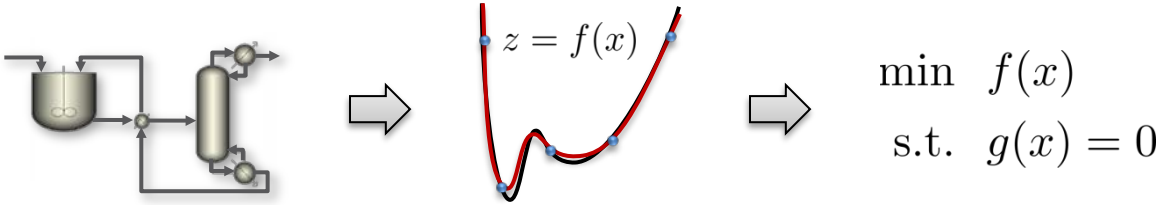


Learning Surrogate Models of Processes From Experiments or Simulations

Alison Cozad[1] Nick Sahinidis[1] David Miller[2]

[1]Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA
[2]US DOE-National Energy Technology laboratory, Morgantown, WV

OBJECTIVE: PROCESS SYNTHESIS

Black box function
(simulation, experiment, etc.)

Ideally

Derivative-based optimization

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0 \end{aligned}$$

OBJECTIVE: PROCESS SYNTHESIS

Black box function
(simulation, experiment, etc.)

Derivative-based optimization

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0 \end{aligned}$$

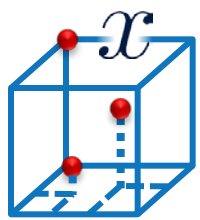
- **Lack of an algebraic model**
- **Computationally costly simulations/experiments**
- **Scarcity of fully robust simulations/experiments**

OBJECTIVE: PROCESS SYNTHESIS

Black box function
(simulation, experiment, etc.)

Derivative-based optimization

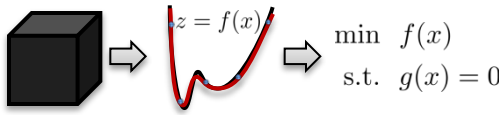
$$\min \quad f(x)$$
$$\text{s.t.} \quad g(x) = 0$$



Data

ALAMO

Automated **L**earning of **A**lgebraic **M**odels for **O**ptimization



$$z_1 = g_1(x)$$
$$z_2 = g_2(x)$$
$$\dots$$
$$z_k = g_k(x)$$

Models

MODELING PROBLEM STATEMENT

- **Build a model of output variables z as a function of input variables x over a specified interval**

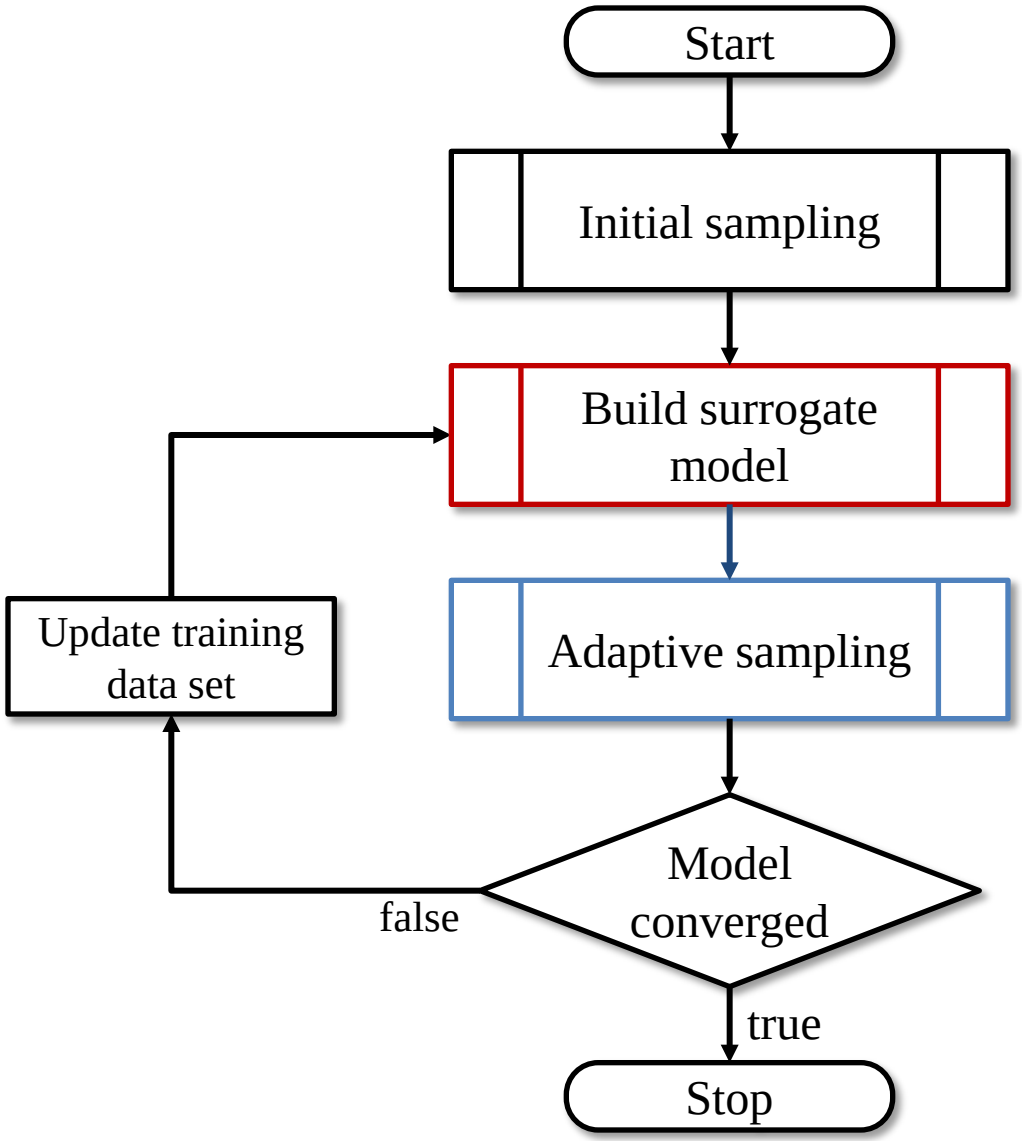
$$x \in \mathbb{R}^D \qquad x^l \leq x \leq x^u \qquad \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \\ \vdots \\ x_D \end{pmatrix} \Rightarrow \boxed{\text{Black box function, simulation, or experiment}} \Rightarrow \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_k \\ \vdots \\ z_K \end{pmatrix} \qquad z \in \mathbb{R}^K \qquad z = f(x)$$

Independent variables:
Operating conditions, Inlet flow properties, Simulation or experimental geometry, Parameters, etc.

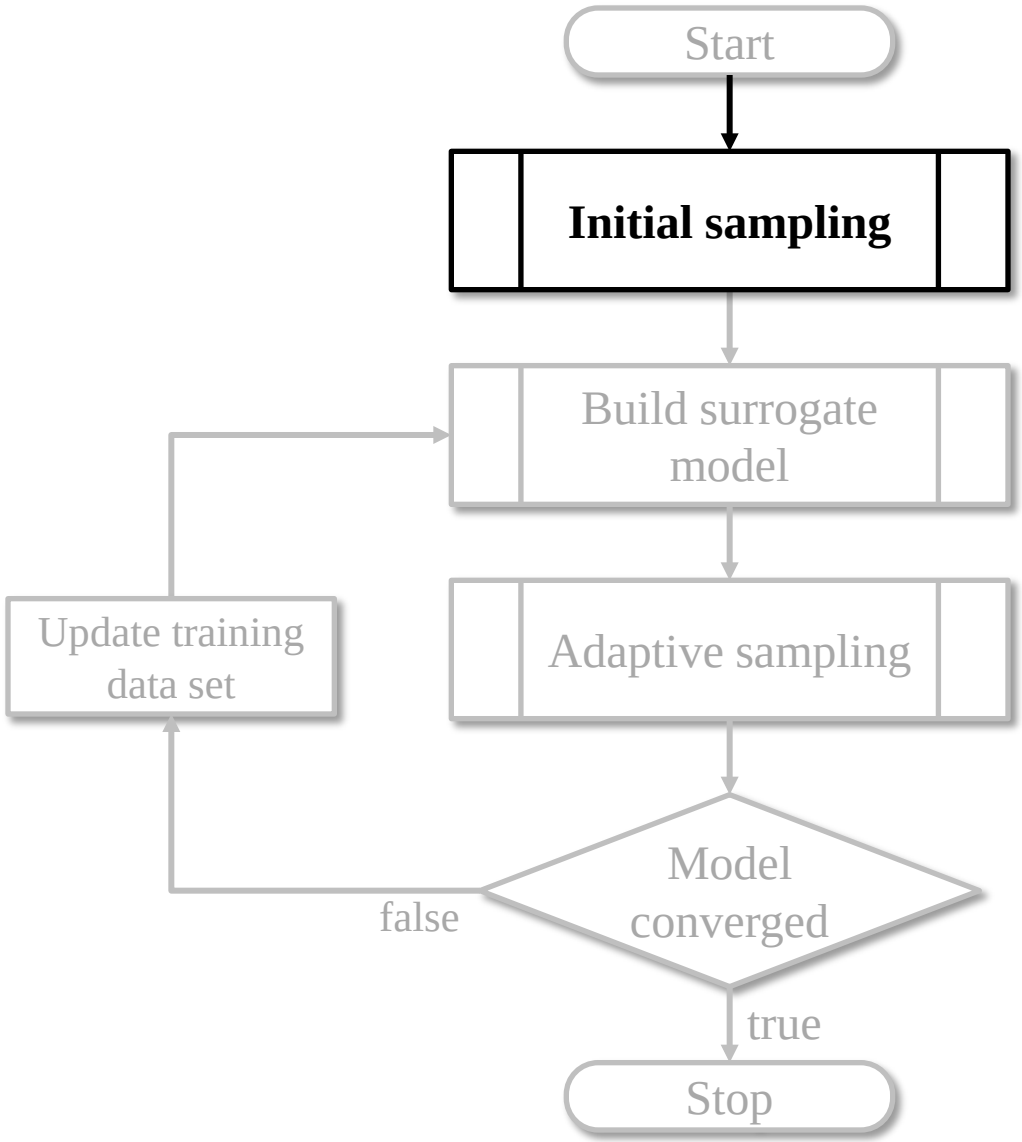
Dependent variables:
Efficiency, Outlet flow conditions, Conversions, Heat flow, etc.

