

Exercises Numerical Methods 2023/2024

::: {.callout-important}

Important:

- Results from these exercises should be added to this document, i.e., you should submit an updated version of this notebook.
- Only one document, that was tested on our [KOARCH Server](#), is allowed.
- Presentation is based on this notebook. No additional slides are allowed!
- Python version 3.11 and spotPython version [0.10.15](#) (or higher) is required, see, e.g., <https://pypi.org/project/spotPython/> or on [GitHub](#).
- Please incorporate the results from the exercises as well as your own, additional findings, into this notebook and submit the updated notebook.
- Your own ideas and findings are welcome and will be rewarded.

:::

Exercises Chapter 08: Multi-dimensional Functions

- The following exercises refer to chapter 08 of the book, see https://sequential-parameter-optimization.github.io/Hyperparameter-Tuning-Cookbook/008_num_spot_multidim.html.

1. The Three Dimensional fun_cubed

- The input dimension is [3](#). The search range is $-1 \leq x \leq 1$ for all dimensions.
- Generate contour plots
- Calculate the variable importance.
- Discuss the variable importance:
 - Are all variables equally important?
 - If not:
 - Which is the most important variable?
 - Which is the least important variable?

Definition The `fun_cubed` function represents a mathematical model that calculates the cube of its input. This type of function is often used in mathematical modeling and optimization problems to introduce non-linearity.

Mathematical Representation The `fun_cubed` function can be expressed mathematically as:

$$f(x) = x^3$$

where:

x is the input variable. x^3 is the cube of x . Characteristics: Univariate: This function takes a single input variable. Non-linear: The relationship between the input and output is non-linear due to the cubic term. Smooth and Continuous: The function is smooth and continuous over all real numbers. Symmetric: The function is symmetric with respect to the origin, meaning $f(-x) = -f(x)$. Monotonic: The function is strictly increasing for positive values of x

x and strictly decreasing for negative values of x

```
In [1]: import numpy as np
from spotPython.fun.objectivefunctions import analytical
from spotPython.utils.init import fun_control_init, surrogate_control_init
from spotPython.spot import spot
import matplotlib.pyplot as plt

# Define the function
fun = analytical().fun_cubed
# Define variables as input to the function
fun_control = fun_control_init(
    PREFIX="03",
    lower = -1.0*np.ones(3),
    upper = 1.0*np.ones(3),
    var_name=["V1", "V2", "V3"],
    show_progress=True)
#Define a surrogate
surrogate_control = surrogate_control_init(n_theta=3)
#Give input to SPOT
spot_3 = spot.Spot(fun=fun,
                    fun_control=fun_control,
                    surrogate_control=surrogate_control)
#Run Spot
spot_3.run()
_ = spot_3.print_results()
spot_3.plot_contour(i=0, j=1)
spot_3.plot_contour(i=0, j=2)
spot_3.plot_contour(i=1, j=2)

_ = spot_3.print_importance()
spot_3.plot_importance()

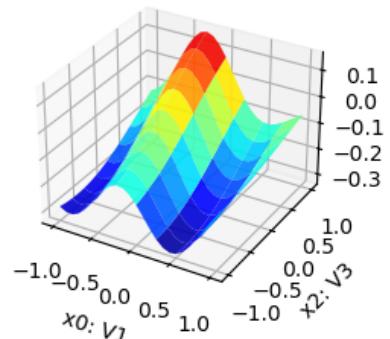
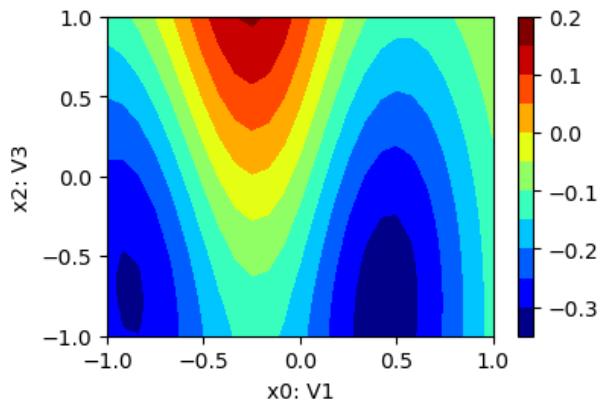
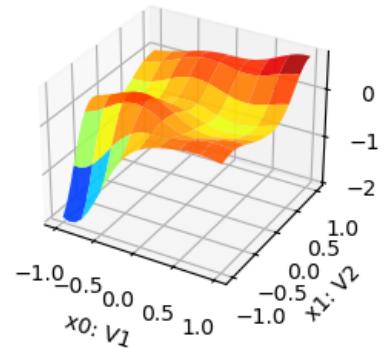
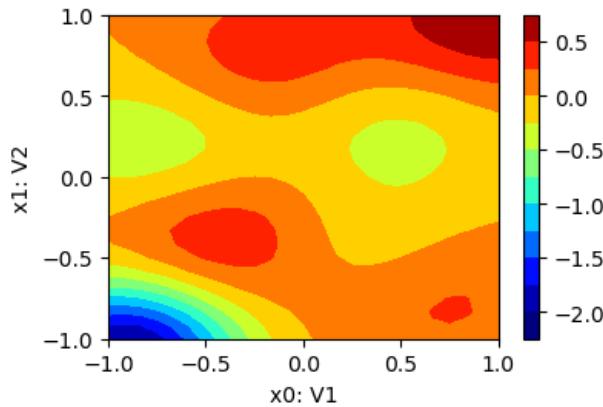
[rank: 0] Seed set to 123
[rank: 0] Seed set to 123
```

