petBOA – A Python tool for Parameter Estimation of “black box” Macro- and Micro-kinetic Models with Bayesian Optimization

Sashank Kasiraju, **Yifan Wang**, **Dionisios Vlachos***
University of Delaware
Saurabh Bhandari, **Aayush Singh**
The Dow Chemical Company

https://github.com/VlachosGroup/petboa
Paper Submitted

VlachosLab
AT THE UNIVERSITY OF DELAWARE

Fine-tuning MKMs

- Microkinetic models are adjusted to ensure better parity with experimental observations
- The adjustments are done to the DFT potential energies of reactant and product intermediates and transition state species (within the DFT error ~20-30 kJ/mol)
- MKM parameters are updated by defining an optimization problem which reduces the disparity between model and experimental quantities of interest.

Literature Example 1: Methanol Synthesis

- Methanol Synthesis on Cu through CO_2 and CO Hydrogenation
- 49 Elementary Steps Included in the Microkinetic Model

Table 2. The 49 Elementary Steps Included in the Microkinetic Model[a]

no.	reaction	E_{int}/eV IS/FS	ΔE/eV	E_a / eV	sticking probability or pre-exponential factor	ω
R1	$CO_2 + * \rightarrow CO_2*$		−0.08		1	
R2	$H_2 + 2* \rightarrow 2\ H*$		−0.29		1	
R3	$CO + * \rightarrow CO*$		−0.86		1	
R4	$H_2O + * \rightarrow H_2O*$		−0.21		1	
R5	$HCOOH + * \rightarrow HCOOH*$		−0.22		1	
R6	$CH_2O + * \rightarrow CH_2O*$		−0.04		1	
R7	$CH_3OH + * \rightarrow CH_3OH*$		−0.28		1	
R8	$HCOOCH_3 + * \rightarrow HCOOCH_3*$		−0.10		1	
R9	$CO* + O* \rightarrow CO_2* + *$	0.14/0.76	−1.12	0.65	1.195×10^{12}	0.44
R10	$CO* + OH* \rightarrow COOH* + *$	0/0	0.14	0.56	4.667×10^{11}	0.36
R11	$COOH* + * \rightarrow CO_2* + H*$	0.00/0.11	−0.55	1.23	2.326×10^{13}	0.32
R12[b]	$COOH* + OH* \rightarrow CO_2* + H_2O*$	−0.34/0.22	−0.76	0	1.000×10^{13}	0.00
R13	$COOH* + H* \rightarrow HCOOH* + *$	−0.01/0.00	−0.59	0.73	6.793×10^{13}	0.35
R14	$H_2O* + * \rightarrow OH* + H*$	0.01/0.08	0.21	1.39	1.436×10^{11}	0.60

$$E_f = E_{f,DFT} + \omega(\Delta H - \Delta E_{DFT})$$

Table 5. Fully Optimized Parameter Set Used in the Microkinetic Model[a]

parameter	optimized value
ΔH_{fit} (H)	−0.07 eV
ΔH_{fit} (O)	0.00 eV
ΔH_{fit} (OH)	−0.41 eV
ΔH_{fit} (H_2O)	0.00 eV
ΔH_{fit} (CO)	0.10 eV
ΔH_{fit} (CO_2)	0.00 eV
ΔH_{fit} (COOH)	−0.31 eV
ΔH_{fit} (HCO)	−0.55 eV
ΔH_{fit} (HCOO)	−0.60 eV
ΔH_{fit} (H_2COO)	0.00 eV
ΔH_{fit} (HCOOH)	−0. 49 eV
ΔH_{fit} (CH_3O_2)	−0.55 eV
ΔH_{fit} (CH_2O)	−0.43 eV
ΔH_{fit} (CH_3O)	−0.64 eV
ΔH_{fit} (CH_2OH)	0.00 eV
ΔH_{fit} (CH_3OH)	−0.23 eV
$\omega(CO* + H* \rightarrow HCO* + *)$	0.68
$\omega(HCO* + H* \rightarrow CH_2O* + *)$	0.26
$\omega(CH_2O* + H* \rightarrow CH_3O* + *)$	0.21
$\omega(CO_2* + H* \rightarrow HCOO* + *)$	0.82
$\omega(HCOO* + H* \rightarrow HCOOH* + *)$	0.68
$\omega(HCOO* + H* \rightarrow H_2CO_2* + *)$	0.33
$\omega(HCOOH* + H* \rightarrow CH_3O_2*)$	0.30
$\omega(CO* + O* \rightarrow CO_2* + *)$	0.23
$\omega(CO* + OH* \rightarrow COOH* + *)$	0.59
$\omega(COOH* + * \rightarrow CO_2* + H*)$	0.33
$\omega(H_2O* + * \rightarrow OH* + H*)$	0.87
$\omega(OH* + * \rightarrow O* + H*)$	0.91
$\omega(2\ OH* + * \rightarrow H_2O* + O*)$	0.66