- **Desired model traits:**
  - ✓ **Accurate**
  - ✓ **Tractable in algebraic optimization: Simple functional forms**
  - ✓ **Generated from a minimal data set**

ALGORITHMIC FLOWSHEET



ALGORITHMIC FLOWSHEET



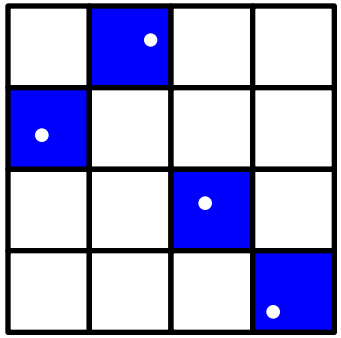
DESIGN OF EXPERIMENTS

- **Goal: To generate an initial set of input variables to evenly sample the problem space**

$$x = \begin{pmatrix} x^1 & x^2 & \cdots & x^i & \cdots & x^N \end{pmatrix} \qquad x^i = \begin{pmatrix} x_1^i \\ x_2^i \\ \vdots \\ x_d^i \\ \vdots \\ x_D^i \end{pmatrix}$$

- **Design of experiments: Latin hypercube design**
  - **Space-filling design**

2D
D=2



INITIAL SAMPLING

- **After running the design of experiments, we will evaluate the black-box function to determine each z^i**

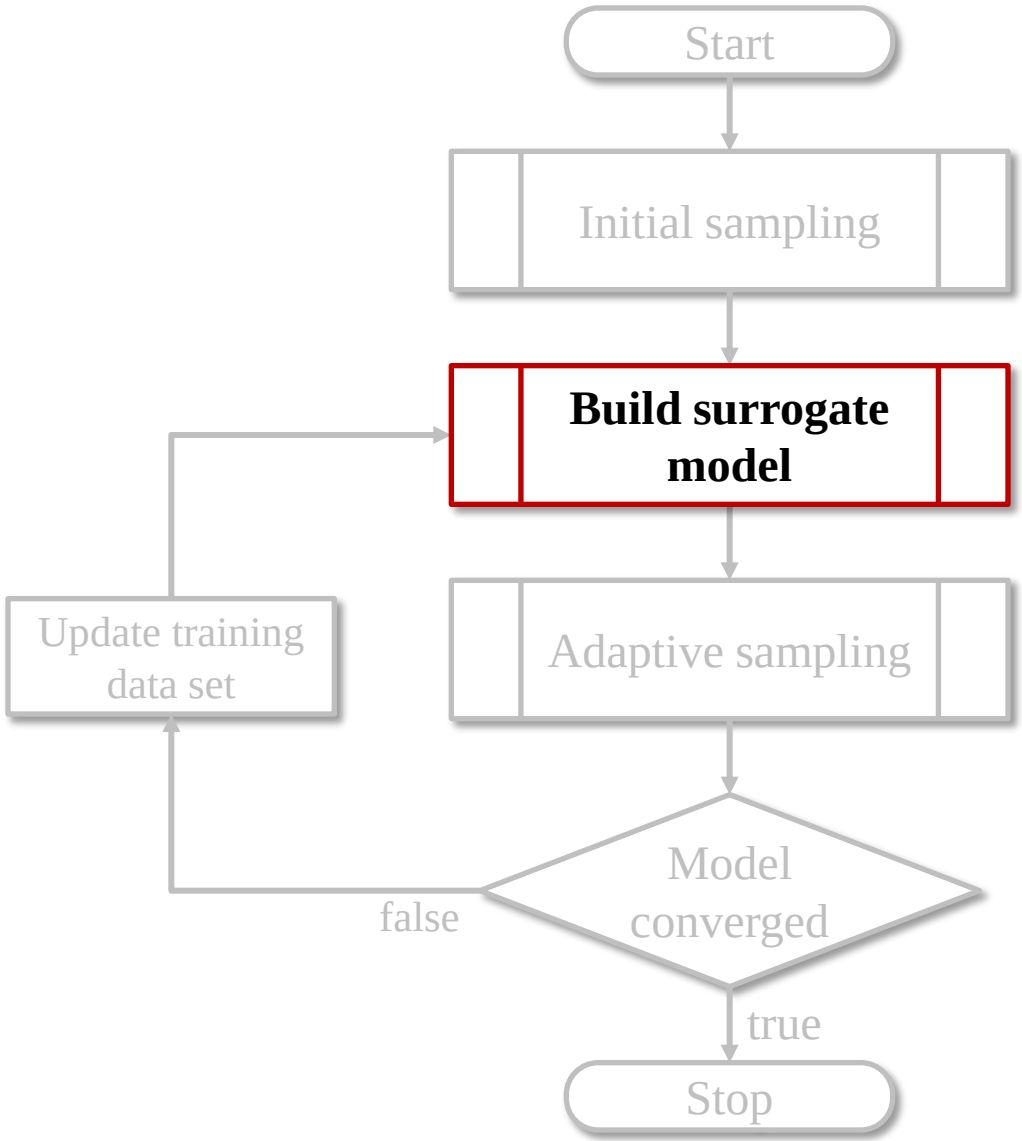
$$x = \begin{pmatrix} x^1 & x^2 & \cdots & x^i & \cdots & x^N \end{pmatrix}$$

Black-box function, simulation, or experiment

$$z = \begin{pmatrix} z^1 & z^2 & \cdots & z^i & \cdots & z^N \end{pmatrix}$$

Initial training set

ALGORITHMIC FLOWSHEET



MODEL IDENTIFICATION

- **Goal: Identify the functional form and complexity of the surrogate models**

$$z = f(x)$$

- **Functional form:**
  - **General functional form is unknown: Our method will identify models with combinations of simple basis functions of the following forms**

	Category	$X_j(x)$
I.	Polynomial	$(x_d)^\alpha$
II.	Multinomial	$\prod_{d=1}^{m} (x_d)^{\alpha_d}$, for $m = 1, 2, \ldots$
III.	Exponential and logarithmic forms	$\exp\left(\frac{x_d}{\gamma}\right)^\alpha$, $\log\left(\frac{x_d}{\gamma}\right)^\alpha$
IV.	Expected bases	From experience, simple inspection, etc.

  - **Surrogate model will have the form:**

$$\hat{z} = \sum_{j \in \mathcal{B}} \beta_j X_j(x)$$

PART A: FIND A FUNCTIONAL FORM

- **Let's assume we know *a priori* the complexity or number of terms in the model, T**

- **Identify the optimal T subset of basis functions such that we**
  - ✓ **Minimize model error**
  - ✓ **Ensure that only T terms are present in the model**
  - ✓ **Enforce the linear least squares condition for the chosen T basis functions**

Ordinary linear least squares problem:

$$\min_{\beta} \sum_{i=1}^{N} \left(z^i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right)^2$$

where X_{ij} is the basis function j at the black-box data point $i = 1, 2, \ldots, N$

SELECTING THE BEST T SUBSET

Linear least squares regression	$$\min_{\beta} \sum_{i=1}^{N} \left(z^i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right)^2$$

SELECTING THE BEST T SUBSET

Linear least squares regression

$$\min_{\beta} \sum_{i=1}^{N} \left(z^i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right)^2$$

$\mathcal{S}_h \subset \mathcal{B}$, such that $|\mathcal{S}_h| = T$

Solve for a subset of bases

$h = 1$: $\min_{\beta} \sum_{i=1}^{N} \left(z^i - \sum_{j \in \mathcal{S}_1} \beta_j X_{ij} \right)^2$, $h = 2$: $\min_{\beta} \sum_{i=1}^{N} \left(z^i - \sum_{j \in \mathcal{S}_2} \beta_j X_{ij} \right)^2$, $h = 3$: $\min_{\beta} \sum_{i=1}^{N} \left(z^i - \sum_{j \in \mathcal{S}_3} \beta_j X_{ij} \right)^2$, ...

SELECTING THE BEST T SUBSET

Linear least squares regression

$$\min_{\beta} \sum_{i=1}^{N} \left(z^i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right)^2$$

$\mathcal{S}_h \subset \mathcal{B}$, such that $|\mathcal{S}_h| = T$

Solve for a subset of bases

$$\min_{h} \left(\underbrace{\min_{\beta} \sum_{i=1}^{N} \left(z^i - \sum_{j \in \mathcal{S}_1} \beta_j X_{ij} \right)^2}_{h=1}, \underbrace{\min_{\beta} \sum_{i=1}^{N} \left(z^i - \sum_{j \in \mathcal{S}_2} \beta_j X_{ij} \right)^2}_{h=2}, \underbrace{\min_{\beta} \sum_{i=1}^{N} \left(z^i - \sum_{j \in \mathcal{S}_3} \beta_j X_{ij} \right)^2}_{h=3}, \ldots \right)$$

SELECTING THE BEST T SUBSET

Linear least squares regression

$$\min_{\beta} \sum_{i=1}^{N} \left(z^i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right)^2$$

$\mathcal{S}_h \subset \mathcal{B}$, such that $|\mathcal{S}_h| = T$

Solve for a subset of bases

$$\min_{h} \left(\underbrace{\min_{\beta} \sum_{i=1}^{N} \left(z^i - \sum_{j \in \mathcal{S}_1} \beta_j X_{ij} \right)^2}_{h=1}, \underbrace{\min_{\beta} \sum_{i=1}^{N} \left(z^i - \sum_{j \in \mathcal{S}_2} \beta_j X_{ij} \right)^2}_{h=2}, \underbrace{\min_{\beta} \sum_{i=1}^{N} \left(z^i - \sum_{j \in \mathcal{S}_3} \beta_j X_{ij} \right)^2}_{h=3}, \ldots \right)$$

Substitute closed form solution

$$\min_{h} \sum_{i=1}^{N} \left(z^i - \sum_{j \in \mathcal{S}_h} \beta_j X_{ij} \right)^2$$

$$\text{s.t.} \sum_{i=1}^{N} X_{ij} \left(z^i - \sum_{j \in \mathcal{S}_1} \beta_j X_{ij} \right) = 0, \sum_{i=1}^{N} X_{ij} \left(z^i - \sum_{j \in \mathcal{S}_2} \beta_j X_{ij} \right) = 0, \sum_{i=1}^{N} X_{ij} \left(z^i - \sum_{j \in \mathcal{S}_3} \beta_j X_{ij} \right) = 0, \ldots$$

