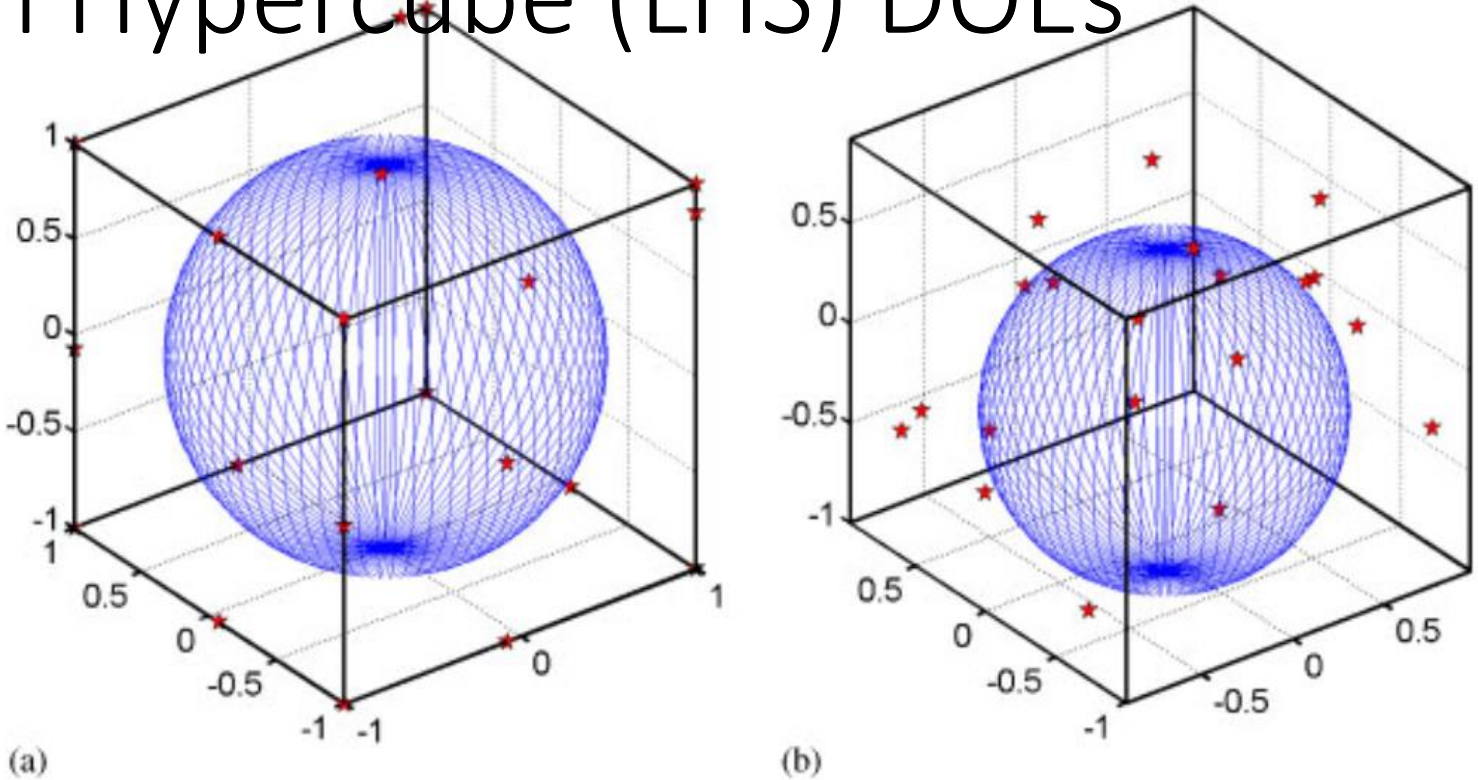
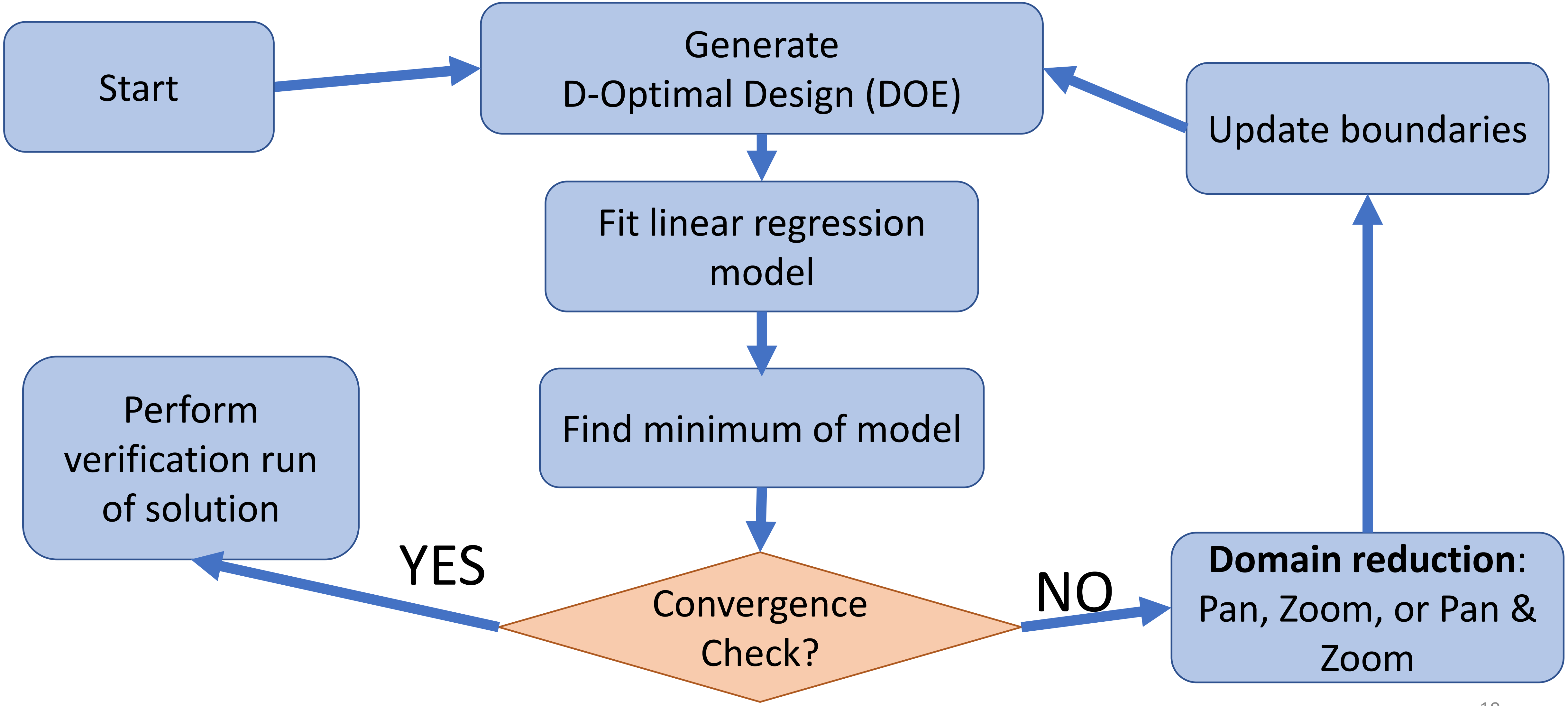


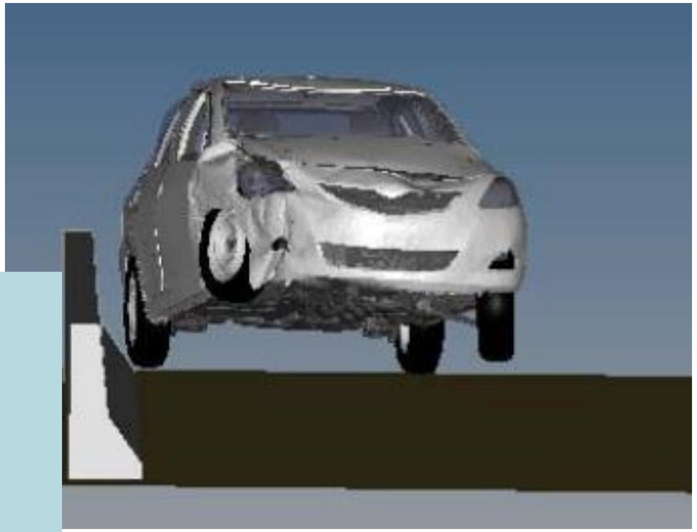
Figure 2. Illustration of the largest spherical empty space inside the three-dimensional design space $[-1, 1]^3$ (20 points): (a) D-optimal design and (b) LHS design.

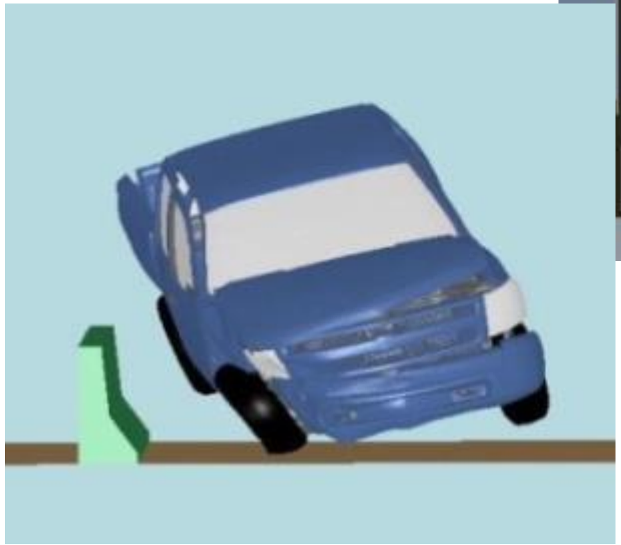
[3] Goel, T., Haftka, R.T., Shyy, W. and Watson, L.T. (2008), Pitfalls of using a single criterion for selecting experimental designs. Int. J. Numer. Meth. Engng., 75: 127-155. doi:10.1002/nme.2242

Successive Response Surface Method (SRSM)

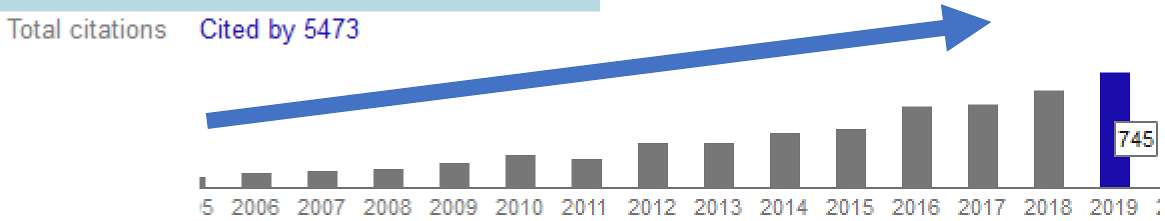


Efficient Global Optimization (EGO)