L. C. Grabow and M. Mavrikakis, Mechanism of Methanol Synthesis on Cu through CO_2 and CO Hydrogenation**,** *ACS Catalysis*, 365-384, **2011.**

Literature Example 1: Methanol Synthesis

- Methanol Synthesis on Cu through CO_2 and CO Hydrogenation
- 49 Elementary Steps Included in the Microkinetic Model

Table 2. The 49 Elementary Steps Included in the Microkinetic Model[a]

no.	reaction	E_{int}/eV IS/FS	ΔE/eV	E_a / eV	sticking probability or pre-exponential factor	ω
R1	$CO_2 + * \rightarrow CO_2*$		−0.08		1	
R2	$H_2 + 2* \rightarrow 2\ H*$		−0.29		1	
R3	$CO + * \rightarrow CO*$		−0.86		1	
R4	$H_2O + * \rightarrow H_2O*$		−0.21		1	
R5	$HCOOH + * \rightarrow HCOOH*$		−0.22		1	
R6	$CH_2O + * \rightarrow CH_2O*$		−0.04		1	
R7	$CH_3OH + * \rightarrow CH_3OH*$		−0.28		1	
R8	$HCOOCH_3 + * \rightarrow HCOOCH_3*$		−0.10		1	
R9	$CO* + O* \rightarrow CO_2* + *$	0.14/0.76	−1.12	0.65	1.195×10^{12}	0.44
R10	$CO* + OH* \rightarrow COOH* + *$	0/0	0.14	0.56	4.667×10^{11}	0.36
R11	$COOH* + * \rightarrow CO_2* + H*$	0.00/0.11	−0.55	1.23	2.326×10^{13}	0.32
R12[b]	$COOH* + OH* \rightarrow CO_2* + H_2O*$	−0.34/0.22	−0.76	0	1.000×10^{13}	0.00
R13	$COOH* + H* \rightarrow HCOOH* + *$	−0.01/0.00	−0.59	0.73	6.793×10^{13}	0.35
R14	$H_2O* + * \rightarrow OH* + H*$	0.01/0.08	0.21	1.39	1.436×10^{11}	0.60