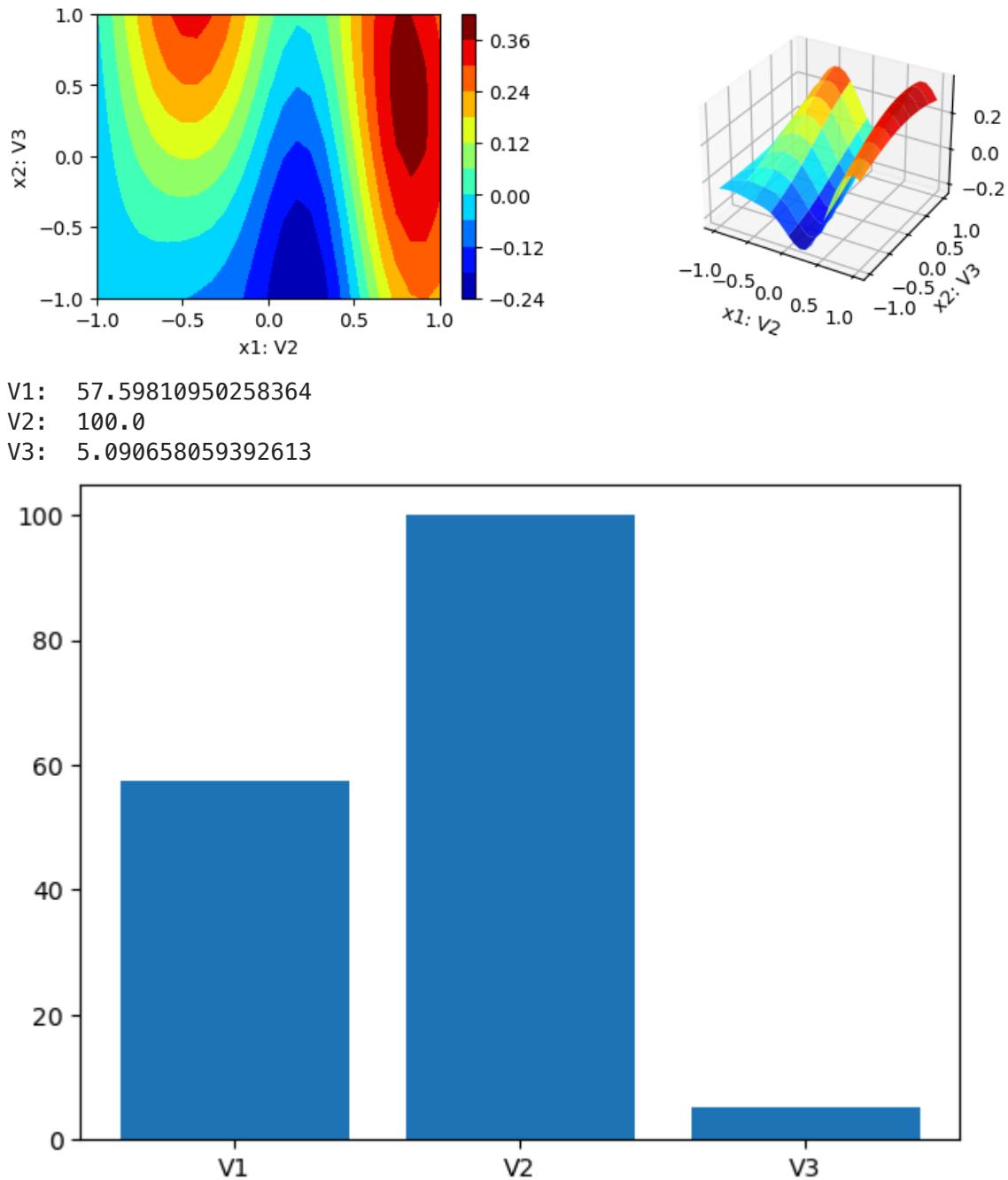
```
Created spot_tensorboard_path: runs/spot_logs/03_c02-b01_2024-05-19_22-5  
2-41 for SummaryWriter()  
spotPython tuning: -1.4616821761042098 [#####---] 73.33%  
spotPython tuning: -1.4616821761042098 [#####---] 80.00%  
spotPython tuning: -2.0535960676161324 [#####---] 86.67%  
spotPython tuning: -2.0535960676161324 [#####---] 93.33%  
spotPython tuning: -2.0535960676161324 [#####---] 100.00% Done...  
  
{'CHECKPOINT_PATH': 'runs/saved_models/',  
 'DATASET_PATH': 'data/',  
 'PREFIX': '03',  
 'RESULTS_PATH': 'results/',  
 'TENSORBOARD_PATH': 'runs/',  
 '_L_in': None,  
 '_L_out': None,  
 '_torchmetric': None,  
 'accelerator': 'auto',  
 'converters': None,  
 'core_model': None,  
 'core_model_name': None,  
 'counter': 15,  
 'data': None,  
 'data_dir': './data',  
 'data_module': None,  
 'data_set': None,  
 'data_set_name': None,  
 'db_dict_name': None,  
 'design': None,  
 'device': None,  
 'devices': 1,  
 'enable_progress_bar': False,  
 'eval': None,  
 'fun_evals': 15,  
 'fun_repeats': 1,  
 'horizon': None,  
 'infill_criterion': 'y',  
 'k_folds': 3,  
 'log_graph': False,  
 'log_level': 50,  
 'loss_function': None,  
 'lower': array([-1., -1., -1.]),  
 'max_surrogate_points': 30,  
 'max_time': 1,  
 'metric_params': {},  
 'metric_river': None,  
 'metric_sklearn': None,  
 'metric_sklearn_name': None,  
 'metric_torch': None,  
 'model_dict': {},  
 'n_points': 1,  
 'n_samples': None,  
 'n_total': None,  
 'noise': False,  
 'num_workers': 0,  
 'ocba_delta': 0,  
 'oml_grace_period': None,  
 'optimizer': None,  
 'path': None,  
 'prep_model': None,  
 'prep_model_name': None},
```

```

'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/03_c02-b01_2024-05-19_22-52-4
1',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x148c57ca3310>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([1., 1., 1.]),
'var_name': ['V1', 'V2', 'V3'],
'var_type': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: -2.0535960676161324
V1: -1.0
V2: -1.0
V3: -0.37703150540115343