- Surrogate based global optimization algorithm
- Uses adaptive sampling
- Bayesian optimization
- Expensive computer simulations
- Typically use Gaussian process (or Kriging) surrogates
- Popular for tuning hyper parameters of ML models



[4] Jones, D.R., Schonlau, M. & Welch, W.J. Efficient Global Optimization of Expensive Black-Box Functions. Journal of Global Optimization 13, 455–492 (1998). https://doi.org/10.1023/A:1008306431147

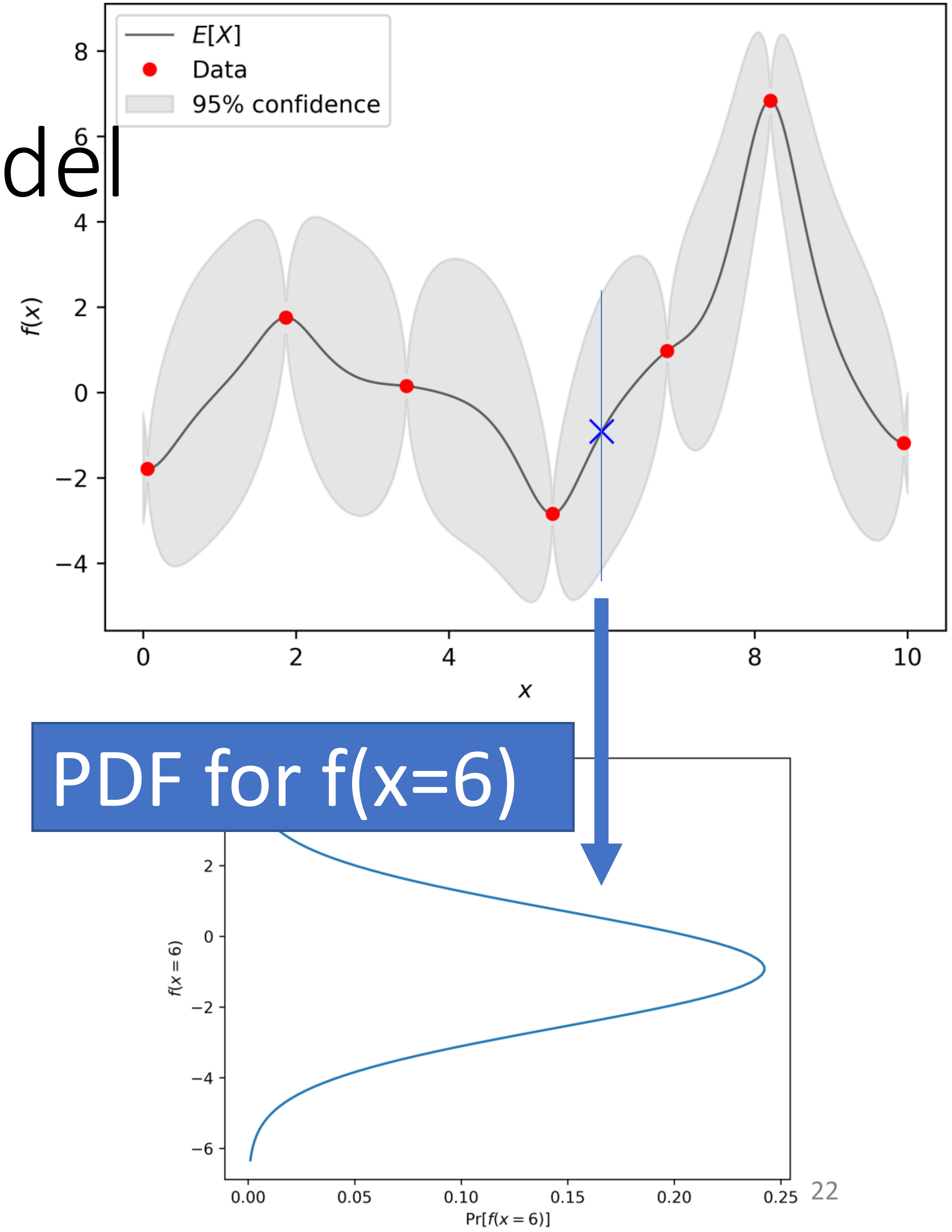
Car crash figures from -> [5] Marzougui D, Kan CD, Opiela KS. Crash test & simulation comparisons of a pickup truck & a small car oblique impacts into a concrete barrier. InThe 13th International LS-DYNA Users Conference, Dearborn, MI 2013.

EGO open source implementations

Name	License	Technologies	URL
GPyOpt	BSD 3 Clause	Python	https://github.com/SheffieldML/GPyOpt
MOE	Apache v2	C++, Python	https://github.com/Yelp/MOE
Spearmint	Non-commercial	Python	https://github.com/JasperSnoek/spearmint
SMT	BSD 3 Clause	Python	https://github.com/SMTorg/smt
GPflow	Apache v2	Python, TensorFlow	https://github.com/GPflow/GPflow
Bayesopt	GNU Affero v3.0	C++, Python	https://github.com/rmcantin/bayesopt

Gaussian Process (GP) model

- Right show GP fit to select Data
- Line shows expected value
- Gray area shows 95% confidence region
- Prediction variance is normally distributed
- **PDF for uncertainty of predictions!**
- Recall what I said about the minimum of interpolation models is usually at an existing point!



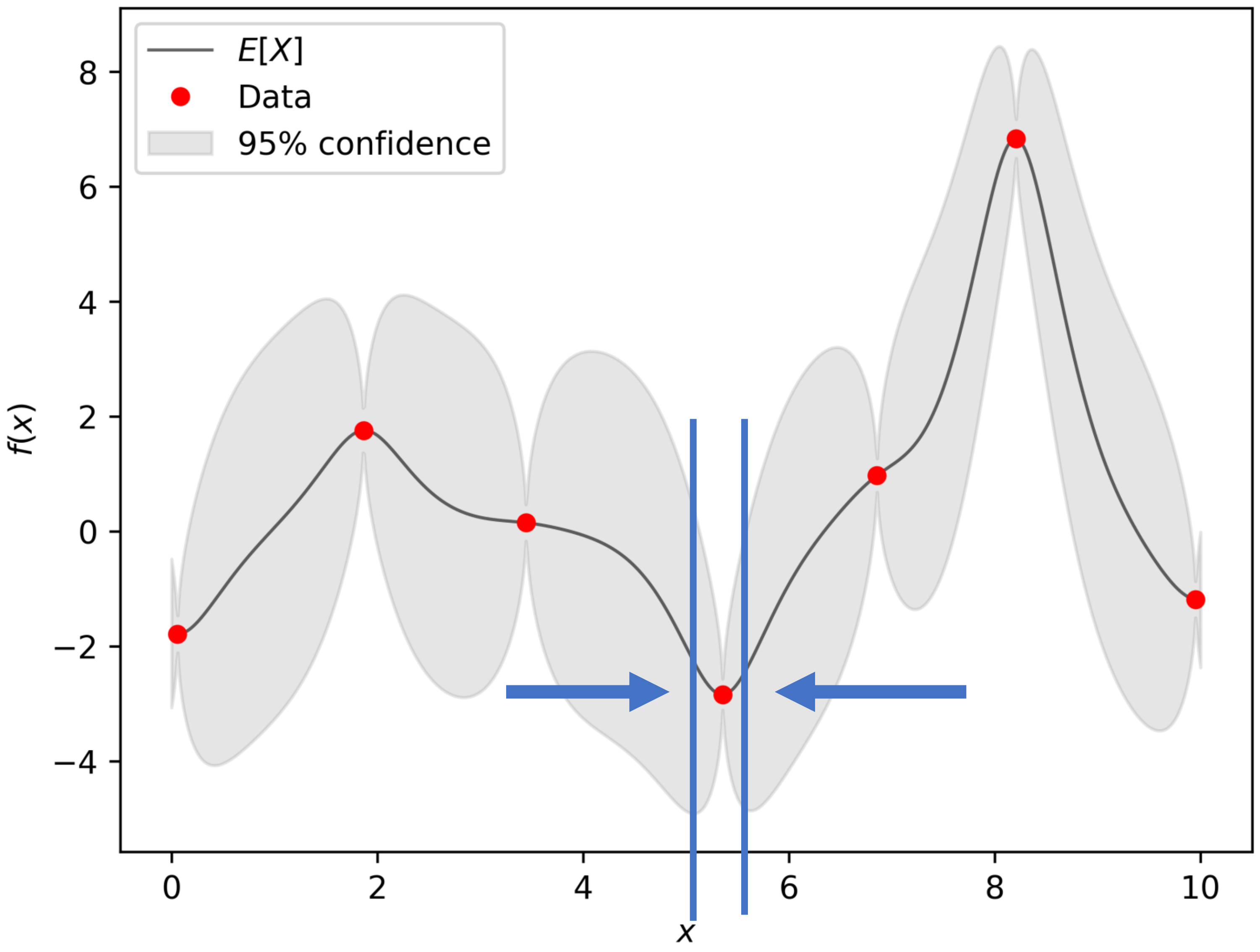
Gaussian Process (GP)
Probability Density Function (PDF)

Global optimization balance between exploitation & exploration

- **Exploitation**
  - Local improvements
  - Small improvements from our best solution
  - Genetic algorithm, offspring from 2 good parents
- **Exploration**
  - Global improvements
  - Improvement across entire domain
  - Genetic algorithm, offspring from random mutation