$$E_f = E_{f,DFT} + \omega(\Delta H - \Delta E_{DFT})$$

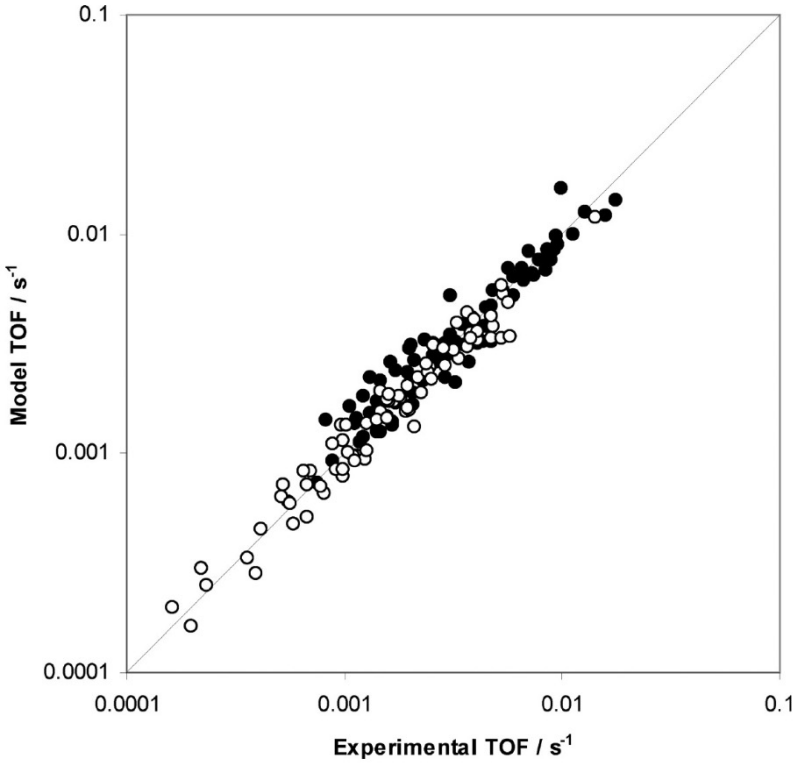

Figure 4. Parity plot of experimental and calculated TOF. Full circles represent CH_3OH production rates; open circles, H_2O production rates. Experimental data is taken from refs 27 and 28. Quality of fit: R^2-(CH_3OH) = 0.90, R^2 (H_2O) = 0.94, R^2 (overall) = 0.92.

L. C. Grabow and M. Mavrikakis, Mechanism of Methanol Synthesis on Cu through CO_2 and CO Hydrogenation**,** *ACS Catalysis*, 365-384, **2011.**

Literature Example 2: Sequential Optimization

- A sequential optimization algorithm for parameter estimation of MKMs

Objective Function

$$Z_{\text{Param.Est}} = \min \sum_{c \in C} (y_c - \overline{y}_c)^T W (y_c - \overline{y}_c) + \delta(\pi - \pi_{\text{nom}})^T (\pi - \pi_{\text{nom}})$$

$$\Delta H_j = \sum_{i \in I_j} \nu_{ij} H_i \qquad \forall j \in J$$

$$H_i = H_i^{\text{DFT}} + \Delta \text{BE}_i \qquad \forall i \in I$$

Binding Energy Deviation

$$E_{a,j} = E_{a,j}^{\text{DFT}} + \Delta E_{a,j}$$

$$H_{\text{TS},j} = H_{\text{TS},j}^{\text{DFT}} + \Delta E_{\text{TS},j} \qquad \forall j \in J$$

Transition State Deviation

$$\frac{dx_c}{dt} = f(x_c, k_c, u_c)$$

$$g(x_c, k_c, u_c) = 0$$

$$y_c = Bx_c(t \to \infty); \qquad x_c(0) = x_0(u_c);$$

$$u_c = \{F_{\text{in},c}, P_{\text{tot},c}, T_c\}$$

$$h(y_c, u_c, k_c) \geq 0$$

$$k_c = \psi(p, \pi_c, u_c)$$

MKM

$$k_{\text{for},j} = k_{0j} e^{-E_{a,j}/RT} \qquad \forall j \in J$$

$$k_{\text{for},j} = K_j k_{\text{rev},j} \qquad \forall j \in J$$

$$K_j = e^{-\Delta H_j/RT} e^{\Delta S_j/R} = K_{0j} e^{-\Delta H_j/RT} \qquad \forall j \in J$$

Rate/Equilibrium constants

$$E_{a,j} \geq 0 \qquad \forall j \in J$$

$$E_{a,j} \geq \Delta H_j \qquad \forall j \in J$$

$$\pi_{\text{LB},n} \leq \pi_n \leq \pi_{\text{UB},n} \qquad \forall n \in \{1, 2, \ldots, N\}$$