SELECTING THE BEST T SUBSET

Solve for the best subset of basis functions

$$\min_{h} \quad \sum_{i=1}^{N} \left(z^i - \sum_{j \in \mathcal{S}_h} \beta_j X_{ij} \right)^2$$

$$\text{s.t.} \quad \sum_{i=1}^{N} X_{ij} \left(z^i - \sum_{j \in \mathcal{S}_h} \beta_j X_{ij} \right) = 0, \quad j \in \mathcal{S}_h, \ \forall \mathcal{S}_h \subset \mathcal{B}$$

BEST T SUBSET MINIMIZATION MODEL

Choose terms to minimize the model error

$$\min \quad SE = \sum_{i=1}^{N} \left| z_i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right|$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{B}} y_j = T$$

$$-U(1-y_j) \leq \sum_{i=1}^{N} X_{ij} \left(z^i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right) \leq U(1-y_j) \qquad j \in \mathcal{B}$$

$$\beta^l y_j \leq \beta_j \leq \beta^u y_j \qquad j \in \mathcal{B}$$

$$y_j = \{0, 1\} \qquad j \in \mathcal{B}$$

BEST T SUBSET MINIMIZATION MODEL

Choose terms to minimize the model error	$\min \quad SE = \sum_{i=1}^{N} \left\lvert z_i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right\rvert$
Exactly *T* terms in the model	$\text{s.t.} \quad \sum_{j \in \mathcal{B}} y_j = T$

$$-U(1-y_j) \leq \sum_{i=1}^{N} X_{ij} \left(z^i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right) \leq U(1-y_j) \qquad j \in \mathcal{B}$$

$$\beta^l y_j \leq \beta_j \leq \beta^u y_j \qquad j \in \mathcal{B}$$

$$y_j = \{0, 1\} \qquad j \in \mathcal{B}$$

BEST T SUBSET MINIMIZATION MODEL

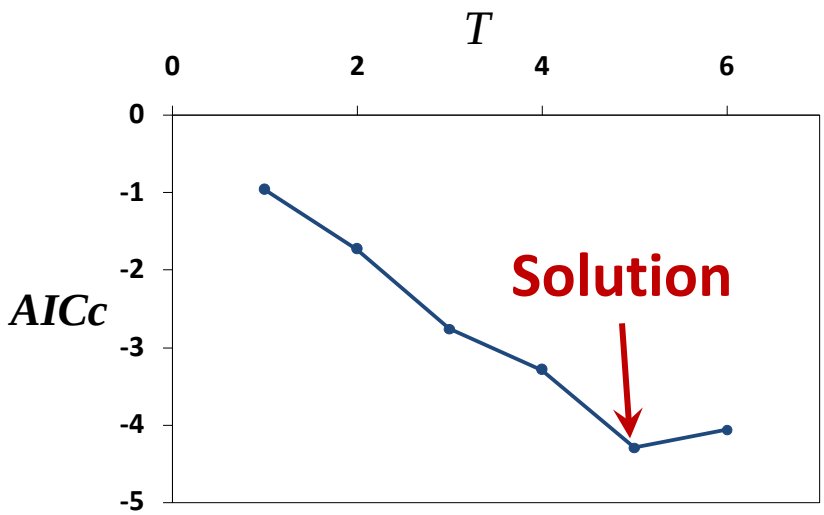
Choose terms to minimize the model error	$\min \quad SE = \sum_{i=1}^{N} \left\| z_i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right\|$
Exactly *T* terms in the model	$\text{s.t.} \quad \sum_{j \in \mathcal{B}} y_j = T$
Defining the subset of basis functions used	$-U(1-y_j) \leq \sum_{i=1}^{N} X_{ij} \left(z^i - \sum_{j \in \mathcal{B}} \beta_j X_{ij} \right) \leq U(1-y_j) \qquad j \in \mathcal{B}$ $\beta^l y_j \leq \beta_j \leq \beta^u y_j \qquad j \in \mathcal{B}$ $y_j = \{0, 1\} \qquad j \in \mathcal{B}$

$$\left(\begin{array}{c} \mathbf{y_j = 1} \\ \\ \textbf{Basis function used in the model} \\ \beta_j \text{ is chosen to satisfy a least} \\ \text{squares regression} \\ \textit{(assumes loose bounds on } \beta_j\textit{)} \end{array} \right) \vee \left(\begin{array}{c} \mathbf{y_j = 0} \\ \\ \textbf{Basis function NOT used} \\ \textbf{in the model} \\ \\ \beta_j = 0 \end{array} \right)$$

PART B: FIND MODEL COMPLEXITY

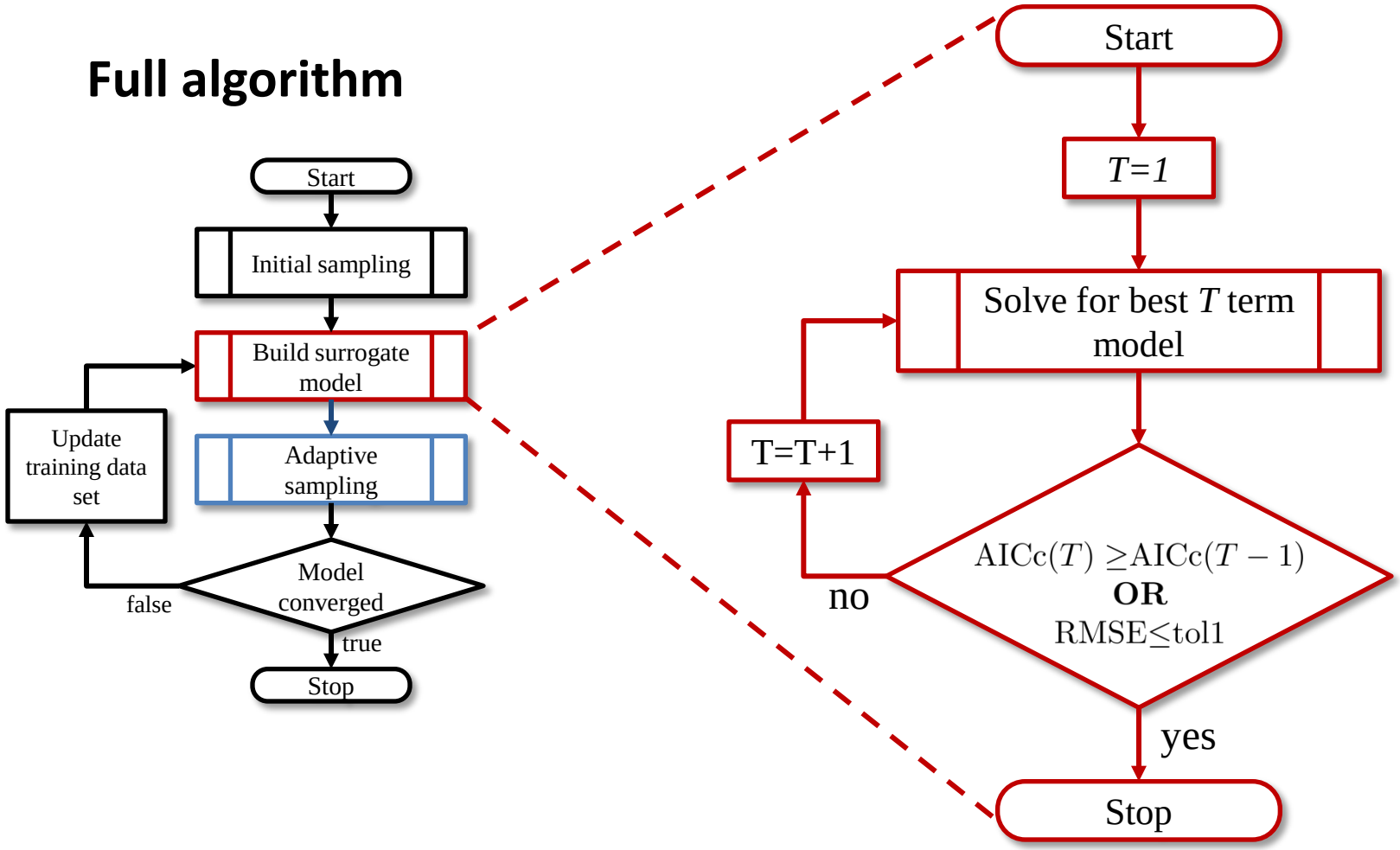
- **In reality, we don't know T *a priori***
- **How to find T:**
  - **Corrected Akaike Information Criterion ($AICc$)**
    - ***Gives an estimate of the difference between a model and the true function***

$$AICc = \underbrace{N \log\left(\frac{SSE}{N}\right)}_{\text{Accuracy}} + \underbrace{2T + \frac{2T\,(T+1)}{n-T-1}}_{\text{Complexity}}$$