Exploitation

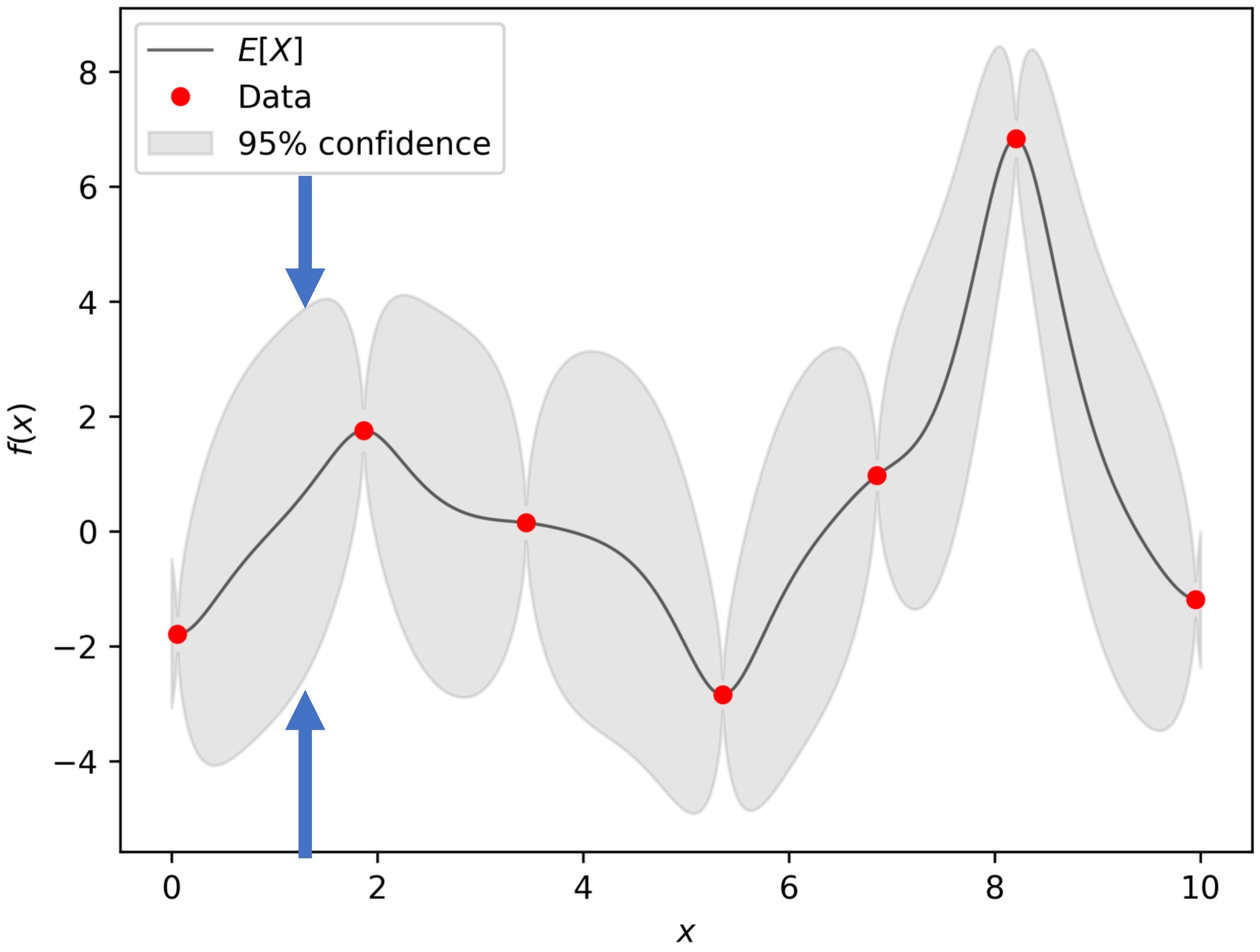
Looking for improvement in the vicinity of our Present Best Solution (PBS)



Exploration

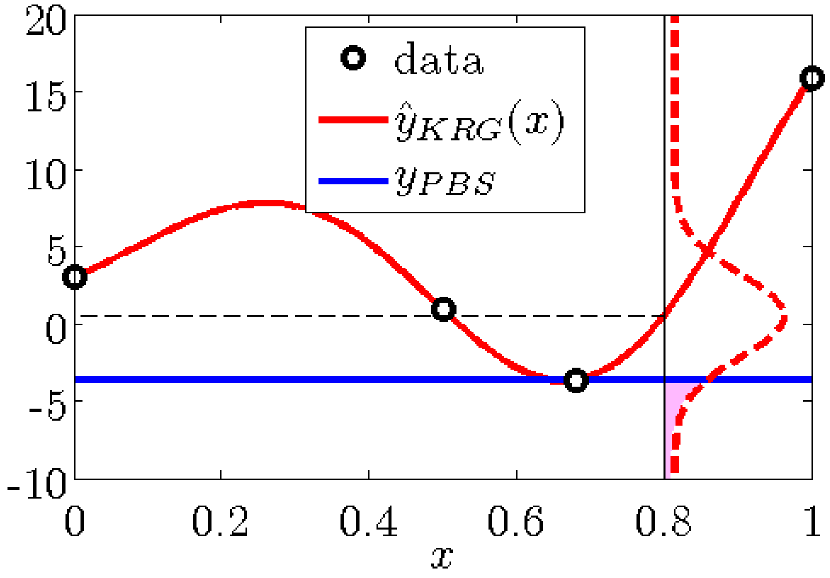
Search to reduce uncertainty!

Where model's variance is largest



Expected Improvement (EI) strikes balance!

- EGO maximizes the EI to select the next point to sample
- How much do we anticipate upon improving from our Present Best Solution (PBS) ?
- EI reflects the probability that the optimizer improves upon PBS



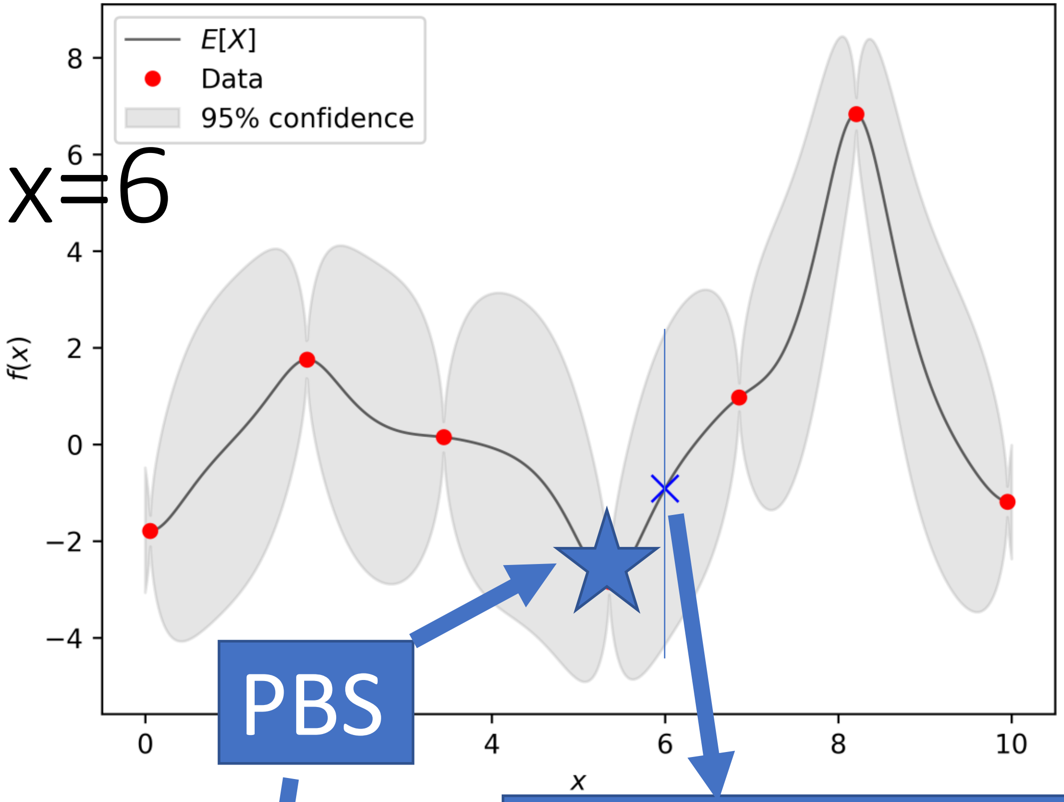
$$E[I(\mathbf{x})] \equiv E[\max(y_{PBS} - Y, 0)]$$

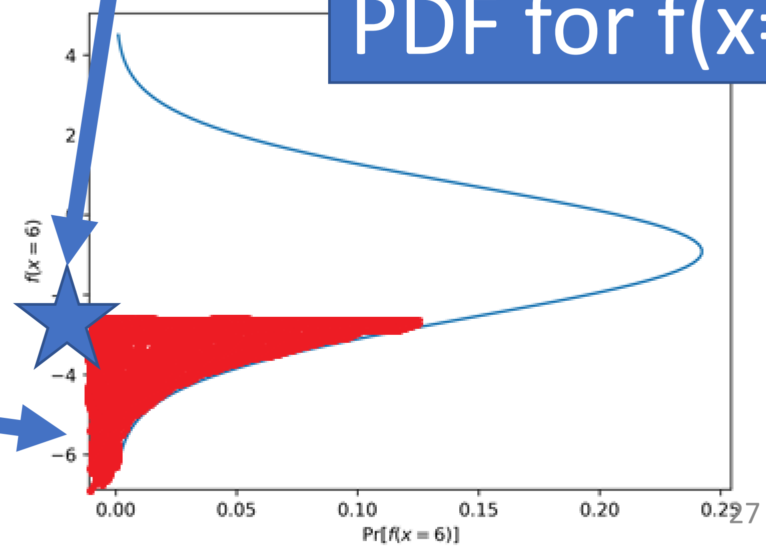
$$E[I(\mathbf{x})] = (y_{PBS} - \hat{y})\Phi\left(\frac{y_{PBS} - \hat{y}}{s}\right) + s\phi\left(\frac{y_{PBS} - \hat{y}}{s}\right)$$

Figure and eqns. From Rafi Haftka's VVUQ lecture on Kriging and EGO, University of Florida. https://mae.ufl.edu/haftka/vvuq/

Expected improvement at x=6

$$E[I(\mathbf{x})] = \underbrace{(y_{PBS} - \hat{y})\Phi\left(\frac{y_{PBS} - \hat{y}}{s}\right)}_{\text{Expected value}} + \overbrace{s\phi\left(\frac{y_{PBS} - \hat{y}}{s}\right)}^{\text{Standard error}}$$