Thermodynamic Constraints

Srinivas Rangarajan et. al., Sequential-Optimization-Based Framework for Robust Modeling and Design of Heterogeneous Catalytic Systems, JPCC 121 (46), 25847-25863, **2017**

Literature Example 2: Sequential Optimization

- A sequential optimization algorithm for parameter estimation of MKMs

Objective Function

$$Z_{\text{Param.Est}} = \min \sum_{c \in C} (y_c - \overline{y}_c)^T W (y_c - \overline{y}_c) + \delta(\pi - \pi_{\text{nom}})^T (\pi - \pi_{\text{nom}})$$

$$\frac{dx_c}{dt} = f(x_c, k_c, u_c)$$

$$g(x_c, k_c, u_c) = 0$$

$$y_c = Bx_c(t \to \infty); \quad x_c(0) = x_0(u_c);$$

$$u_c = \{F_{\text{in},c}, P_{\text{tot},c}, T_c\}$$

$$h(y_c, u_c, k_c) \geq 0$$

$$k_c = \psi(p, \pi_c, u_c)$$

MKM

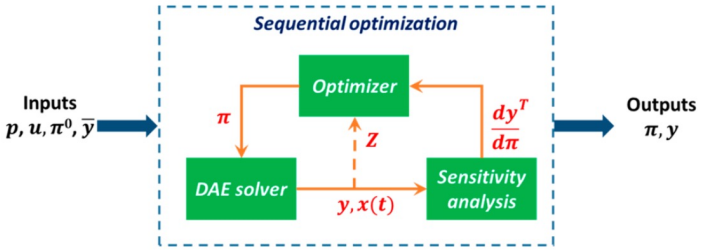

Figure 1. Schematic of the information flow in the sequential method for solving nonlinear optimization problems with an embedded system of DAEs. The flow of decision variables π, objective Z, state profiles y, $x(t)$, and sensitivities $\left(\frac{dy}{d\pi}\right)^T$ have been marked explicitly. Inputs to the optimization block are constants p such as the DFT-derived kinetics and thermochemistry, initial guesses of decision variables π^0, reactor inputs u such as the temperature, pressure, and feed composition, and experimental data $\overline{y}$, while the outputs are the optimal values of the decision variables π and model predictions y.

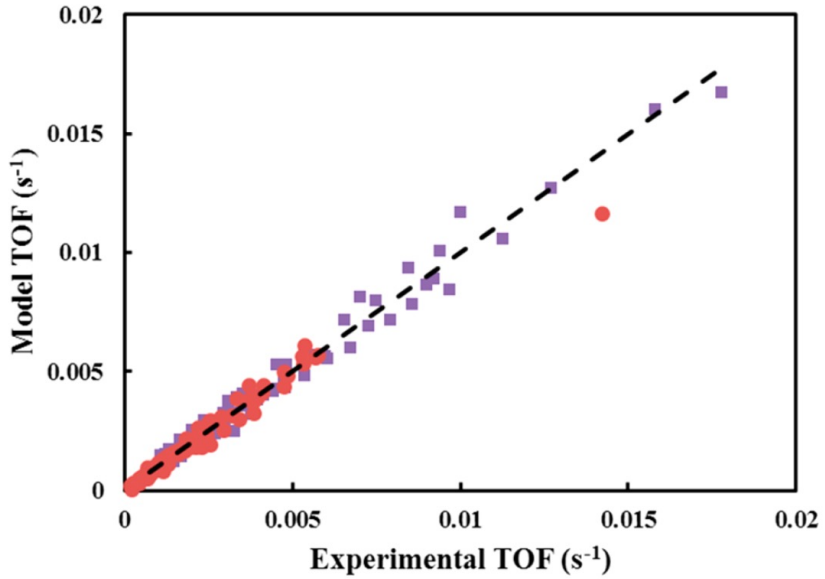

Figure 3. Model versus experiment turnover frequency (TOF) parity plot of the primary solution for the parameter estimation problem for methanol synthesis. Purple squares (■) indicate methanol TOF, red circles (●) indicate water TOF, and the dashed line is the parity line.