```





2. The Ten Dimensional fun_wing_wt

- The input dimension is **10**. The search range is $0 \leq x \leq 1$ for all dimensions.
- Calculate the variable importance.
- Discuss the variable importance:
 - Are all variables equally important?
 - If not:
 - Which is the most important variable?
 - Which is the least important variable?
 - Generate contour plots for the three most important variables. Do they confirm your selection?

Definition The `fun_wing_wt` function represents a mathematical model used to estimate the weight of an aircraft wing. This type of function is often used in aerospace engineering for design and optimization purposes.

Mathematical Representation The wing weight function typically has multiple input variables that correspond to different design parameters of the wing. A commonly used model is:

$$W_{\text{wing}} = 0.036 \times S_w \cdot 0.758 \times W_{\text{fw}} \cdot 0.0035 \times (A \cos 2(\Lambda)) \cdot 0.6 \times q \cdot 0.006 \times \lambda \cdot 0.04 \\ \times (100 \times t/c \cos (\Lambda)) - 0.3 \times (N_z \times W_{\text{dg}}) \cdot 0.49$$

•

$$S_w \times W_p \cdot W$$

where:

$$S_w$$

: Wing area

W_{fw} : Weight of fuel in the wing

A : Aspect ratio

Λ : Quarter-chord sweep angle (in degrees)

q : Dynamic pressure at cruise λ : Taper ratio t/c : Thickness-to-chord ratio N_z :

Ultimate load factor W_{dg} : Flight design gross weight W_p

: Paint weight per unit area Characteristics Multivariable: This function has multiple input parameters, each representing a physical characteristic or design parameter of the wing. Non-linear: The relationship between the inputs and the wing weight is non-linear, as evidenced by the exponents and trigonometric functions. Dimensional Analysis: The function involves dimensional quantities like area, weight, pressure, and angle. Optimization: It is often used in optimization problems to minimize or constrain the weight of the wing while satisfying design requirements.

```
In [1]: import numpy as np
from spotPython.fun.objectivefunctions import analytical
from spotPython.utils.init import fun_control_init, surrogate_control_init
from spotPython.spot import spot
import matplotlib.pyplot as plt

# Define the function
fun = analytical().fun_wingwt
# Define variables as input to the function
fun_control = fun_control_init(
    PREFIX="10",
    lower = 0*np.ones(10),
    upper = 1.0*np.ones(10),
    var_name=["V1", "V2", "V3", "V4", "V5", "V6", "V7", "V8", "show_progress=True")
#Define a surrogate
```

```
surrogate_control = surrogate_control_init(n_theta=10)
#Give input to SPOT
spot_10 = spot.Spot(fun=fun,
                     fun_control=fun_control,
                     surrogate_control=surrogate_control)

#Run Spot
spot_10.run()
_ = spot_10.print_results()
spot_10.plot_contour(i=7, j=8)
spot_10.plot_contour(i=8, j=2)
spot_10.plot_contour(i=7, j=2)

_ = spot_10.print_importance()
spot_10.plot_importance()
```