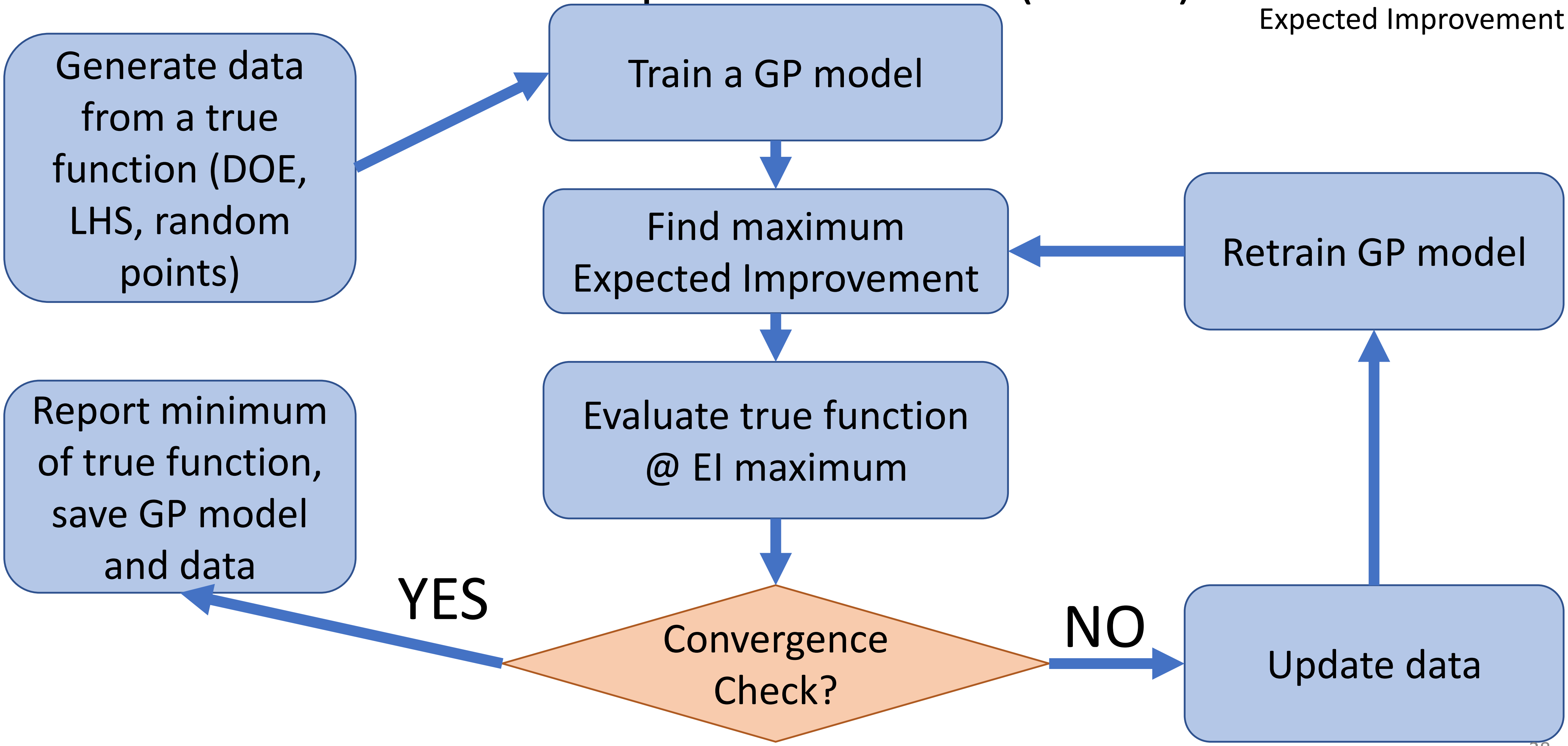




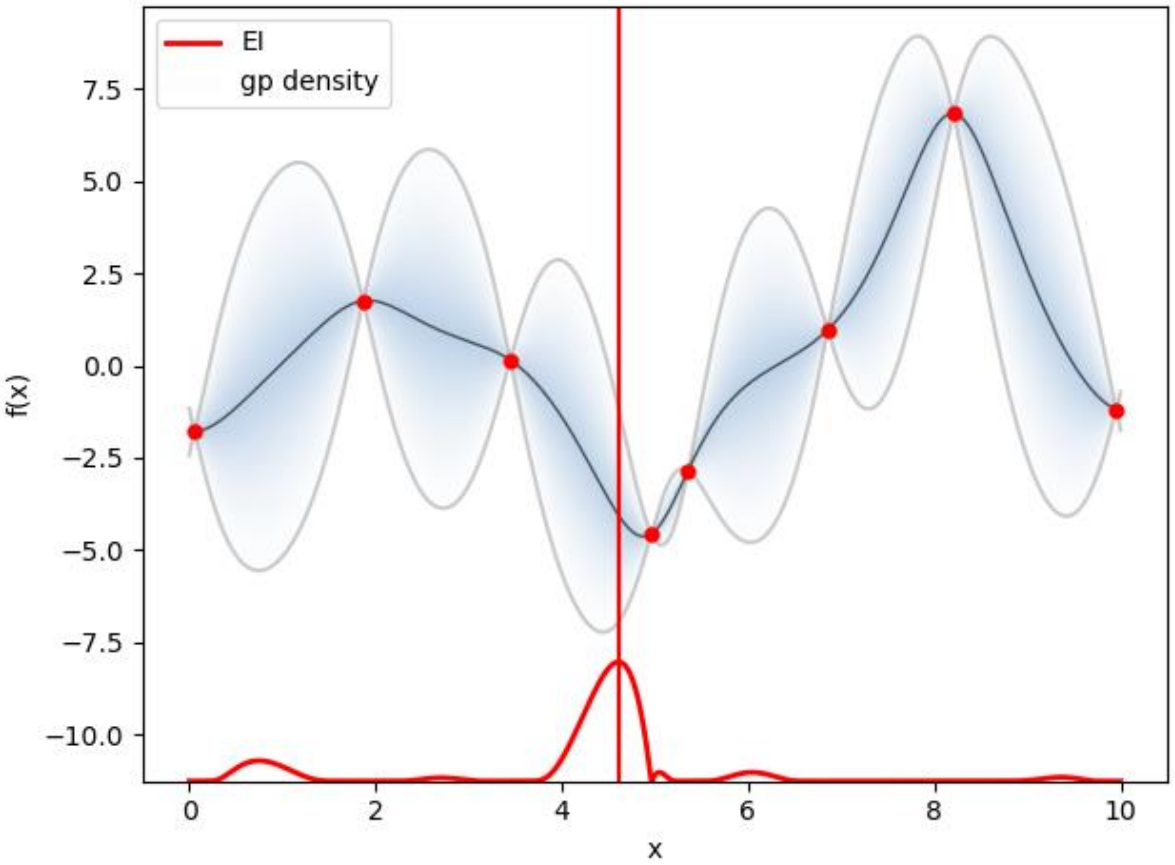
Present Best Solution (PBS)
Probability Density Function (PDF)

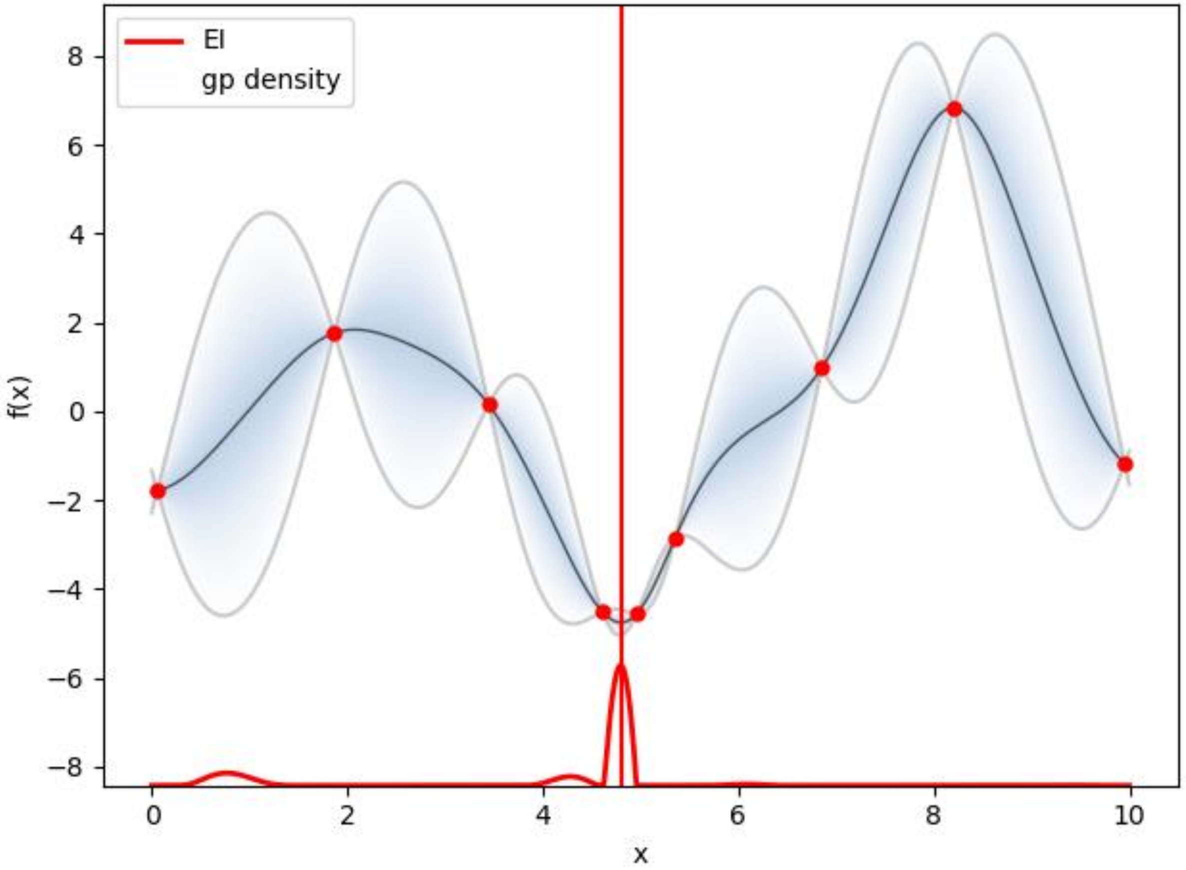
Efficient Global Optimization (EGO)

Design of Experiments (DOE)
Latin Hypercube Sampling (LHS)
Gaussian Process (GP)
Expected Improvement (EI)



Example: Iteration 1 (left), Iteration 2 (right)