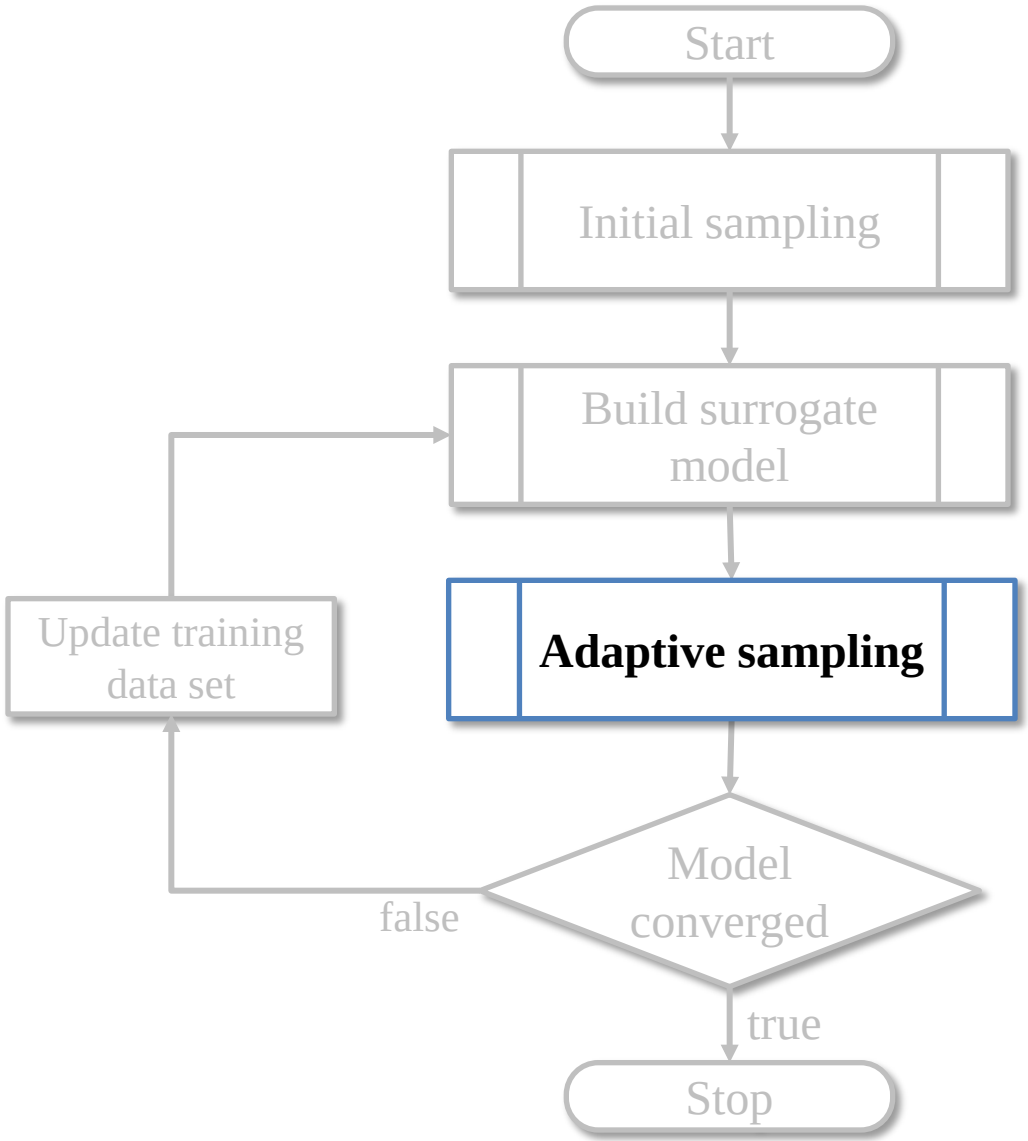


BUILD MODEL FLOWSHEET

Full algorithm



ALGORITHMIC FLOWSHEET



ADAPTIVE SAMPLING

- **Goal: Search the problem space for areas of model inconsistency or model mismatch**

- **More succinctly, we are trying to find points that maximize the model error with respect to the independent variables**

Surrogate model

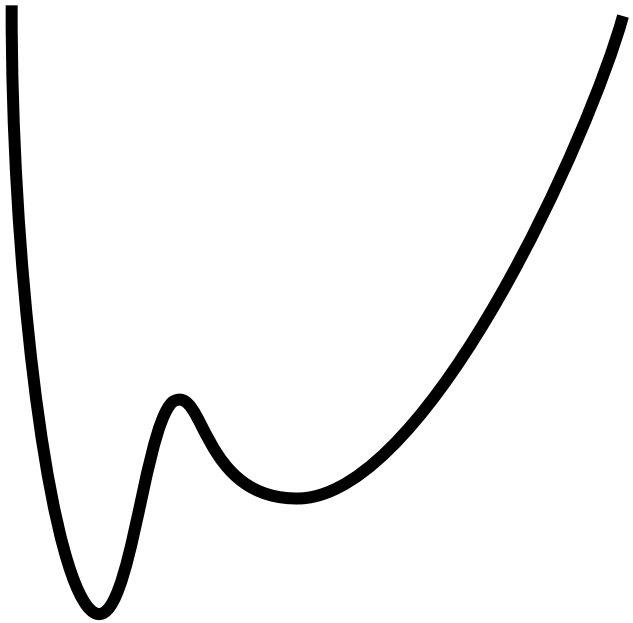
$$\max_{x} \left(\frac{z(x) - \hat{z}(x)}{z(x)} \right)^2$$

  – **Optimized using a black-box or derivative-free solver (SNOBFIT)**

ADAPTIVE SAMPLING

- **Goal: Search the problem space for areas of model inconsistency or model mismatch**

- **More succinctly, we are trying to find points that maximizes the model error with respect to the independent variables**

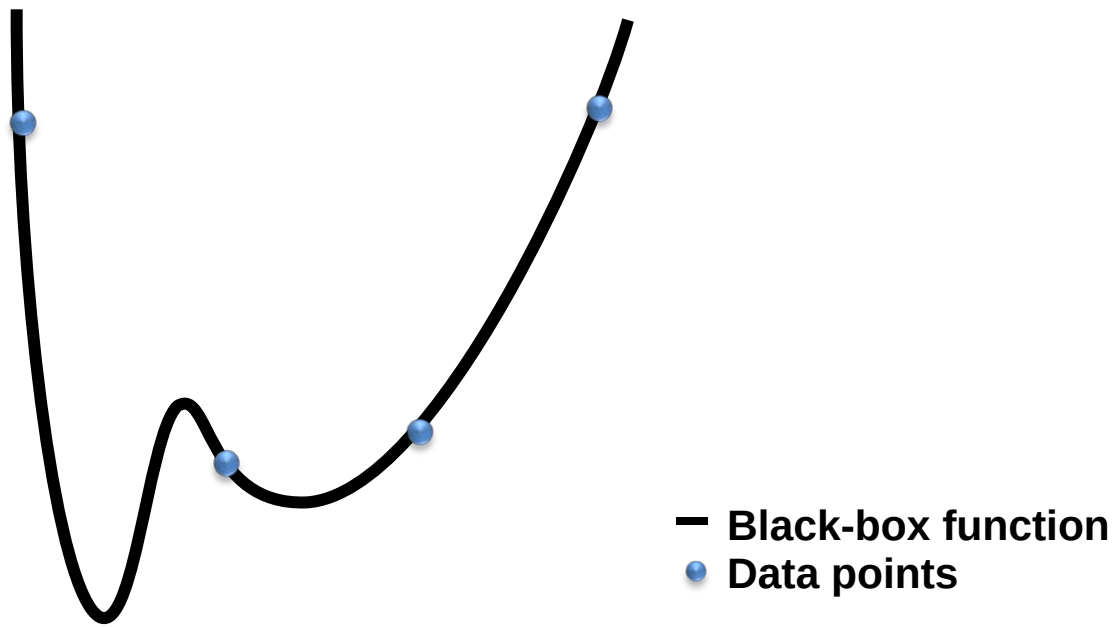


– Black-box function

ADAPTIVE SAMPLING

- **Goal: Search the problem space for areas of model inconsistency or model mismatch**

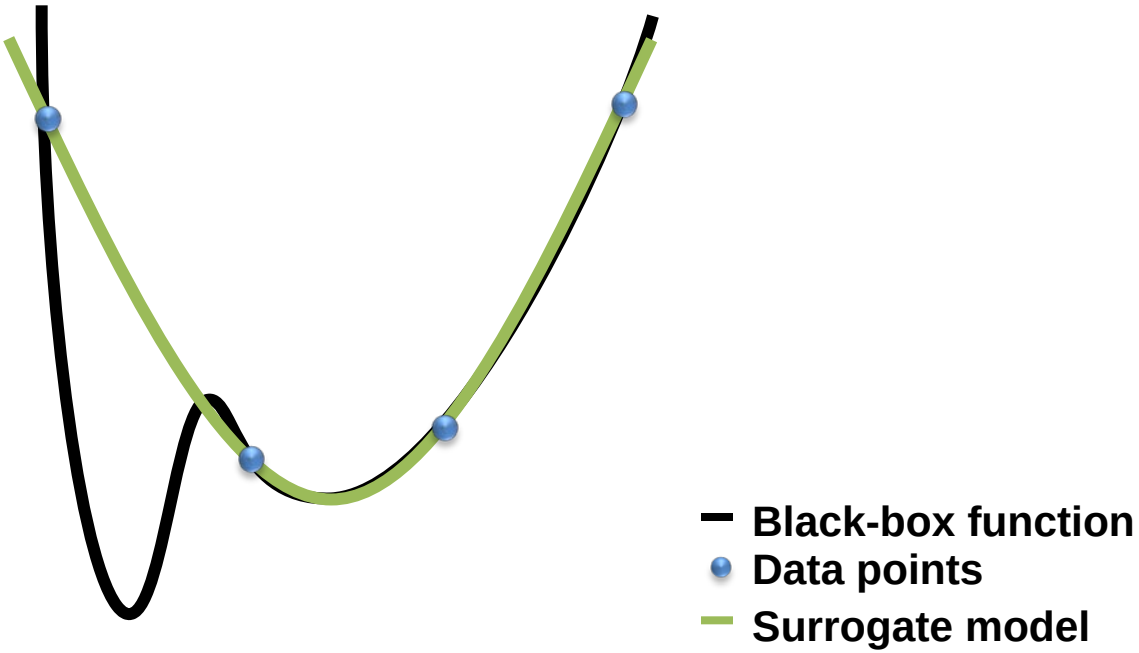
- **More succinctly, we are trying to find points that maximizes the model error with respect to the independent variables**



ADAPTIVE SAMPLING

- **Goal: Search the problem space for areas of model inconsistency or model mismatch**