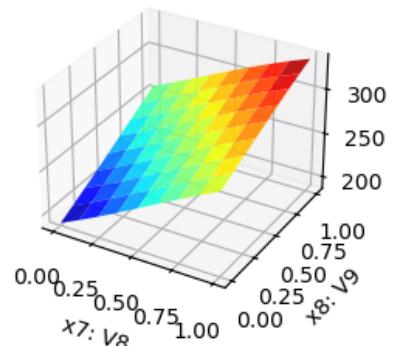
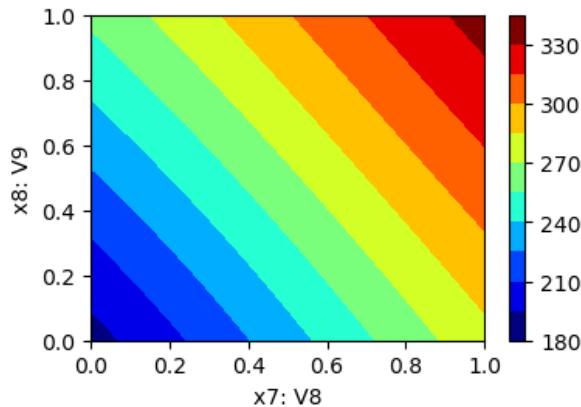
```
[rank: 0] Seed set to 123
[rank: 0] Seed set to 123
```