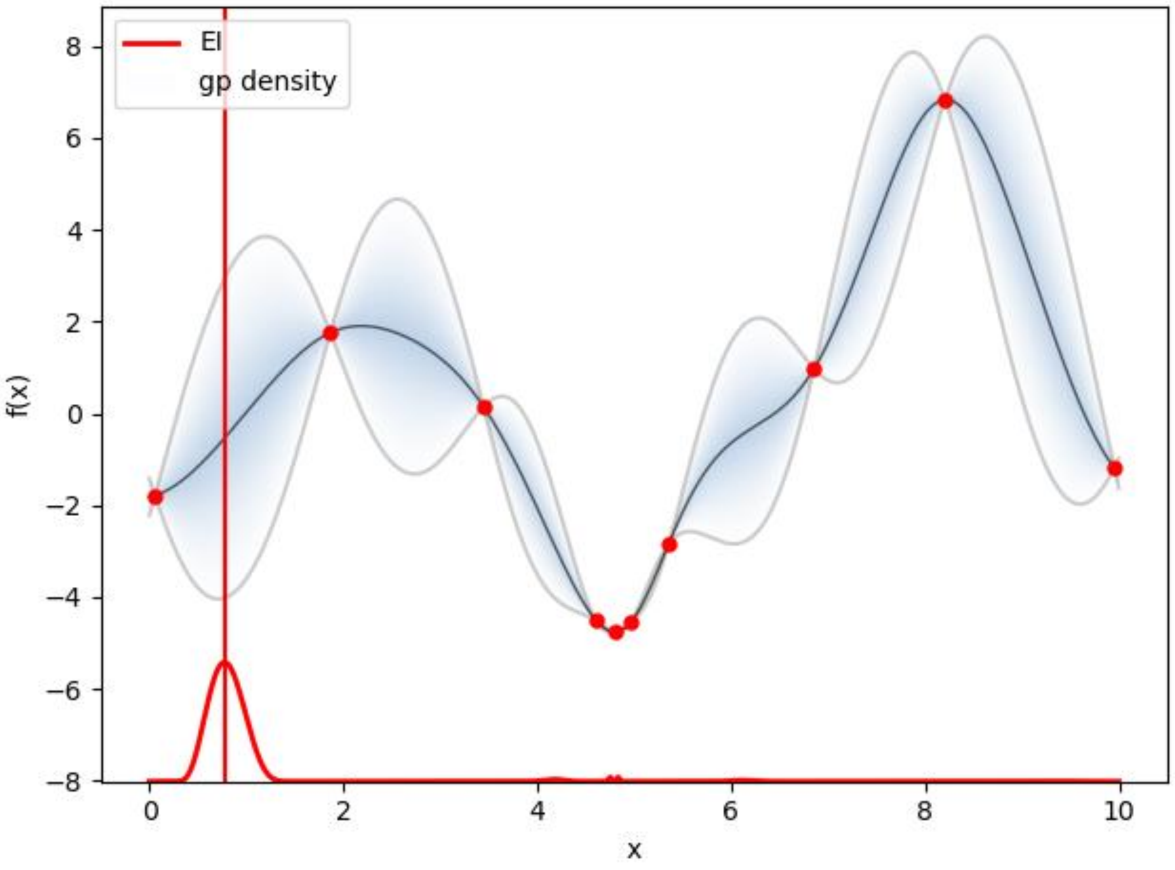


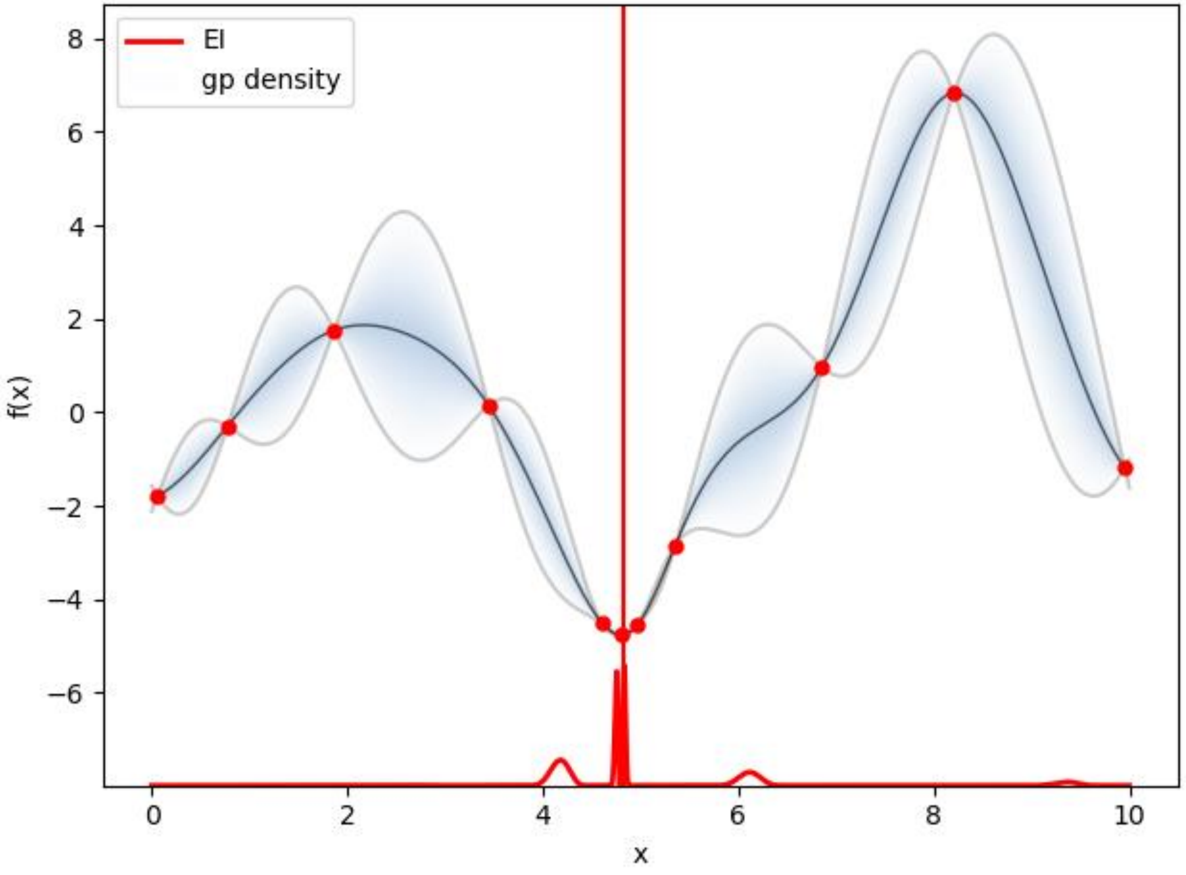


Exploitation

Exploitation

Example: Iteration 3 (left), Iteration 4 (right)

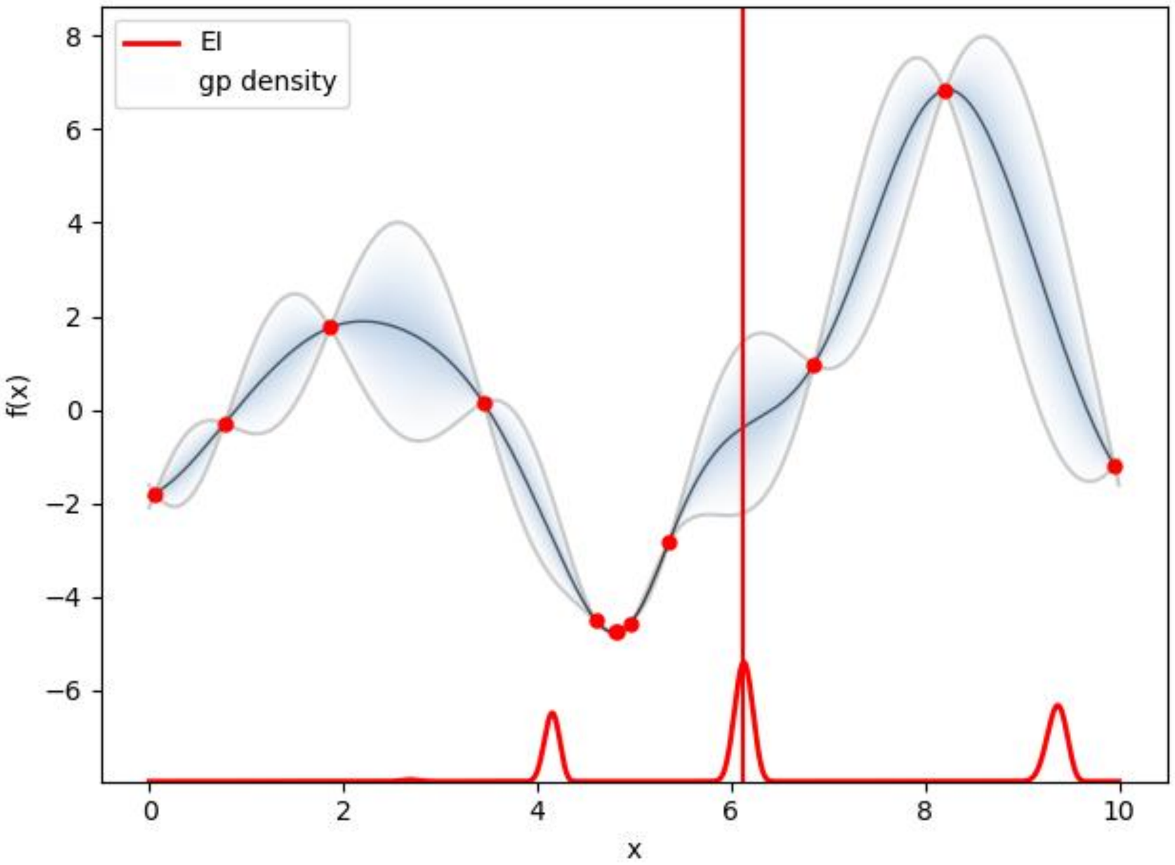




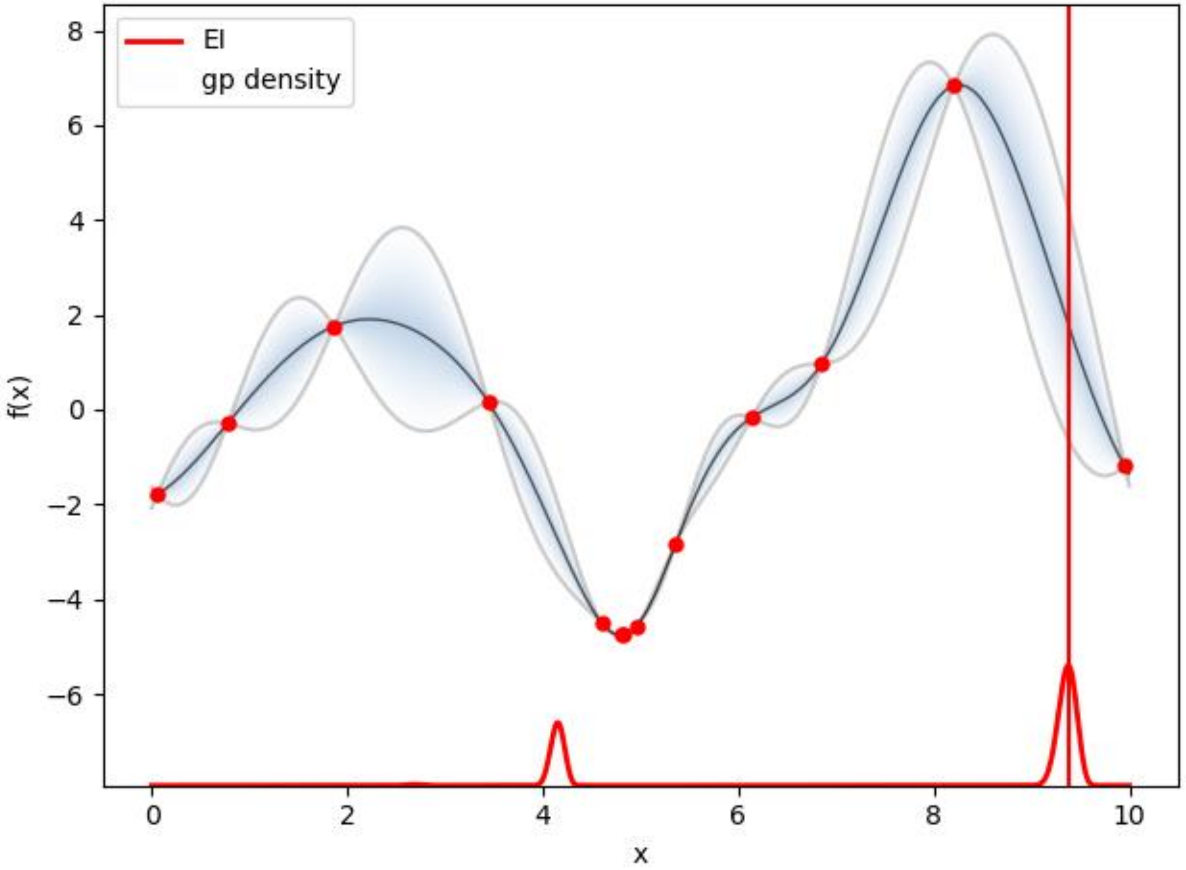
Exploration

Exploitation

Example: Iteration 5 (left), Iteration 6 (right)