Srinivas Rangarajan et. al., Sequential-Optimization-Based Framework for Robust Modeling and Design of Heterogeneous Catalytic Systems, JPCC 121 (46), 25847-25863, **2017**

petBOA

- petBOA uses Bayesian Optimization for a gradient-free parameter estimation
- petBOA relies on NEXTorch[1] for the Bayesian Optimization routines
- Parameter updates for MKMs i.e., (adsorbate reactant/product/transition state deviations) are handled “on-the-fly” leveraging pMuTT and OpenMKM
- Code and detailed example templates can be found at
  - GitHub: https://github.com/vlachosGroup/petBOA

Folder Tree for the main project

```
.
├── petboa                  # Source files: classes and methods for BO based parameter estimation
├── examples                # All example templates with respective input data and results
├── LICENSE.md
├── setup.py
└── README.md
```

Examples folder tree

This describes the folder tree for the templates.

```
.
.
├── example_1_rosenbrock                    # Generic PE template for a parametrized model
│   ├── evaluate_GSA                        # Global sensitivity analysis using SALib to identify
│   ├── petboa                              # Fit parameters using petBOA
│   └── scipy                               # Fit parameters using SciPy Nelder-Mead
├── example_2_python_EDH_model              # Python Macrokinetic model: Batch Reactor for the et
│   ├── evaluate_gsa                        # Global sensitivity analysis using SALib to identify
│   ├── gen_data                            # Generate simulated data used for fitting the macrok
│   ├── petboa_all_params                   # Fit parameters using petBOA
│   ├── petboa_noisy_data                   # Fit parameters with noisy simulated data using petB
│   └── scipy_all_params                    # Fit parameters using SciPy
```

[1] https://github.com/vlachosGroup/nextorch

[1] Y. Wang, T.-Y. Chen, and D.G. Vlachos, NEXTorch: A Design and Bayesian Optimization Toolkit for Chemical Sciences and Engineering, J. Chem. Inf. Model. 2021, 61, 11, 5312–5319

petBOA

- petBOA uses Bayesian Optimization for a gradient-free parameter estimation
- petBOA relies on NEXTorch[1] for the Bayesian Optimization routines
- Parameter updates for MKMs i.e., (adsorbate reactant/product/transition state deviations) are handled “on-the-fly” leveraging pMuTT and OpenMKM
- Code and detailed example templates can be found at
  - GitHub: https://github.com/vlachosGroup/petBOA