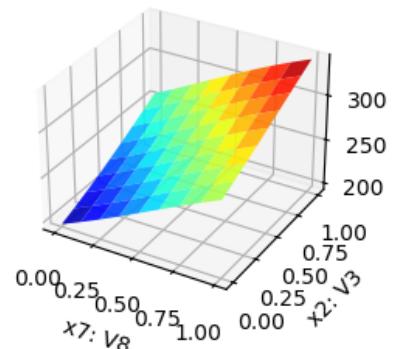
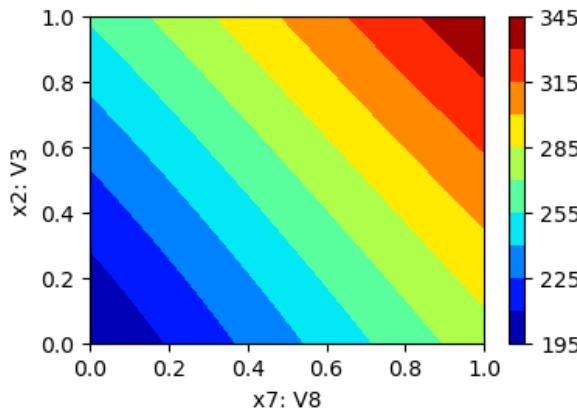
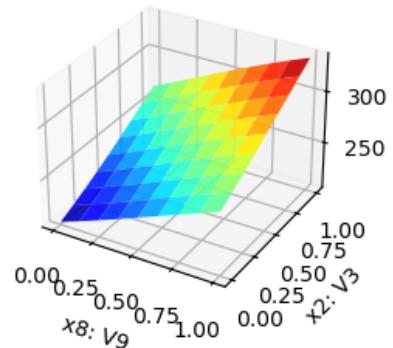
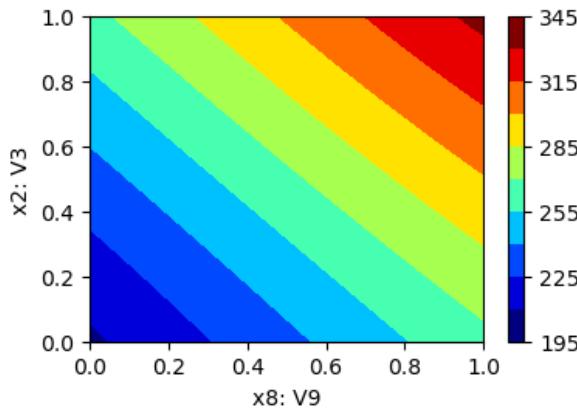
```
Created spot_tensorboard_path: runs/spot_logs/10_c02-b03_2024-05-20_18-3  
7-55 for SummaryWriter()  
spotPython tuning: 130.98167265774384 [#####---] 73.33%  
spotPython tuning: 124.96150746789587 [#####---] 80.00%  
spotPython tuning: 123.38367978743302 [#####---] 86.67%  
spotPython tuning: 123.38367978743302 [#####---] 93.33%  
spotPython tuning: 123.38367978743302 [#####---] 100.00% Done...  
  
{'CHECKPOINT_PATH': 'runs/saved_models/',  
 'DATASET_PATH': 'data/',  
 'PREFIX': '10',  
 'RESULTS_PATH': 'results/',  
 'TENSORBOARD_PATH': 'runs/',  
 '_L_in': None,  
 '_L_out': None,  
 '_torchmetric': None,  
 'accelerator': 'auto',  
 'converters': None,  
 'core_model': None,  
 'core_model_name': None,  
 'counter': 15,  
 'data': None,  
 'data_dir': './data',  
 'data_module': None,  
 'data_set': None,  
 'data_set_name': None,  
 'db_dict_name': None,  
 'design': None,  
 'device': None,  
 'devices': 1,  
 'enable_progress_bar': False,  
 'eval': None,  
 'fun_evals': 15,  
 'fun_repeats': 1,  
 'horizon': None,  
 'infill_criterion': 'y',  
 'k_folds': 3,  
 'log_graph': False,  
 'log_level': 50,  
 'loss_function': None,  
 'lower': array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]),  
 'max_surrogate_points': 30,  
 'max_time': 1,  
 'metric_params': {},  
 'metric_river': None,  
 'metric_sklearn': None,  
 'metric_sklearn_name': None,  
 'metric_torch': None,  
 'model_dict': {},  
 'n_points': 1,  
 'n_samples': None,  
 'n_total': None,  
 'noise': False,  
 'num_workers': 0,  
 'ocba_delta': 0,  
 'oml_grace_period': None,  
 'optimizer': None,  
 'path': None,  
 'prep_model': None,  
 'prep_model_name': None,
```

```

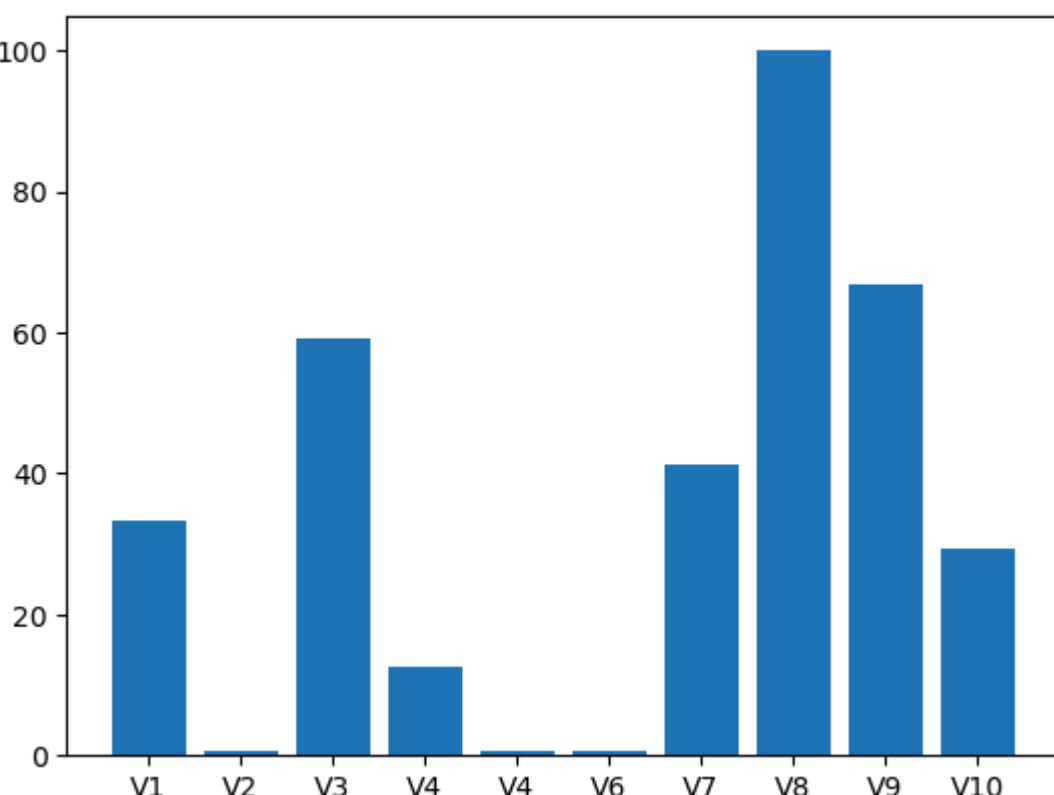
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/10_c02-b03_2024-05-20_18-37-5
5',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x1513fc309410>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.]),
'var_name': ['V1', 'V2', 'V3', 'V4', 'V4', 'V6', 'V7', 'V8', 'V9', 'V1
0'],
'var_type': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 123.38367978743302
V1: 0.0
V2: 1.0
V3: 0.0
V4: 0.5056764397096142
V4: 0.0
V6: 0.0
V7: 1.0
V8: 0.0
V9: 0.0
V10: 0.0

```





V1: 33.21379512913731
 V2: 0.5377341606911471
 V3: 59.25092727660866
 V4: 12.628480950437131
 V4: 0.5377341606911471
 V6: 0.5377341606911471
 V7: 41.221496564998425
 V8: 100.0
 V9: 66.7691019669039
 V10: 29.46506485851672



3. The Three Dimensional fun_runge

- The input dimension is 3. The search range is $-5 \leq x \leq 5$ for all dimensions.
- Generate contour plots
- Calculate the variable importance.
- Discuss the variable importance:
 - Are all variables equally important?
 - If not:
 - Which is the most important variable?
 - Which is the least important variable?

The fun_runge is typically a reference to the Runge function, which is a well-known example in numerical analysis and approximation theory. The Runge function is often used to demonstrate the phenomenon known as Runge's phenomenon, which illustrates the problems of using high-degree polynomials for interpolation on equispaced nodes.