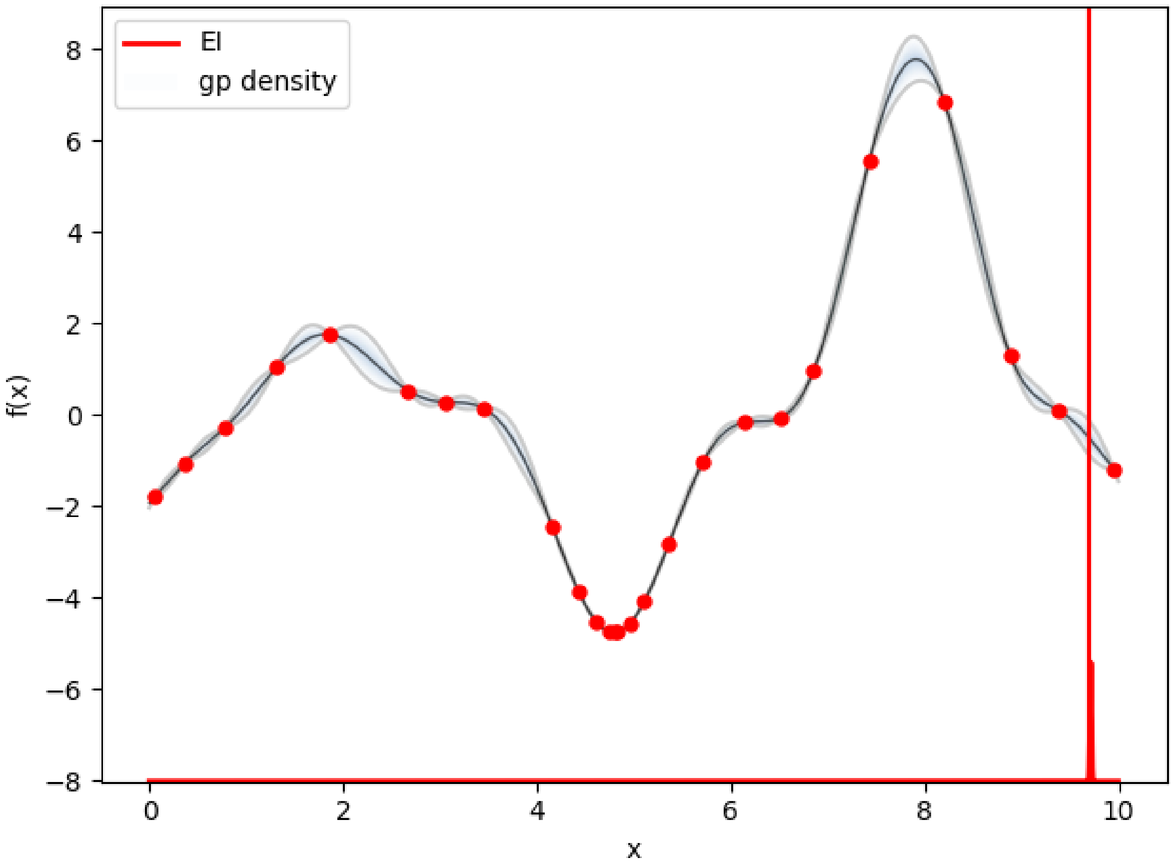
Exploration

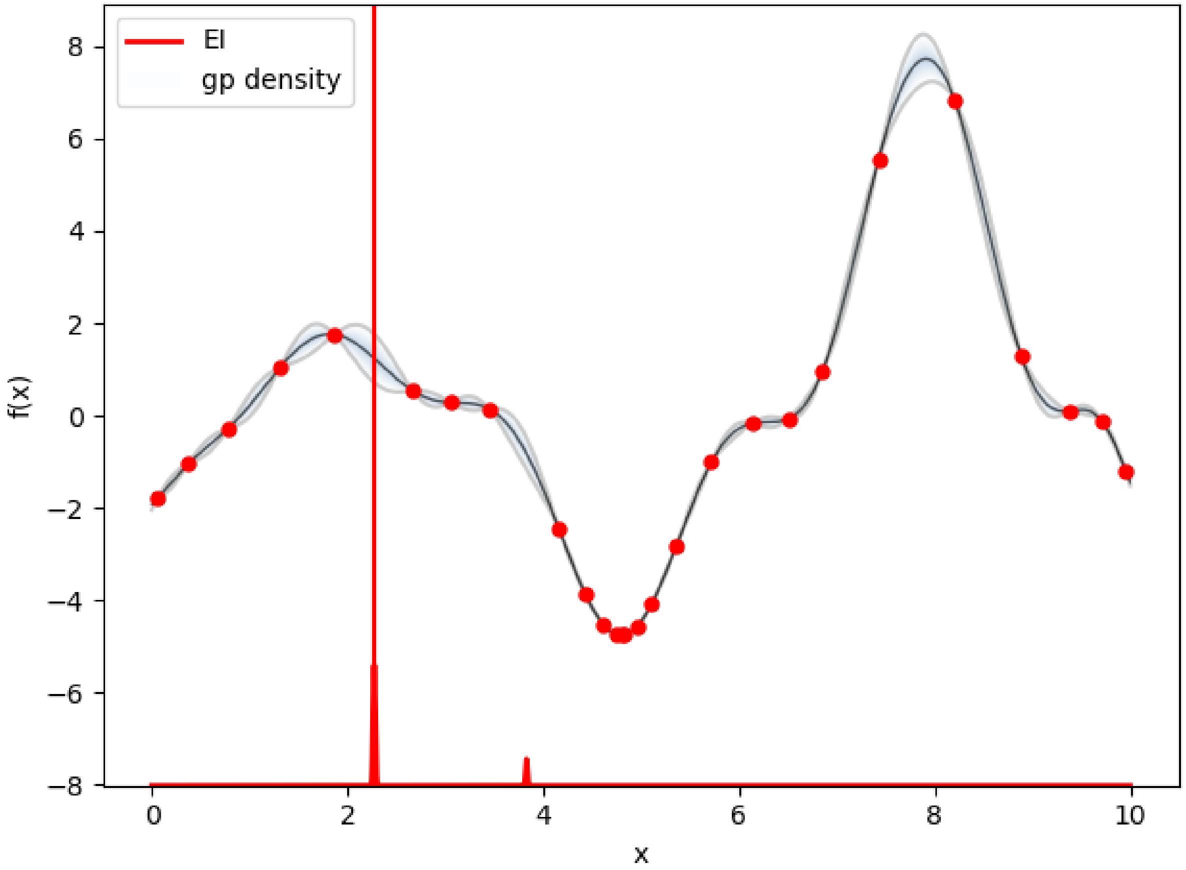


Exploration

…

Example: Iteration 19 (left), Iteration 20 (right)





Exploration

Exploration

Efficient Global Optimization (EGO) Issues

- Final local convergence may take a long time
  - Similar for all global optimization algorithms
  - Consider using local optimizer from EGO result
- How to budge total EGO function evaluations?
  - Good starting point is half of your budge to DOE
- Long serial loop
  - Many parallelization adaptations…
  - Parallelized EGO might always perform worse per function evaluation, but faster real end user time
- EGO is prohibitively costly on cheap functions

Efficient Global Reliability Analysis (EGRA)

- Extension of EGO to reliability analysis problems
- Equations from [6]
- Investigates maximizing Expected Feasibility
- Final GP model used to evaluated system probability of failure (which would be impossible using numerical model)
- Sandia DAKOTA UQ implementation

Constraint

$$p_f = \int \cdots \int_{g > \bar{z}} f_{\mathbf{x}}(\mathbf{x}) \mathrm{d}\mathbf{x}$$

Expected Feasibility

$$\begin{aligned} EF[\hat{G}(\mathbf{u})] = (\mu_G - \bar{z})\left[2\Phi\left(\frac{\bar{z} - \mu_G}{\sigma_G}\right) - \Phi\left(\frac{z^- - \mu_G}{\sigma_G}\right)\right. \\ \left. - \Phi\left(\frac{z^+ - \mu_G}{\sigma_G}\right)\right] - \sigma_G\left[2\phi\left(\frac{\bar{z} - \mu_G}{\sigma_G}\right) - \phi\left(\frac{z^- - \mu_G}{\sigma_G}\right)\right. \\ \left. - \phi\left(\frac{z^+ - \mu_G}{\sigma_G}\right)\right] + \epsilon\left[\Phi\left(\frac{z^+ - \mu_G}{\sigma_G}\right) - \Phi\left(\frac{z^- - \mu_G}{\sigma_G}\right)\right] \end{aligned} \quad (17)$$

[6] Efficient Global Reliability Analysis for Nonlinear Implicit Performance Functions
B. J. Bichon, M. S. Eldred, L. P. Swiler, S. Mahadevan, and J. M. McFarland
AIAA Journal 2008 46:10, 2459-2468

What did I talk about today?

- Response Surface Methodology as a traditional approach to minimize experiments and processes
- Basic surrogate based optimization algorithm and issues
  - Training and evaluation of your surrogate should be orders of magnitude cheaper than your function evaluation
- Successive Response Surface Method (local optimizer)
- Efficient Global Optimization (global optimizer)
  - Most popular surrogate based optimization algorithm
  - Adaptative sampling technique
  - Many variants, including Efficient Global Reliability Analysis

Shape Optimization

- Objective: minimize displacement gap and contact pressure variance
- 2 Design Variables
- Implicit non-linear FE program
- Non-linear hyperplastic material model
- Contact: self, and rigid body