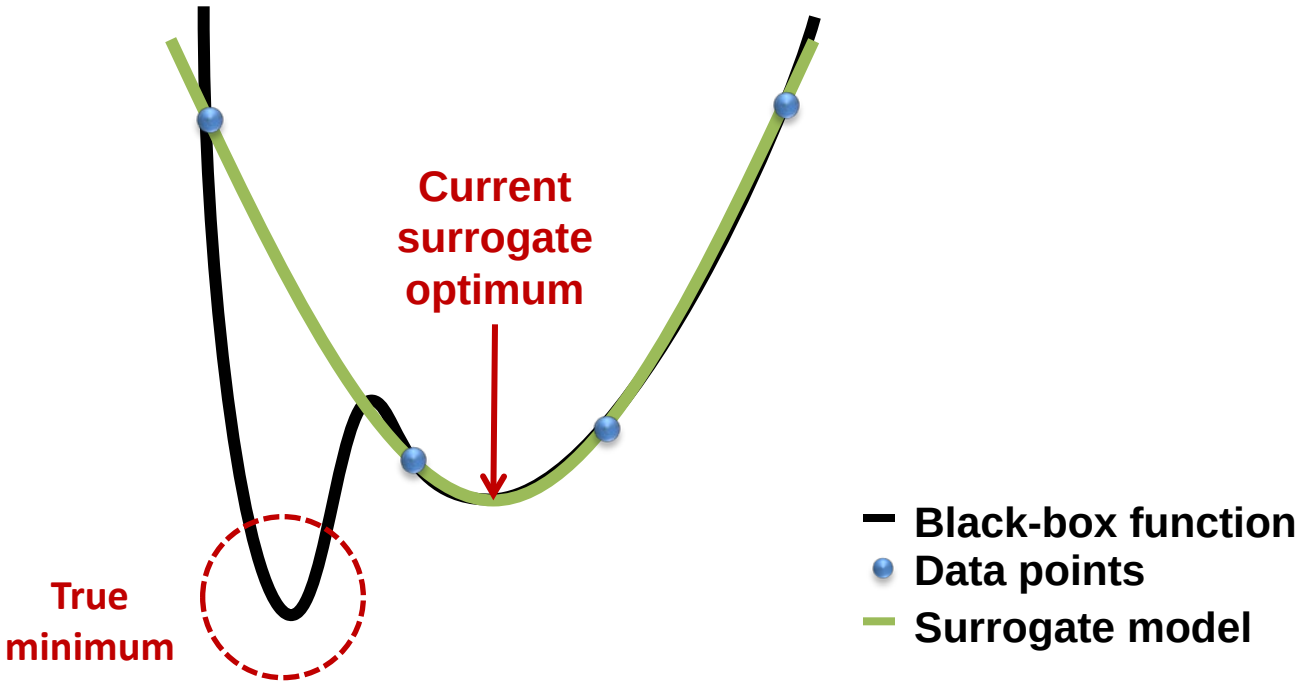
- **More succinctly, we are trying to find points that maximizes the model error with respect to the independent variables**



ADAPTIVE SAMPLING

- **Goal: Search the problem space for areas of model inconsistency or model mismatch**

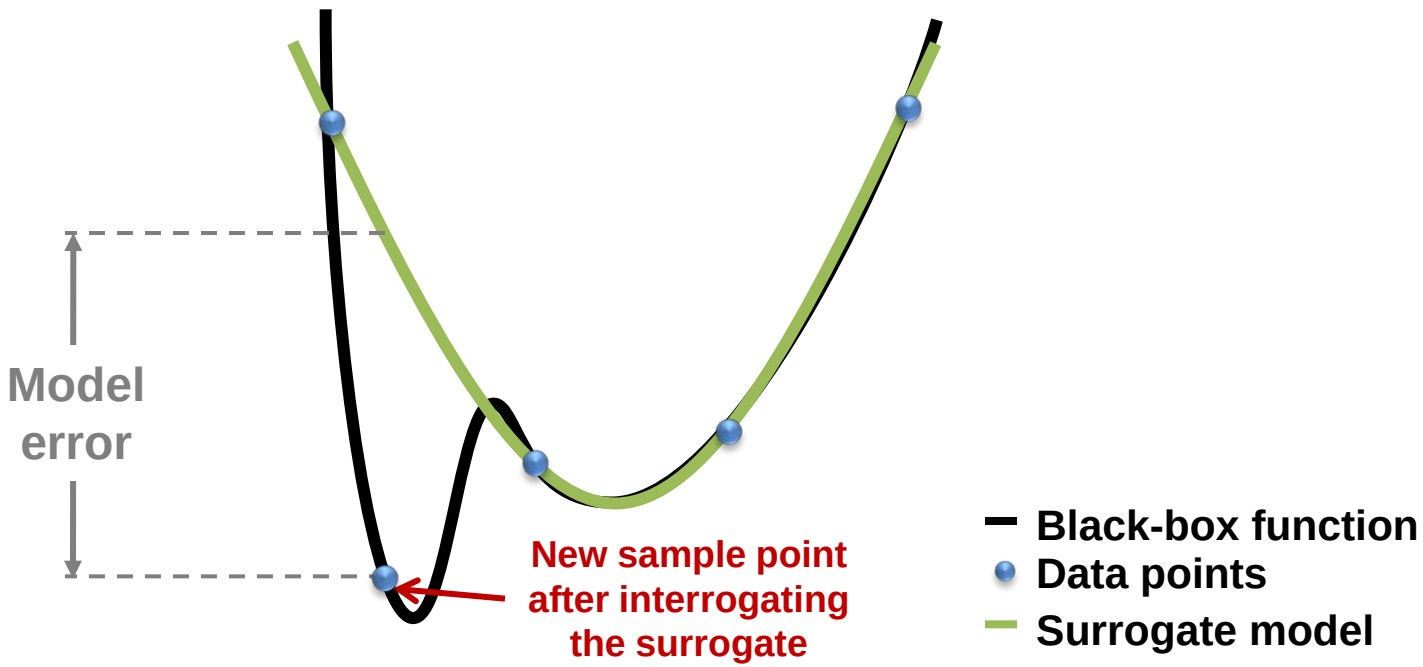
- **More succinctly, we are trying to find points that maximizes the model error with respect to the independent variables**



ADAPTIVE SAMPLING

- **Goal: Search the problem space for areas of model inconsistency or model mismatch**

- **More succinctly, we are trying to find points that maximizes the model error with respect to the independent variables**

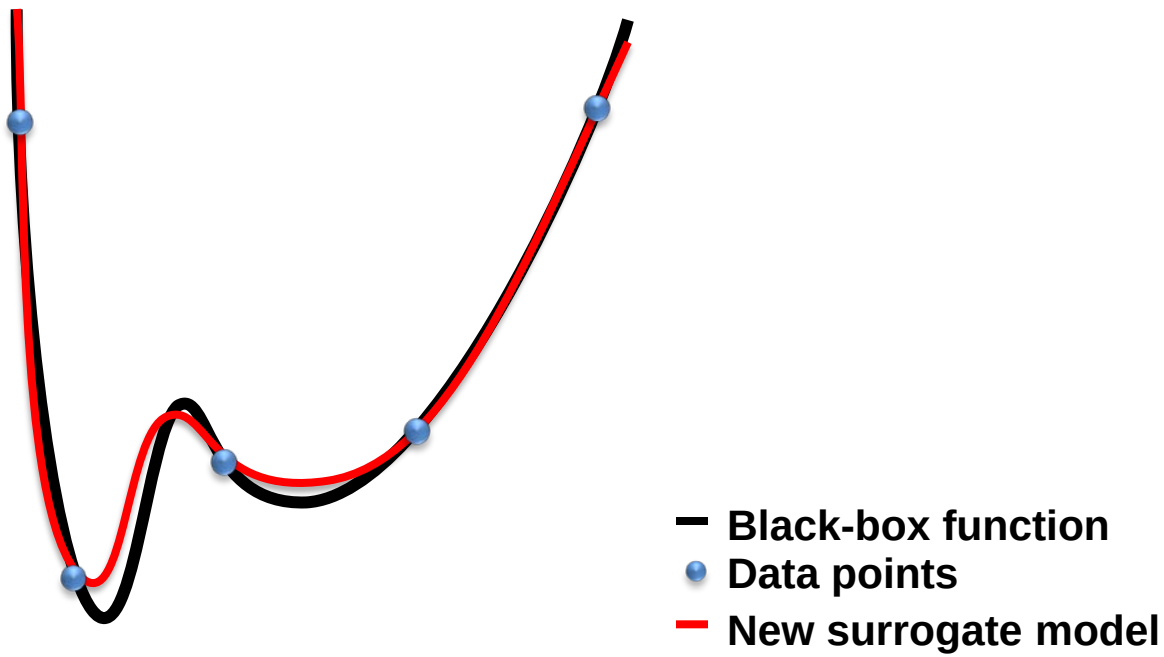


ADAPTIVE SAMPLING

- **Goal: Search the problem space for areas of model inconsistency or model mismatch**

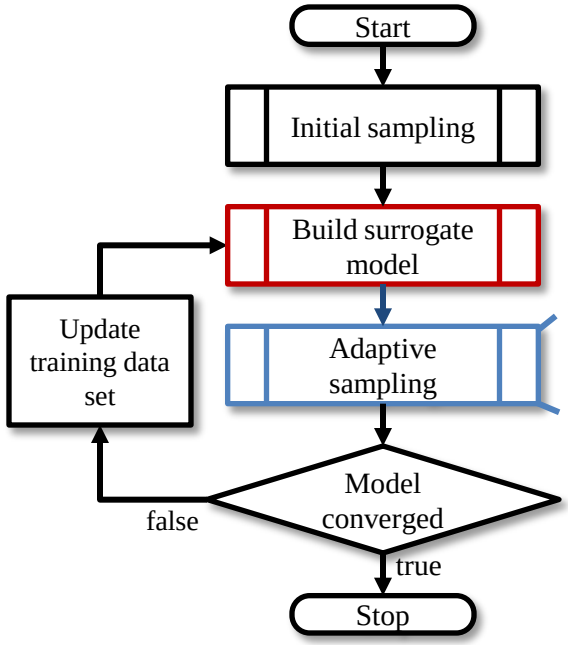
- **More succinctly, we are trying to find points that maximizes the model error with respect to the independent variables**