Definition The Runge function is defined as: $f(x) = 1/(1+25x^2)$

Characteristics Shape: The function has a peak at $x = 0$ and tails off to zero as x moves away from zero. Domain: It is often considered on the interval $[-1, 1]$. Interpolation Challenge: When trying to interpolate this function using polynomials at equally spaced points, the interpolant can oscillate wildly, particularly near the endpoints of the interval. This issue becomes more pronounced with higher-degree polynomials.

```
In [3]: import numpy as np
from spotPython.fun.objectivefunctions import analytical
from spotPython.utils.init import fun_control_init, surrogate_control_init
from spotPython.spot import spot
import matplotlib.pyplot as plt

# Define the function
fun = analytical().fun_runge
# Define variables as input to the function
fun_control = fun_control_init(
    PREFIX="10",
    lower = -5.0*np.ones(3),
    upper = 5.0*np.ones(3),
    var_name=["V1", "V2", "V3"],
    show_progress=True)
#Define a surrogate
surrogate_control = surrogate_control_init(n_theta=3)
#Give input to SPOT
spot_3 = spot.Spot(fun=fun,
                    fun_control=fun_control,
                    surrogate_control=surrogate_control)
#Run Spot
spot_3.run()
_ = spot_3.print_results()
spot_3.plot_contour(i=0, j=1)
```

```
spot_3.plot_contour(i=1, j=2)
spot_3.plot_contour(i=0, j=2)

_= spot_3.print_importance()
spot_3.plot_importance()
```

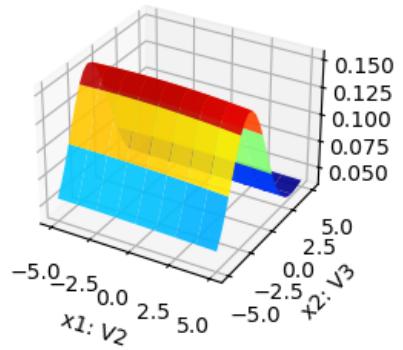
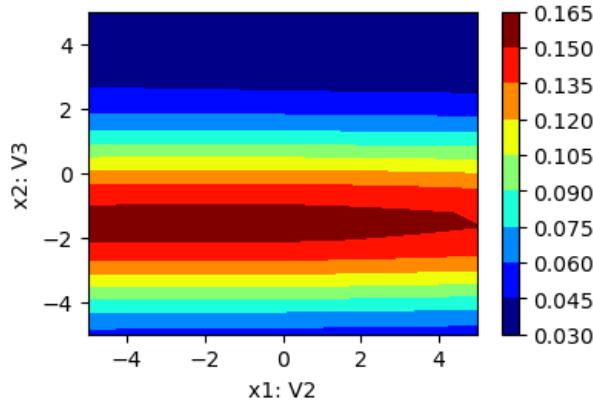
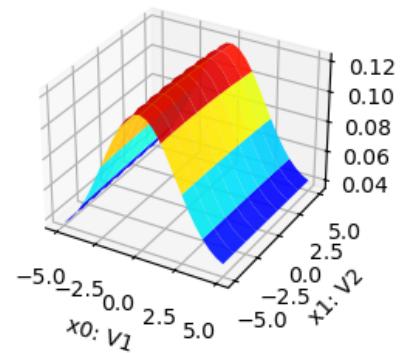
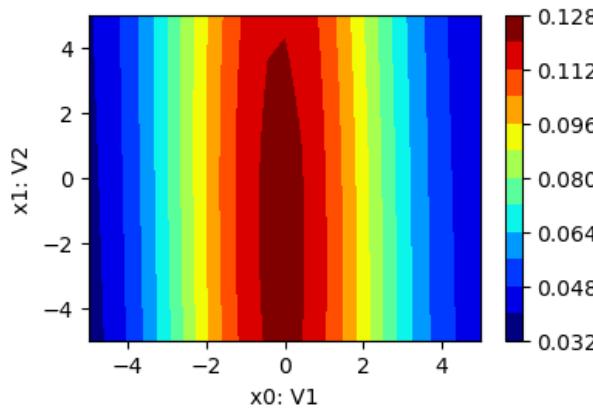
```
[rank: 0] Seed set to 123
```