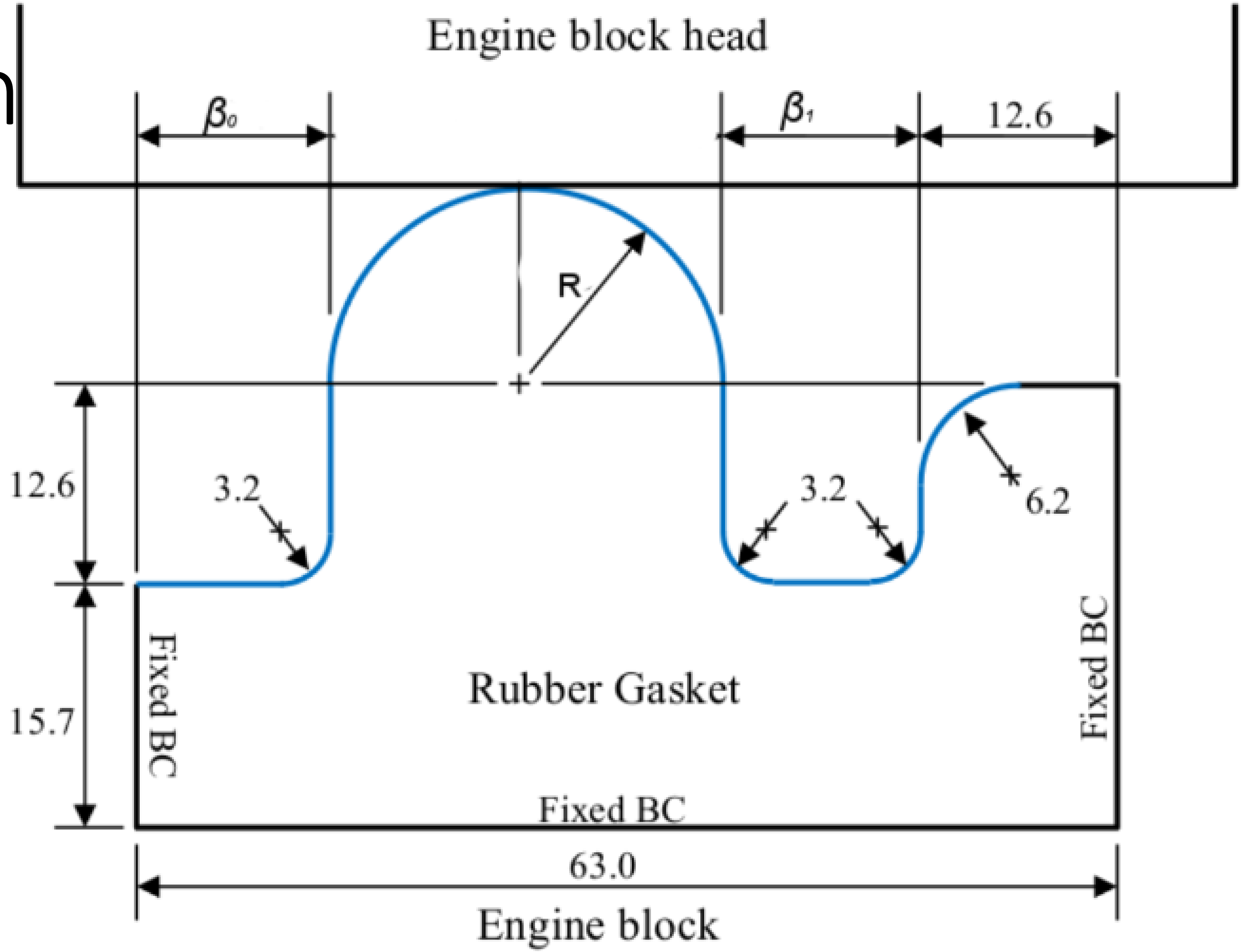
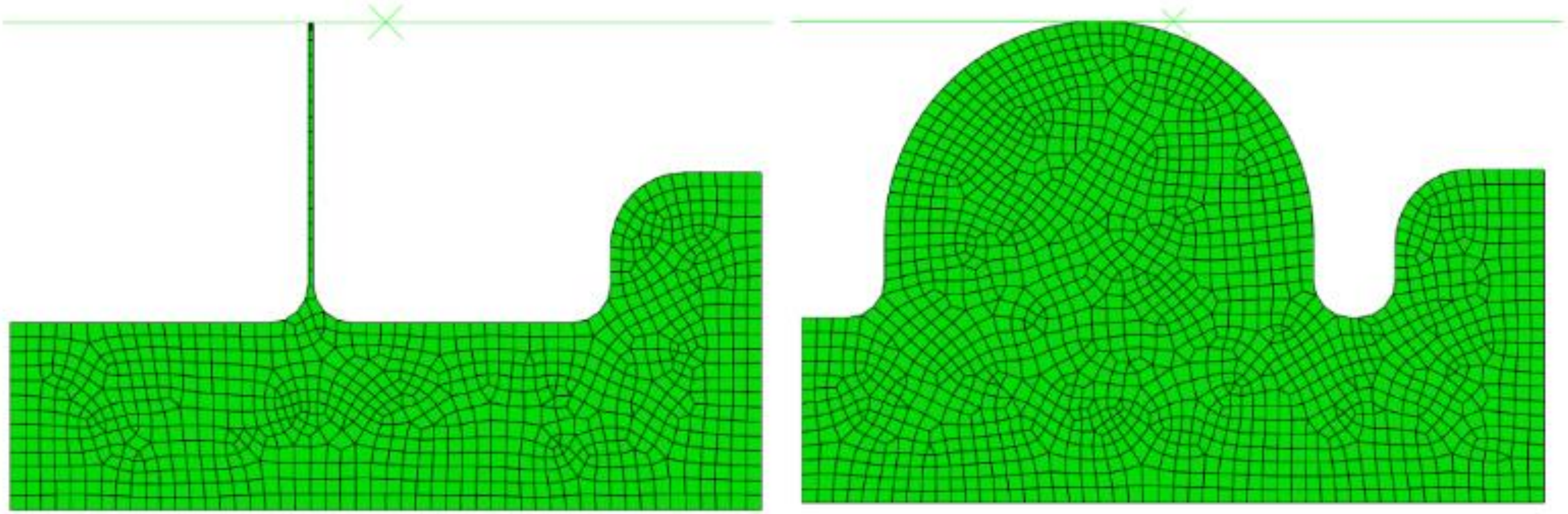


Figure 13: Parametric geometry (β_0, β_1, R) of the rubber engine gasket.

Different possible geometry with automatic meshing with nominal element size



(a) $\beta_0 = 0.025$ and $\beta_1 = 0.025$ (m) (b) $\beta_0 = 0.007$ and $\beta_1 = 0.007$ (m)

Figure 14: Two examples of possible meshed geometries provided β_0 and β_1.

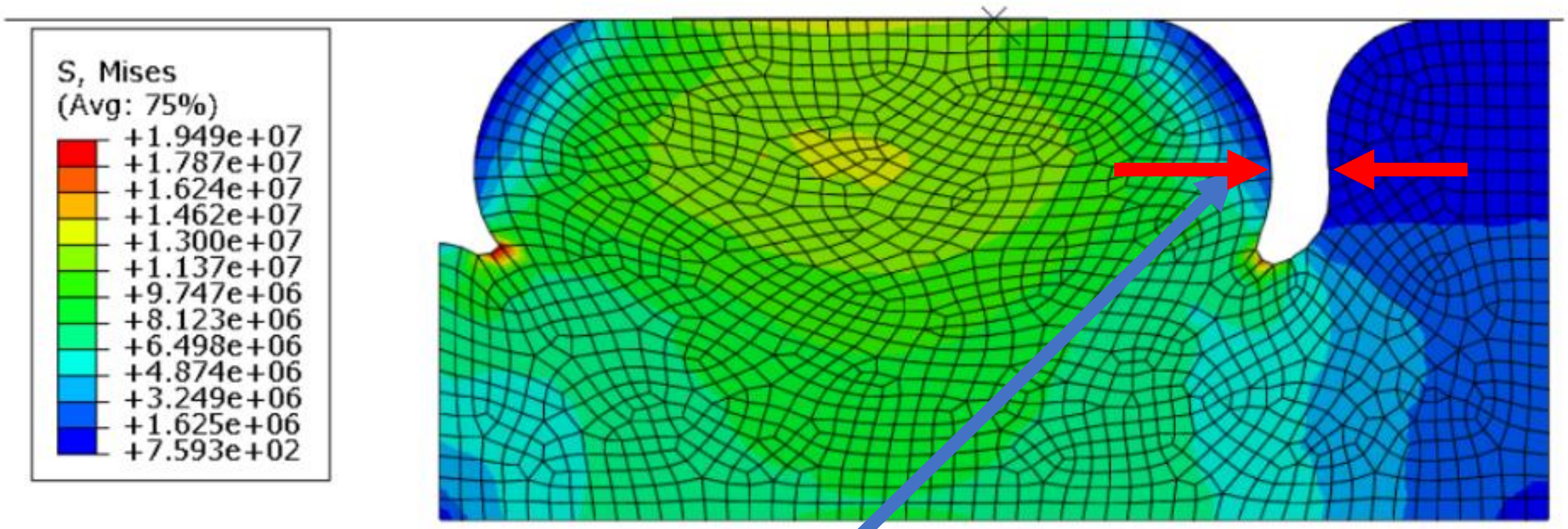


Figure 18: The von-Mieses stress of the final optimum design at the fully closed position.

$$\text{minimize:} \quad f(\boldsymbol{\beta}) = \left(\frac{d(\beta_0, \beta_1)}{d_0} + \frac{\sigma(\beta_0, \beta_1)}{\sigma_0} \right)$$

subject to: **Valid mesh generation and FEA run**

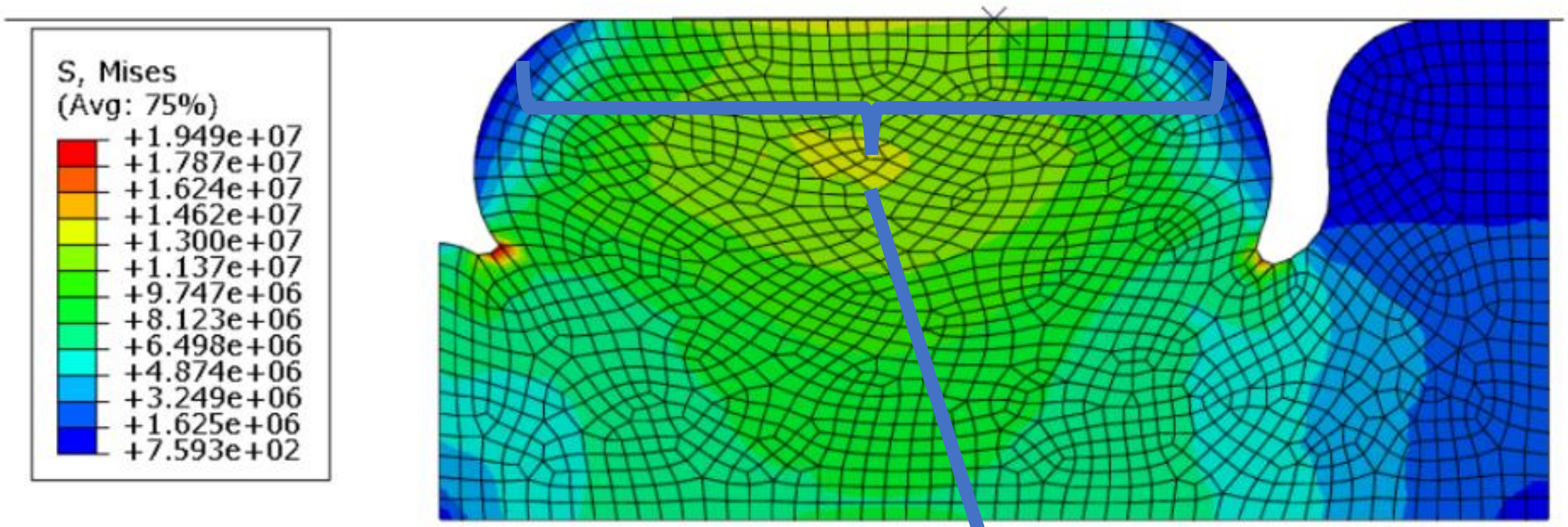


Figure 18: The von-Mieses stress of the final optimum design at the fully closed position.