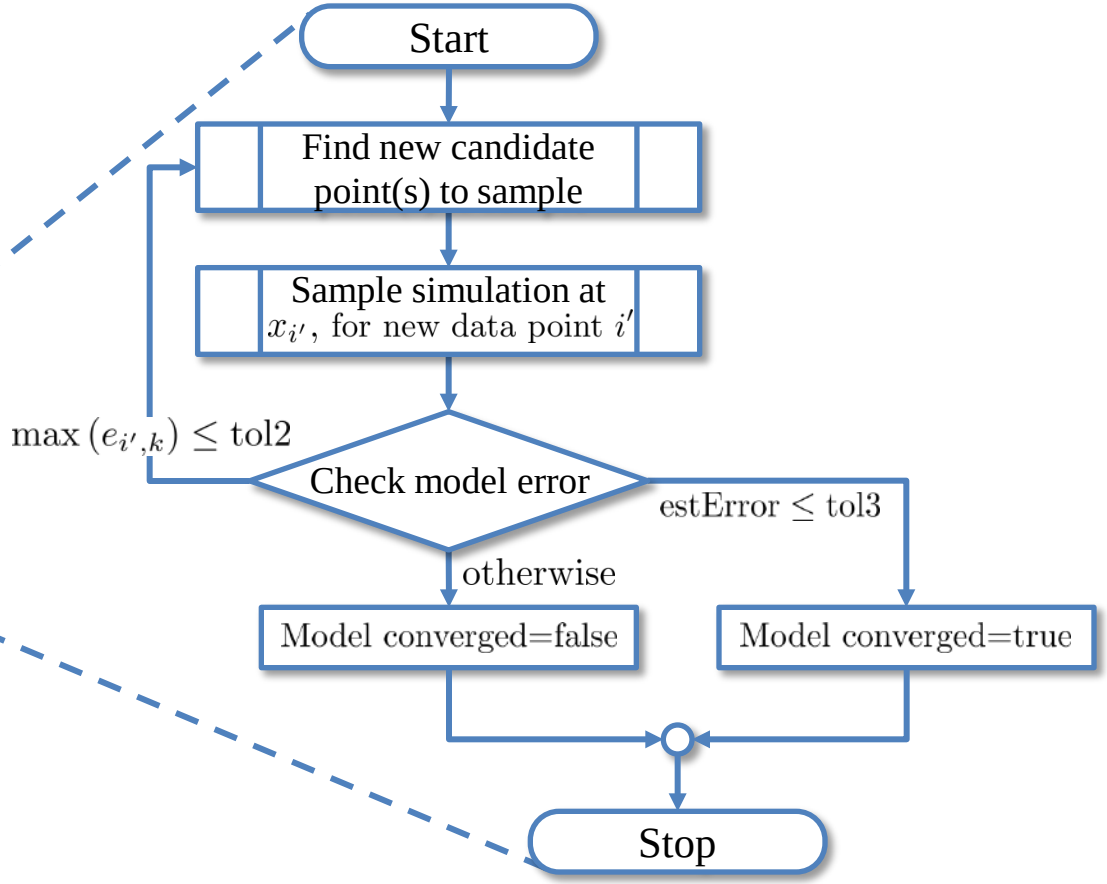
Reducing the order of the model is not always desirable



ADAPTIVE SAMPLING FLOWSHEET

Full algorithm

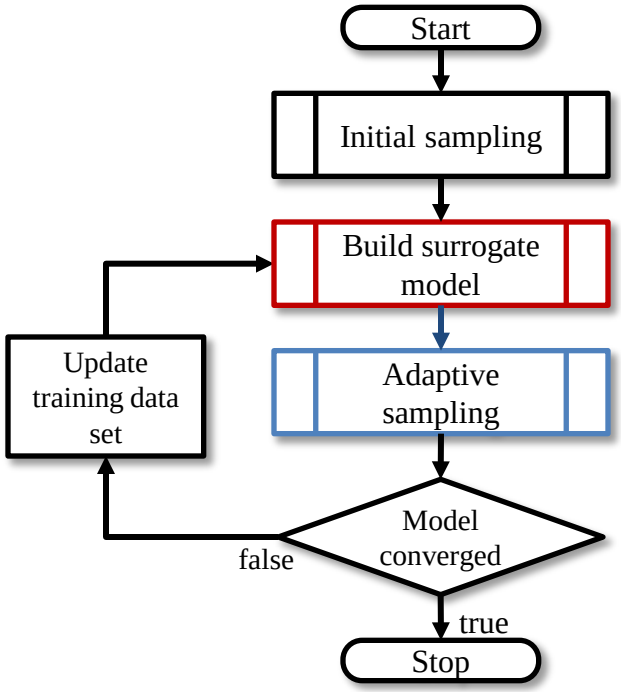




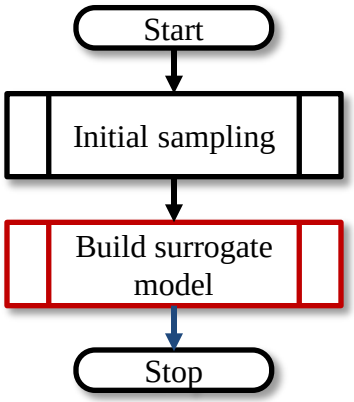
ACCURACY VALIDATION

- **Compare the method with and without adaptive sampling:**

Full algorithm:

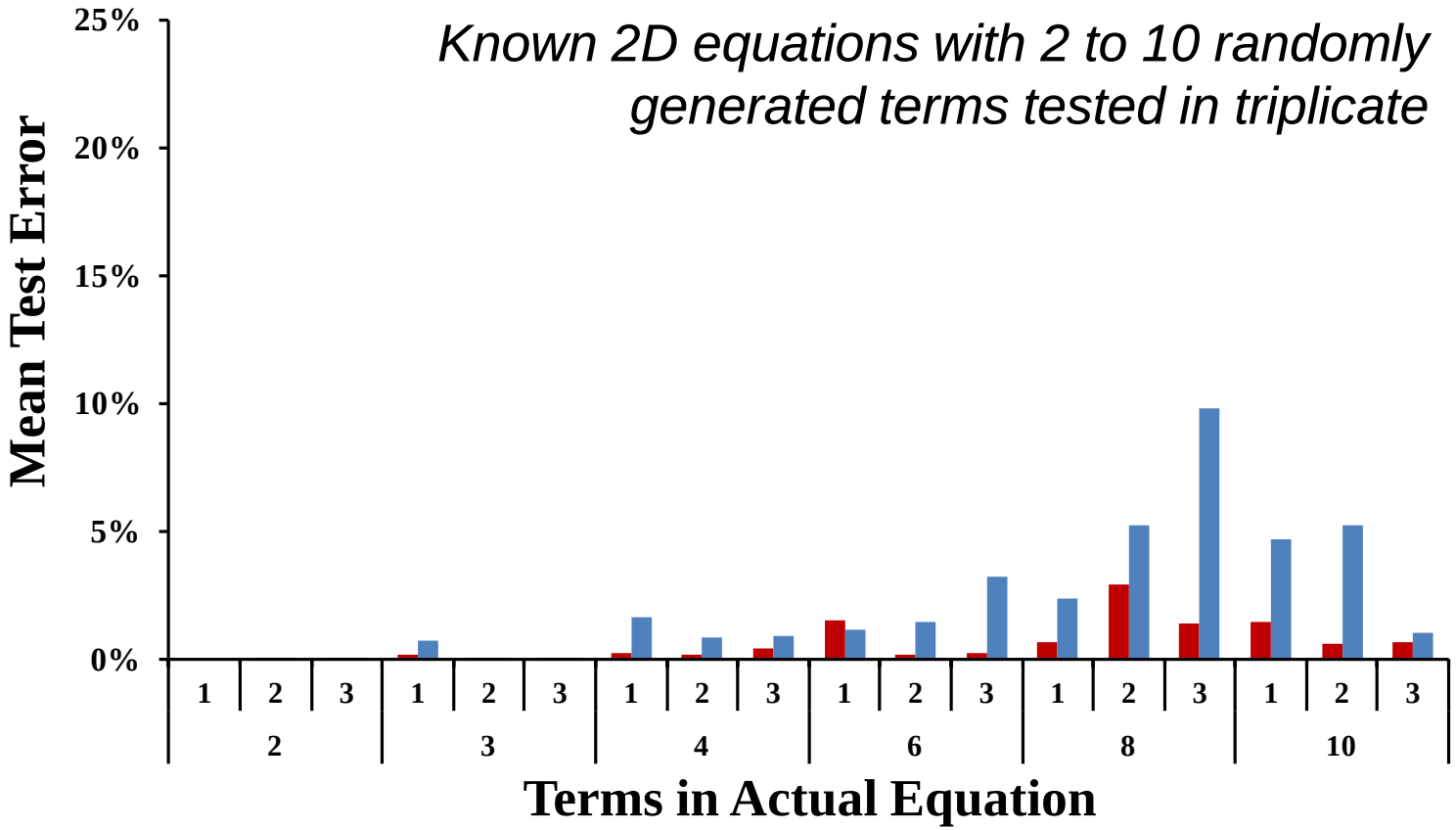


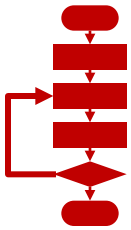
Single Latin hypercube:



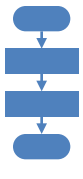
- **Two sets of known equations made up of functions**
  - **Present in the algorithm's basis set**
  - **Not present in the algorithm's basis set**