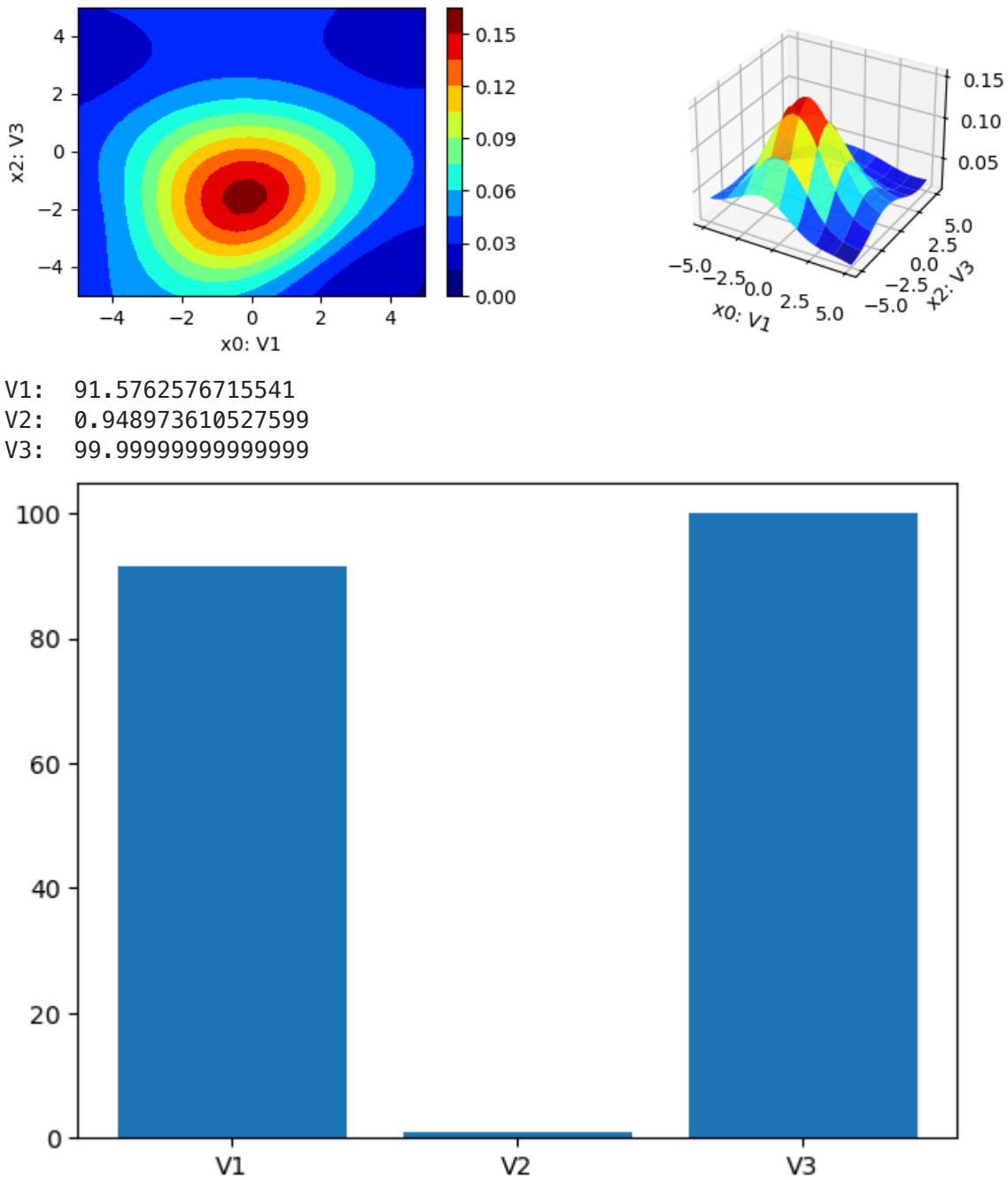
```
Created spot_tensorboard_path: runs/spot_logs/10_c02-b01_2024-05-19_22-5  
3-28 for SummaryWriter()  
spotPython tuning: 0.014160612902528853 [#####---] 73.33%  
spotPython tuning: 0.014160612902528853 [#####---] 80.00%  
spotPython tuning: 0.013157894736842105 [#####---] 86.67%  
spotPython tuning: 0.013157894736842105 [#####---] 93.33%  
spotPython tuning: 0.013157894736842105 [#####---] 100.00% Done...  
  
{'CHECKPOINT_PATH': 'runs/saved_models/',  
 'DATASET_PATH': 'data/',  
 'PREFIX': '10',  
 'RESULTS_PATH': 'results/',  
 'TENSORBOARD_PATH': 'runs/',  
 '_L_in': None,  
 '_L_out': None,  
 '_torchmetric': None,  
 'accelerator': 'auto',  
 'converters': None,  
 'core_model': None,  
 'core_model_name': None,  
 'counter': 15,  
 'data': None,  
 'data_dir': './data',  
 'data_module': None,  
 'data_set': None,  
 'data_set_name': None,  
 'db_dict_name': None,  
 'design': None,  
 'device': None,  
 'devices': 1,  
 'enable_progress_bar': False,  
 'eval': None,  
 'fun_evals': 15,  
 'fun_repeats': 1,  
 'horizon': None,  
 'infill_criterion': 'y',  
 'k_folds': 3,  
 'log_graph': False,  
 'log_level': 50,  
 'loss_function': None,  
 'lower': array([-5., -5., -5.]),  
 'max_surrogate_points': 30,  
 'max_time': 1,  
 'metric_params': {},  
 'metric_river': None,  
 'metric_sklearn': None,  
 'metric_sklearn_name': None,  
 'metric_torch': None,  
 'model_dict': {},  
 'n_points': 1,  
 'n_samples': None,  
 'n_total': None,  
 'noise': False,  
 'num_workers': 0,  
 'ocba_delta': 0,  
 'oml_grace_period': None,  
 'optimizer': None,  
 'path': None,  
 'prep_model': None,  
 'prep_model_name': None},
```

```

'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/10_c02-b01_2024-05-19_22-53-2
8',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x148d380e5a50>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([5., 5., 5.]),
'var_name': ['V1', 'V2', 'V3'],
'var_type': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 0.013157894736842105
V1: 5.0
V2: 5.0
V3: -5.0

```





4. The Three Dimensional `fun_linear`

- The input dimension is 3. The search range is $-5 \leq x \leq 5$ for all dimensions.
- Generate contour plots
- Calculate the variable importance.
- Discuss the variable importance:
 - Are all variables equally important?
 - If not:
 - Which is the most important variable?
 - Which is the least important variable?