$$\text{minimize: } f(\boldsymbol{\beta}) = \left(\frac{d(\beta_0,\beta_1)}{d_0} + \frac{\sigma(\beta_0,\beta_1)}{\sigma_0}\right)$$

subject to: **Valid mesh generation and FEA run**

Function Evaluation Procedure

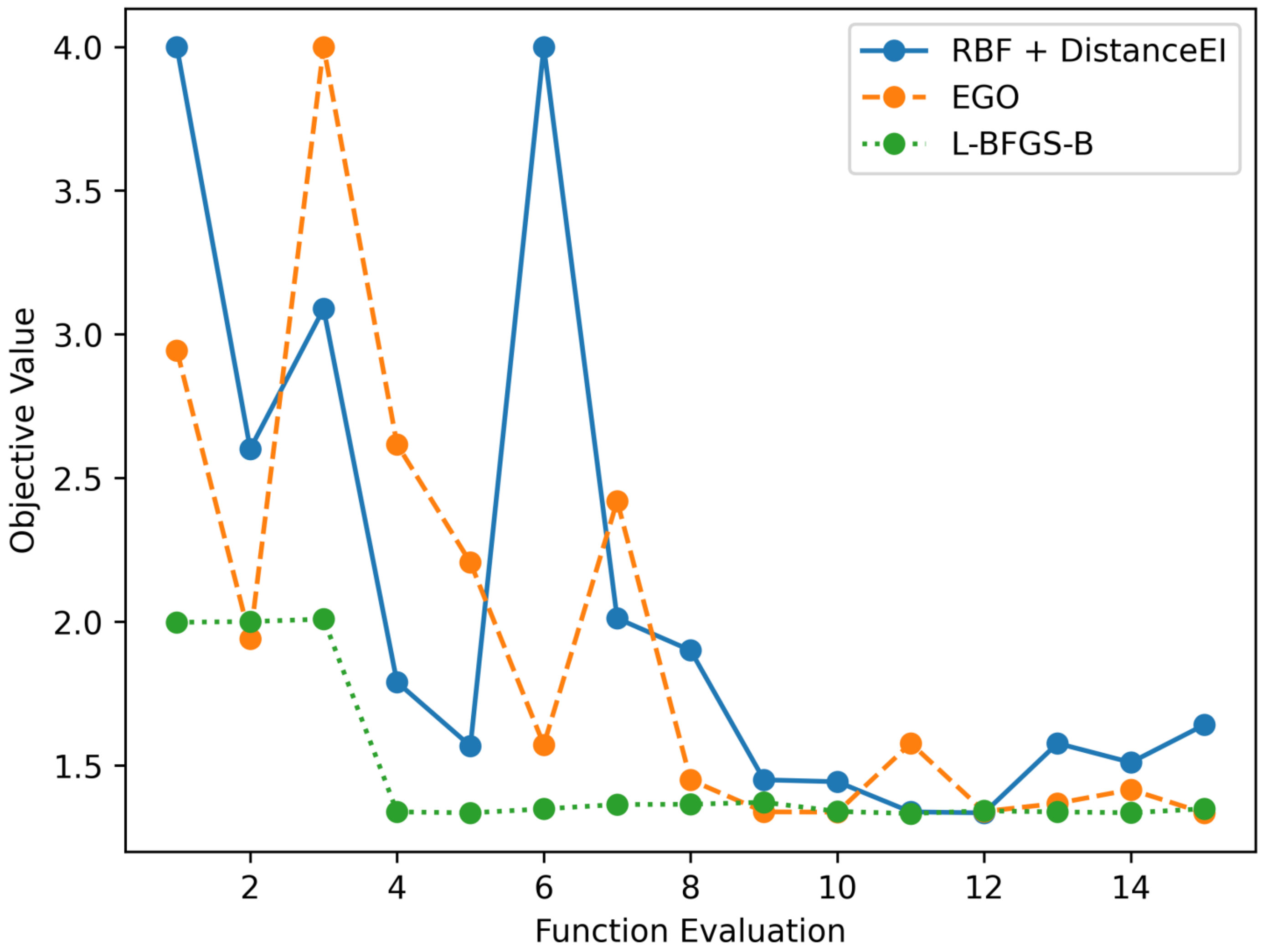
- Open Abaqus sketch and change dimensions
- Generate new mesh
- Write Abaqus input file and close Abaqus
- Submit Abaqus input file
- Verify that Abaqus solver completely successfully
- Run Abaqus post processing script to export distance and contract pressure
- Calculate objective function in Python from exported data

Run 3 different optimization algorithms

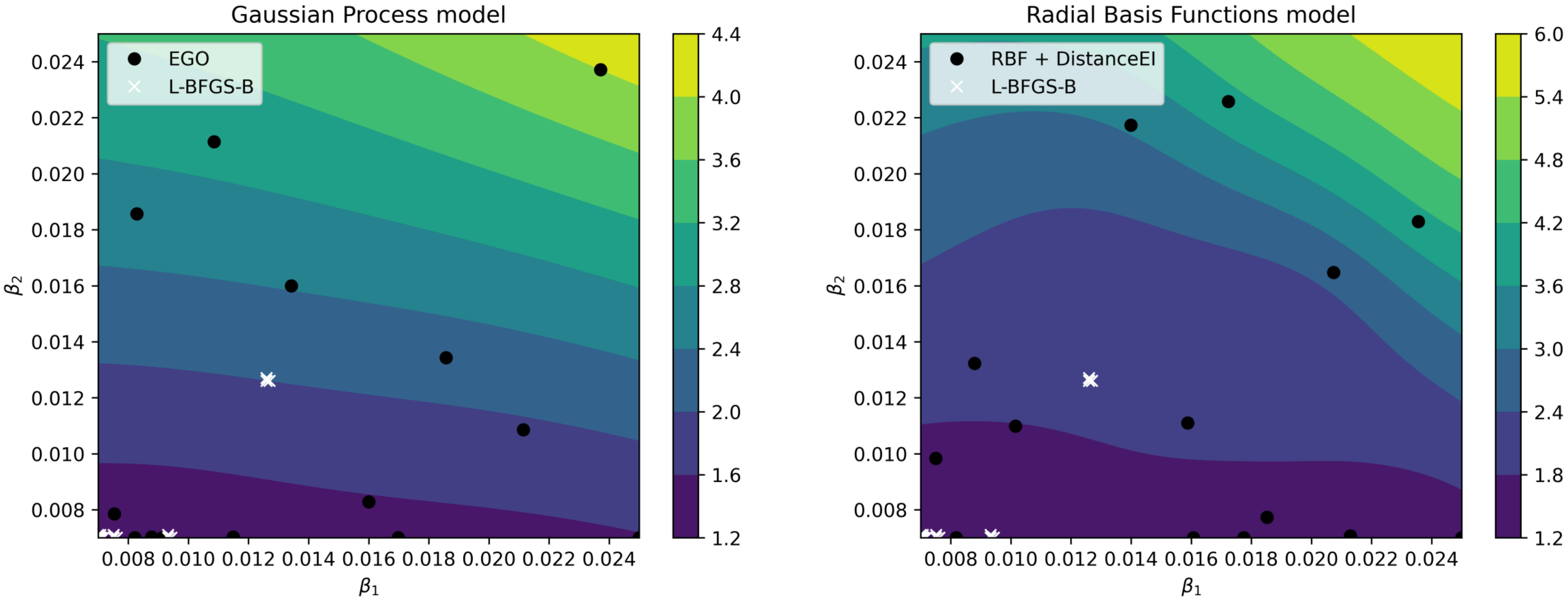
- L-BFGS-B
  - Popular local optimize
  - Uses finite differences to approximate gradients
  - Limit of 15 function evaluations
- Efficient Global Optimization (EGO)
  - Uses Gaussian Process model
  - 7 Latin Hypercube samples
  - 8 Optimization Iterations
- Radial Basis Function (RBF) Based Surrogated Optimization
  - Principles from today
  - 7 Latin Hypercube samples
  - 8 Optimization Iterations
  - https://github.com/cjekel/sbopt

Optimization results

- Function Evaluations against Objective Value
- Efficient Global Optimization (EGO) had best function value
- L-BFGS-B massive improvement after finite differences
- More or less all of these produce the same design



Resulting Surrogate Models



- The L-BFGS-B results wouldn't give us a good surrogate
- Use surrogate model to understand the design domain

There are many ways to solve a problem…

- Advantages of L-BFGS-B (gradient based optimizer)
  - Local convergence guarantees
  - Deterministic from starting point

- Advantages of the Surrogate Based Optimization
  - Creates database that can be used to further understand problem
  - Potential to re-use simulation results for another objective
  - Visualization of the optimization problem
  - Bypass local minimum

References

I used the GPyOpt implement of EGO in these slides [7].

[1] Nestor V. Queipo, Raphael T. Haftka, Wei Shyy, Tushar Goel, Rajkumar Vaidyanathan, P. Kevin Tucker, Surrogate-based analysis and optimization, Progress in Aerospace Sciences, Volume 41, Issue 1, 2005, Pages 1-28, ISSN 0376-0421, https://doi.org/10.1016/j.paerosci.2005.02.001

[2] Stander, N. and Craig, K.J. (2002), "On the robustness of a simple domain reduction scheme for simulation-based optimization", *Engineering Computations*, Vol. 19 No. 4, pp. 431-450. https://doi.org/10.1108/02644400210430190

[3] Goel, T., Haftka, R.T., Shyy, W. and Watson, L.T. (2008), Pitfalls of using a single criterion for selecting experimental designs. Int. J. Numer. Meth. Engng., 75: 127-155. doi:10.1002/nme.2242

[4] Jones, D.R., Schonlau, M. & Welch, W.J. Efficient Global Optimization of Expensive Black-Box Functions. Journal of Global Optimization 13, 455–492 (1998). https://doi.org/10.1023/A:1008306431147

[5] Marzougui D, Kan CD, Opiela KS. Crash test & simulation comparisons of a pickup truck & a small car oblique impacts into a concrete barrier. InThe 13th International LS-DYNA Users Conference, Dearborn, MI 2013.

[6] Efficient Global Reliability Analysis for Nonlinear Implicit Performance FunctionsB. J. Bichon, M. S. Eldred, L. P. Swiler, S. Mahadevan, and J. M. McFarland AIAA Journal 2008 46:10, 2459-2468

[7] González J, Dai Z. GPyOpt: a Bayesian optimization framework in Python. https://github.com/SheffieldML/GPyOpt