FUNCTIONS PRESENT IN THE BASIS



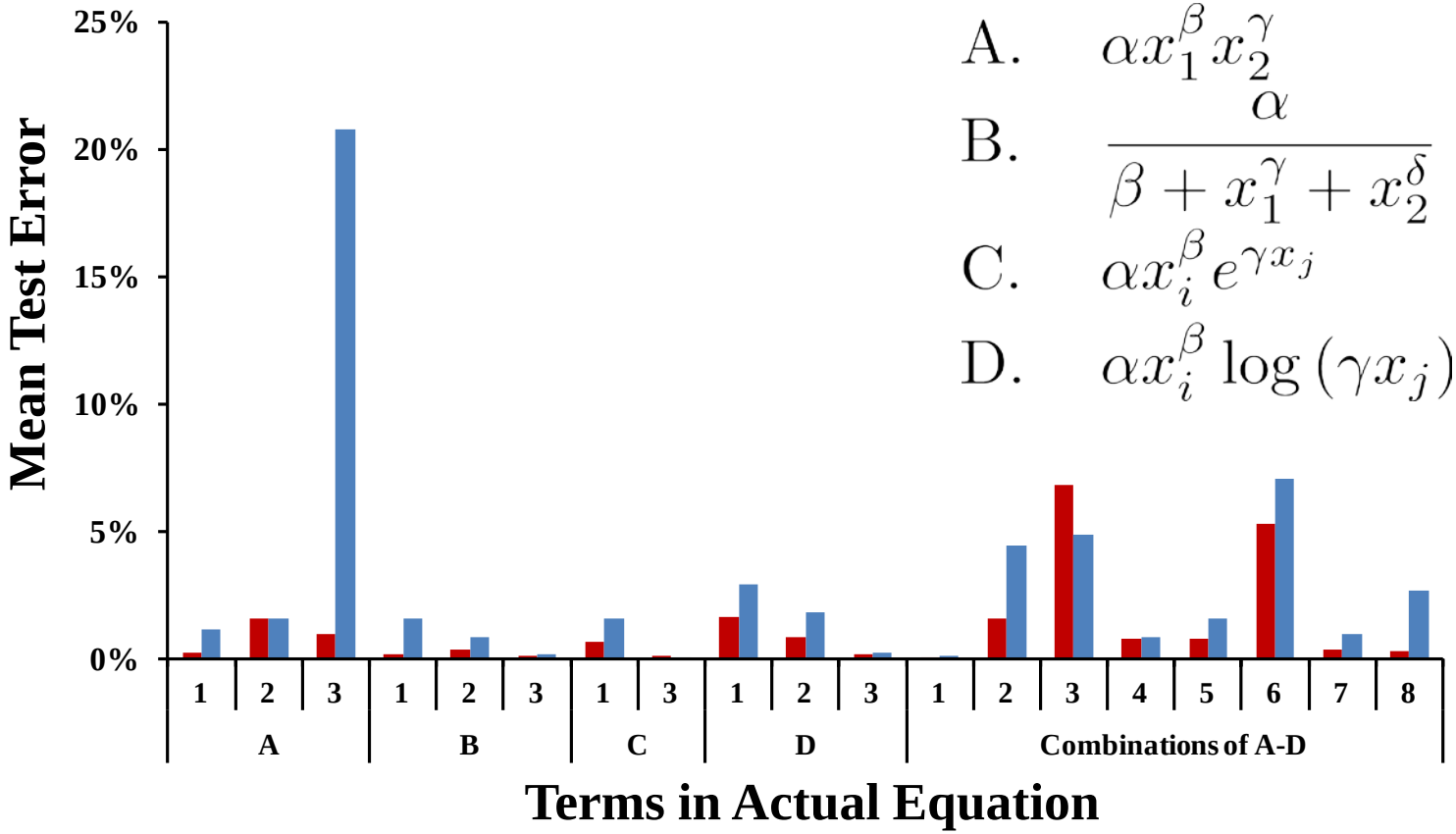


With adaptive sampling feedback using *N* points



Without feedback using *N* points (only Latin hypercube sampling)

FUNCTIONS NOT IN THE BASIS

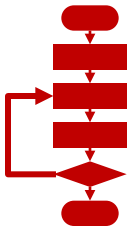


A. $\alpha x_1^{\beta} x_2^{\gamma}$

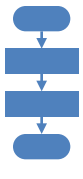
B. $\dfrac{\alpha}{\beta + x_1^{\gamma} + x_2^{\delta}}$

C. $\alpha x_i^{\beta} e^{\gamma x_j}$

D. $\alpha x_i^{\beta} \log(\gamma x_j)$

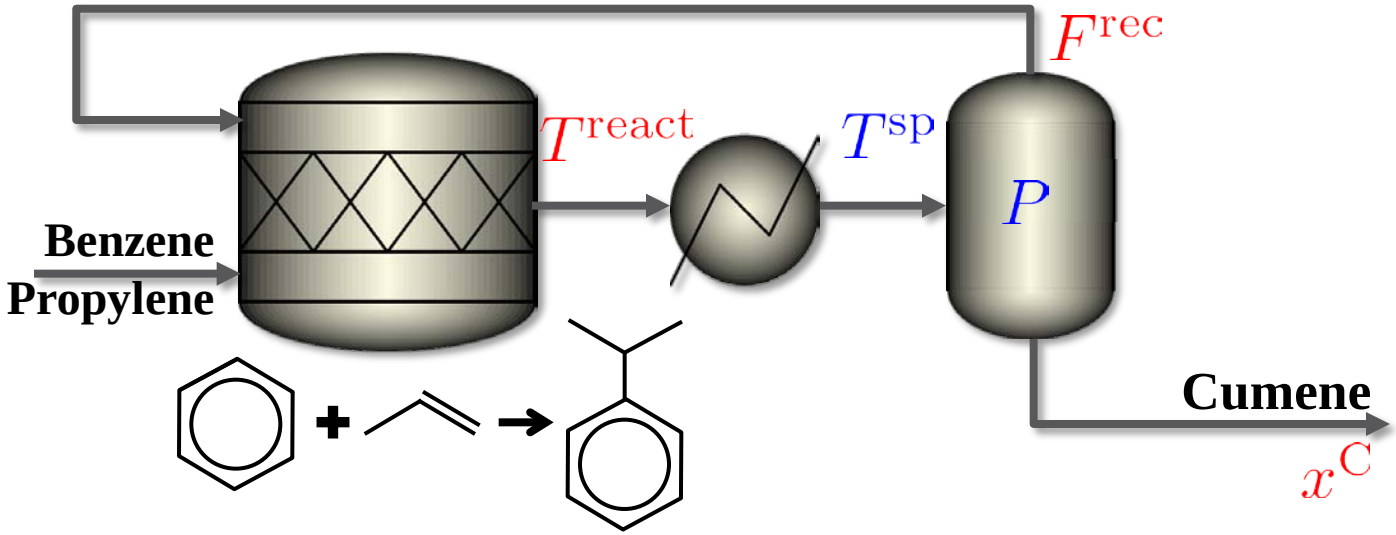


With adaptive sampling feedback using *N* points



Without feedback using *N* points (only Latin hypercube sampling)

TEST CASE: CUMENE PRODUCTION



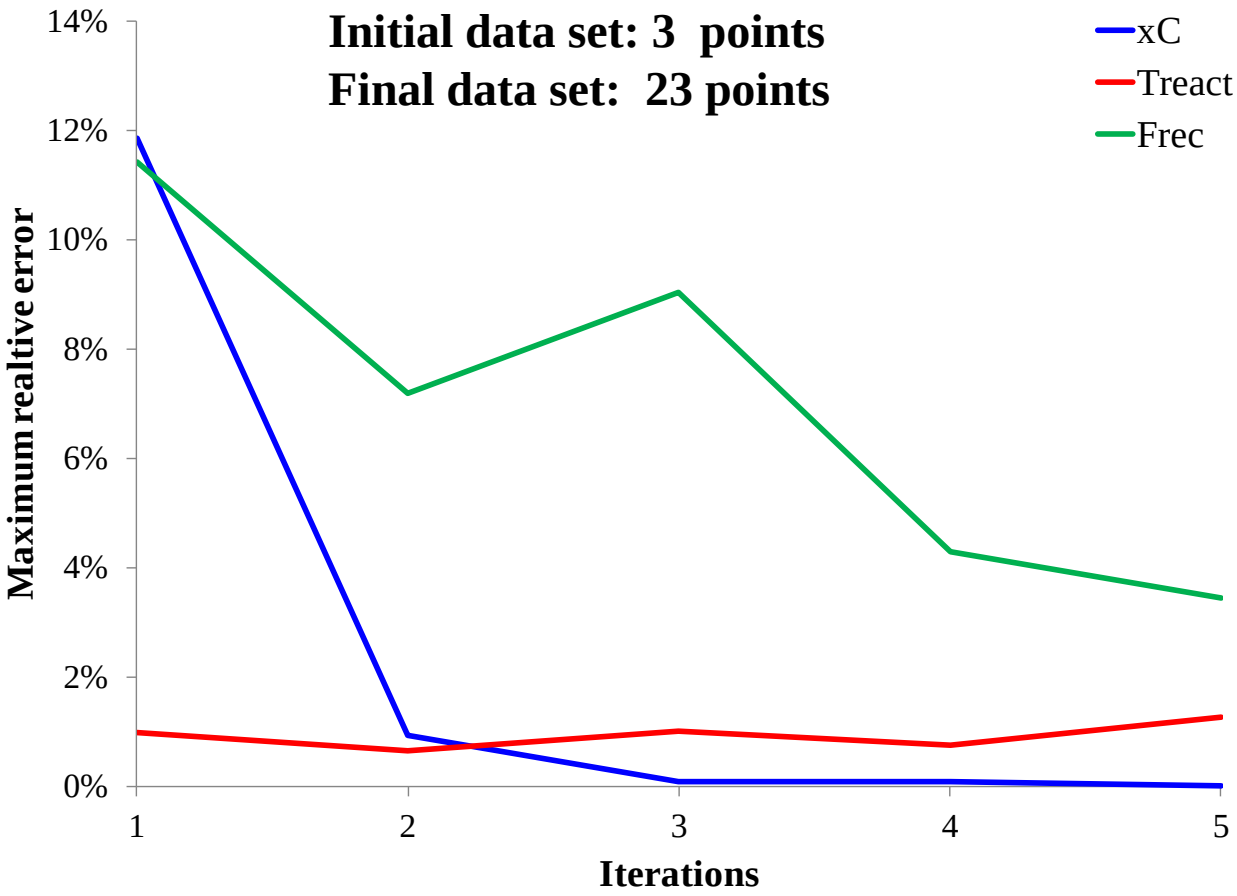
Generate Models:

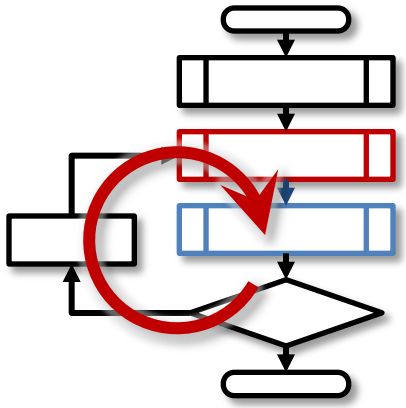
$$x^{\mathrm{C}}(T^{\mathrm{sp}}, P) = f_1(T^{\mathrm{sp}}, P)$$
$$T^{\mathrm{react}}(T^{\mathrm{sp}}, P) = f_2(T^{\mathrm{sp}}, P)$$
$$F^{\mathrm{rec}}(T^{\mathrm{sp}}, P) = f_3(T^{\mathrm{sp}}, P)$$

Over the Range:

$$100^\circ F \leq T^{\mathrm{sp}} \leq 250^\circ F$$
$$0.82 \text{ atm} \leq P \leq 1.36 \text{ atm}$$