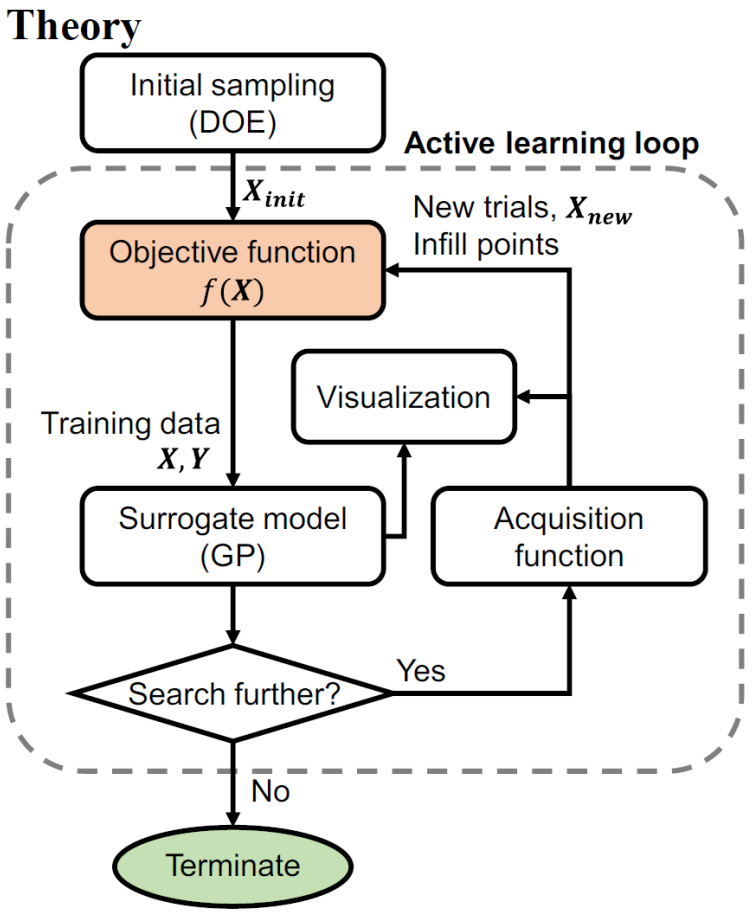

Figure 1. Overview of the active learning framework.

[1] https://github.com/vlachosGroup/nextorch

[1] Y. Wang, T.-Y. Chen, and D.G. Vlachos, NEXTorch: A Design and Bayesian Optimization Toolkit for Chemical Sciences and Engineering, J. Chem. Inf. Model. 2021, 61, 11, 5312–5319

petBOA with OpenMKM

- MKM's solved using OpenMKM
- Parameter tuning using NEXTorch's Bayesian Optimization
- All input files are generated using pMuTT during optimization
- Species binding energies are updated by changing the coefficients (a_5, F) of
- Nasa Polynomials
  - $\frac{H}{RT} = a_0 + a_1\frac{T}{2} + a_2\frac{T^2}{3} + a_3\frac{T^3}{4} + a_4\frac{T^4}{5} + a_5\frac{1}{T}$
- Shomate Polynomials
  - $\frac{H}{RT} = \frac{1}{RT}\left(At + B\frac{t^2}{2} + C\frac{t^3}{3} + D\frac{t^4}{4} - E\frac{1}{t} + F\right)$

- When making OpenMKM input files with using updated NASA/Shomate polynomials:
  - pMuTT updates thermo (yaml/xml) for new thermochemistry
  - pMuTT recalculated activation energy (E_a) of all elementary steps in which the species participates
  - pMuTT ensure newer activation energy is thermodynamically consistent

petBOA Workflow

- Use pMuTT-Excel link to develop OpenMKM input files.
- Decide on what parameters to fit/tune in this exercise i.e., deviations to ground-state species, transition state energies, or explicit activation barriers.
  - Perform sensitivity analysis to identify key params!
- Using Python Programming and pMuTT
  - Editing “reactor.yaml” input file based on experimental data to change operating conditions: P, T, Q, V, initial X, etc.
  - Thermodynamic specification file with new perturbations
  - Edit the “loss function” routine to decide on the logic used for evaluation the loss of the objective function.
- Parts of this workflow are automated, but some tweaks must be done manually.
- Run the parameter estimation and repeat it by “bracketing” the bounds, or multi-starting it identify optimal solution!

```
...
def loss_function(params, **kwargs):
        create_thermo_yaml(params, *args)
        create_reactor_yaml(params, *args)
        qois = model_wrapper.run(OpenMKM)
        loss = calculate(qois, input_data)
        return loss

def create_thermo_yaml(params):
        parse_params(params)
        new_thermo = perturb_species_thermo(params, old_species_thermo)
        kin_params = recalculate_kinetic_params(new_thermo)
        write_thermo_file(new_thermo, kin_params)

def main:
        # Transform experimental data
        ...
        # Read Parameter Data
        param_data = read_param_input_file()
        # Populate Model Wrapper with data
        ...
        # Run Bayesian Optimization
        result = run_BO_optimizer(wrapper, params, loss,*,**)
        # plot and save results
        ...
...
```

Demo 1: petBOA – NH_3 Model

See Example at:

VLAB-Workshop/petBOA/

VlachosLab
AT THE UNIVERSITY OF DELAWARE

Acknowledgements

1. Department of Energy
2. RAPID (AIChE)
3. State of Delaware

Dr. Jeffrey Frey, HPC
Kelly Walker – Logos
Jaynell Keely - Logos