GENERATING THE SURROGATES





- **Maximum error found at each iteration may increase**
  - **Due to the derivative-free solver is given more information at each iteration**

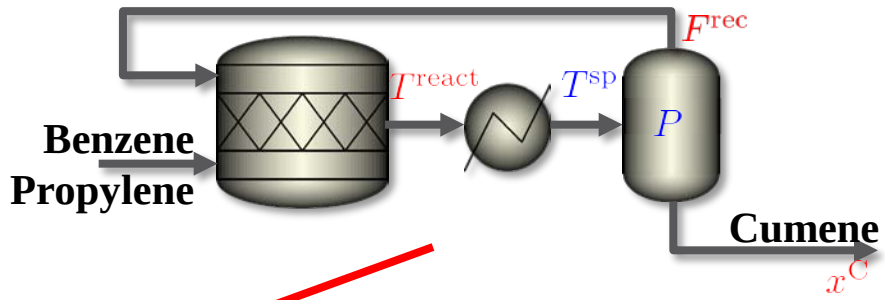
FINAL MODELS

$$x^{\mathrm{C}}(T^{\mathrm{sp}}, P) = 0.00156\, T^{\mathrm{sp}} - \left(5.08 \cdot 10^{-6}\right)\, \left(T^{\mathrm{sp}}\right)^2 + \left(7.12 \cdot 10^{-9}\right)\, \left(T^{\mathrm{sp}}\right)^3 + \frac{3.08}{P\, T^{\mathrm{sp}}} + \frac{\left(4.15 \cdot 10^{-10}\right) \left(T^{\mathrm{sp}}\right)^3}{P^3} - \frac{12.2\, \sqrt{P}}{\sqrt{T^{\mathrm{sp}}}} + \frac{7.06\, \sqrt[3]{P}}{\sqrt[3]{T^{\mathrm{sp}}}} + 0.427$$

$$T^{\mathrm{react}}(T^{\mathrm{sp}}, P) = \frac{77.1}{\sqrt[3]{P}} + 780.0$$

$$F^{\mathrm{rec}}(T^{\mathrm{sp}}, P) = \left(5.37 \cdot 10^4\right)\, \sqrt{0.01\, T^{\mathrm{sp}}}\, \sqrt{1.22\, P} - 163.0\, T^{\mathrm{sp}} - \left(6.26 \cdot 10^4\right)\, \left(0.01\, T^{\mathrm{sp}}\right)^{\frac{1}{3}} \left(1.22\, P\right)^{\frac{1}{3}} + \frac{206.0\, T^{\mathrm{sp}}}{P} + \frac{9377.0}{\sqrt{0.01\, T^{\mathrm{sp}}}} - \frac{2055.0}{P^3} + \left(2.76 \cdot 10^{-4}\right)\, \left(T^{\mathrm{sp}}\right)^3 - \frac{0.452\, \left(T^{\mathrm{sp}}\right)^2}{P^2} - 0.022\, P^2\, \left(T^{\mathrm{sp}}\right)^2 + \frac{\left(5.44 \cdot 10^{-4}\right) \left(T^{\mathrm{sp}}\right)^3}{P^3} - \left(1.28 \cdot 10^{-5}\right)\, P^3\, \left(T^{\mathrm{sp}}\right)^3$$

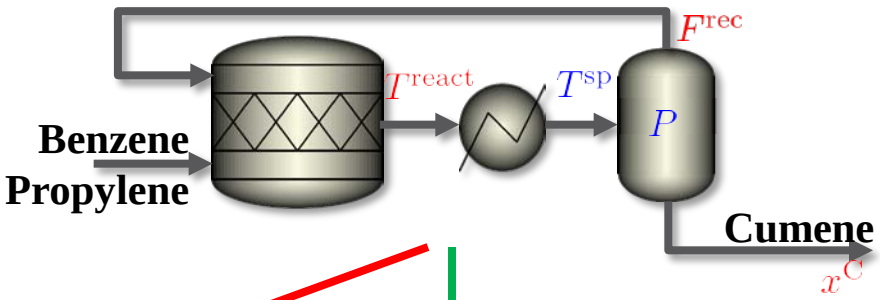
PROCESS OPTIMIZATION



$$\max \quad x^{\mathrm{C}}$$

$$\underbrace{\begin{aligned} T^{\mathrm{sp}} &= 250^\circ\mathrm{F} \\ P &= 0.82\mathrm{atm} \\ x^{\mathrm{C}} &= 0.981 \\ T^{\mathrm{react}} &= 828.1^\circ\mathrm{F} \\ F^{\mathrm{rec}} &= 645.7\frac{\mathrm{lb}}{\mathrm{hr}} \\ \dot{\mathrm{m}}^{\mathrm{cw}} &= 5.00 \cdot 10^7 \frac{\mathrm{lb}}{\mathrm{hr}} \\ \mathrm{A} &= 7,706\mathrm{ft}^2 \end{aligned}}_{\leq 5.2\%\mathrm{error}}$$

PROCESS OPTIMIZATION



max x^{C}

$$
\begin{aligned}
T^{\mathrm{sp}} &= 250^\circ\mathrm{F} \\
P &= 0.82\mathrm{atm} \\
x^{\mathrm{C}} &= 0.981 \\
T^{\mathrm{react}} &= 828.1^\circ\mathrm{F} \\
F^{\mathrm{rec}} &= 645.7\frac{\mathrm{lb}}{\mathrm{hr}} \\
\dot{\mathrm{m}}^{\mathrm{cw}} &= 5.00 \cdot 10^7 \frac{\mathrm{lb}}{\mathrm{hr}} \\
\mathrm{A} &= 7,706\mathrm{ft}^2
\end{aligned}
$$

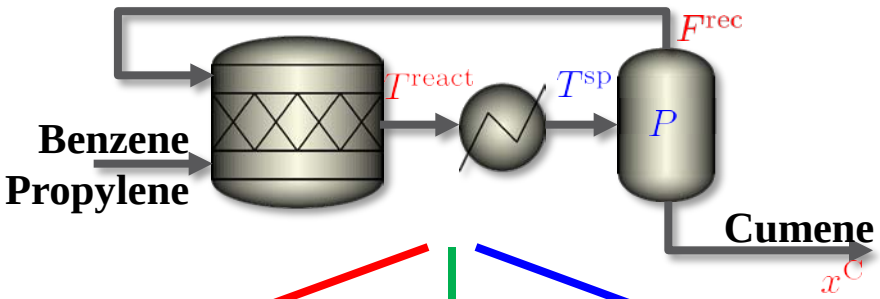
$\leq 5.2\%\mathrm{error}$

min Utilities

$$
\begin{aligned}
T^{\mathrm{sp}} &= 250^\circ\mathrm{F} \\
P &= 1.36\mathrm{atm} \\
x^{\mathrm{C}} &= 0.964 \\
T^{\mathrm{react}} &= 848.9^\circ\mathrm{F} \\
F^{\mathrm{rec}} &= 373.16\frac{\mathrm{lb}}{\mathrm{hr}} \\
\dot{\mathrm{m}}^{\mathrm{cw}} &= 4.04 \cdot 10^4 \frac{\mathrm{lb}}{\mathrm{hr}} \\
\mathrm{A} &= 2,364\mathrm{ft}^2
\end{aligned}
$$

$\leq 3.0\%\mathrm{error}$

PROCESS OPTIMIZATION



max x^{C}

$$T^{\mathrm{sp}} = 250°\mathrm{F}$$
$$P = 0.82\mathrm{atm}$$
$$x^{\mathrm{C}} = 0.981$$
$$T^{\mathrm{react}} = 828.1°\mathrm{F}$$
$$F^{\mathrm{rec}} = 645.7\frac{\mathrm{lb}}{\mathrm{hr}}$$
$$\dot{\mathrm{m}}^{\mathrm{cw}} = 5.00 \cdot 10^{7}\frac{\mathrm{lb}}{\mathrm{hr}}$$
$$\mathrm{A} = 7,706\mathrm{ft}^2$$

$\leq 5.2\%$error

min Utilities

$$T^{\mathrm{sp}} = 250°\mathrm{F}$$
$$P = 1.36\mathrm{atm}$$
$$x^{\mathrm{C}} = 0.964$$
$$T^{\mathrm{react}} = 848.9°\mathrm{F}$$
$$F^{\mathrm{rec}} = 373.16\frac{\mathrm{lb}}{\mathrm{hr}}$$
$$\dot{\mathrm{m}}^{\mathrm{cw}} = 4.04 \cdot 10^{4}\frac{\mathrm{lb}}{\mathrm{hr}}$$
$$\mathrm{A} = 2,364\mathrm{ft}^2$$

$\leq 3.0\%$error

min CaptialCost

s.t. $x^{\mathrm{C}} \geq 0.95$

$$T^{\mathrm{sp}} = 168.1°\mathrm{F}$$
$$P = 0.82\mathrm{atm}$$
$$x^{\mathrm{C}} = 0.948$$
$$T^{\mathrm{react}} = 860.4°\mathrm{F}$$
$$F^{\mathrm{rec}} = 172.7\frac{\mathrm{lb}}{\mathrm{hr}}$$
$$\dot{\mathrm{m}}^{\mathrm{cw}} = 4.40 \cdot 10^{4}\frac{\mathrm{lb}}{\mathrm{hr}}$$
$$\mathrm{A} = 2,310\mathrm{ft}^2$$

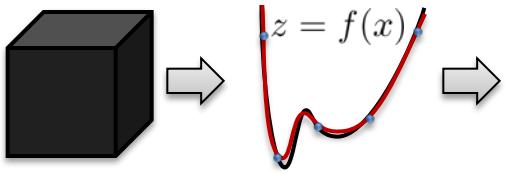
$\leq 5.7\%$error

CONCLUSIONS

- **The algorithm we developed is able to model black-box functions for use in optimization such that the models are**
  - ✓ **Accurate**
  - ✓ **Tractable in an optimization framework (low-complexity models)**
  - ✓ **Generated from a minimal number of function evaluations**
- **Surrogate models can then be incorporated within a optimization framework flexible objective functions and additional constraints**

ALAMO

Automated **L**earning of **A**lgebraic **M**odels for **O**ptimization



$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0 \end{aligned}$$