Definition A linear function is one of the simplest types of functions and is defined by the general form: $f(x) = mx$

•
b

where

m is the slope and

b is the y-intercept.

For a specific example, we might define a linear function as: $f(x) = 2x + 3$

•
3

where the slope

m is 2, and the y-intercept

b is 3.

Characteristics Shape: The function produces a straight line when graphed. Slope (m): Determines the steepness of the line. A positive slope means the line ascends from left to right, while a negative slope means it descends. Intercept (b): The point where the line crosses the y-axis. Domain: The domain of a linear function is all real numbers, \mathbb{R} . Range: The range of a linear function is also all real numbers, \mathbb{R} . Linearity: Linear functions exhibit constant rates of change and are additive and scalable. This makes them easy to work with in optimization and modeling contexts. No Curvature: Unlike quadratic or higher-order polynomial functions, a linear function has no curvature (i.e., it is not curved, it is straight).

```
In [4]: import numpy as np
from spotPython.fun.objectivefunctions import analytical
from spotPython.utils.init import fun_control_init, surrogate_control_init
from spotPython.spot import spot
import matplotlib.pyplot as plt

# Define the function
fun = analytical().fun_linear
# Define variables as input to the function
fun_control = fun_control_init(
    PREFIX="10",
    lower = -5.0*np.ones(3),
    upper = 5.0*np.ones(3),
    var_name=["V1", "V2", "V3"],
    show_progress=True)
#Define a surrogate
surrogate_control = surrogate_control_init(n_theta=3)
#Give input to SPOT
spot_3 = spot.Spot(fun=fun,
                    fun_control=fun_control,
                    surrogate_control=surrogate_control)
#Run Spot
spot_3.run()
```

```
_ = spot_3.print_results()
spot_3.plot_contour(i=0, j=1)
spot_3.plot_contour(i=1, j=2)
spot_3.plot_contour(i=0, j=2)

_= spot_3.print_importance()
spot_3.plot_importance()
```

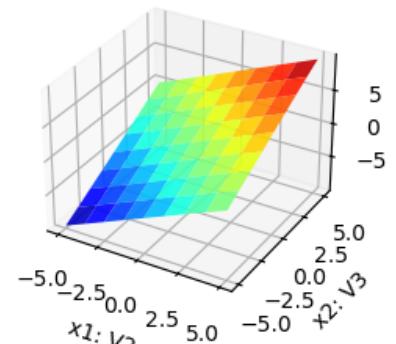
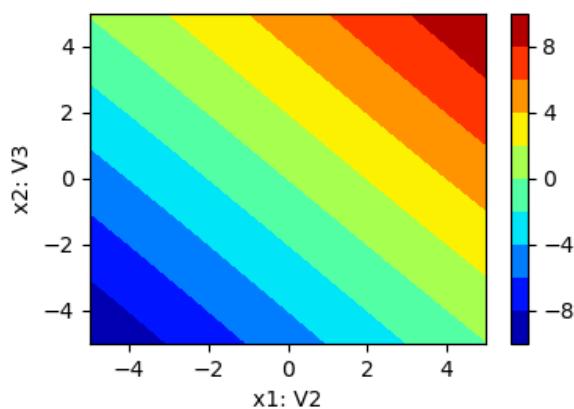
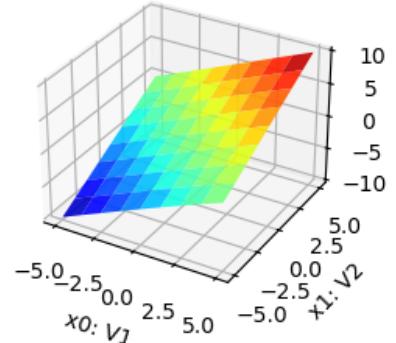
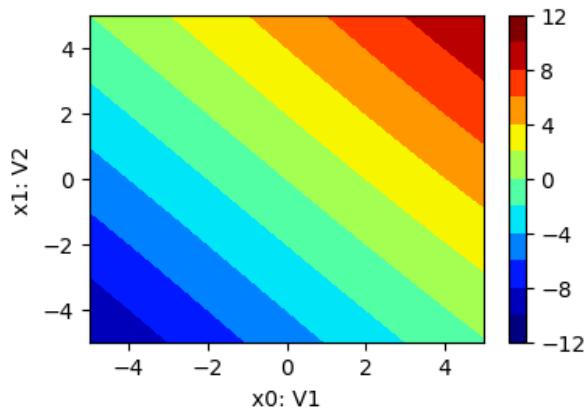
[rank: 0] Seed set to 123

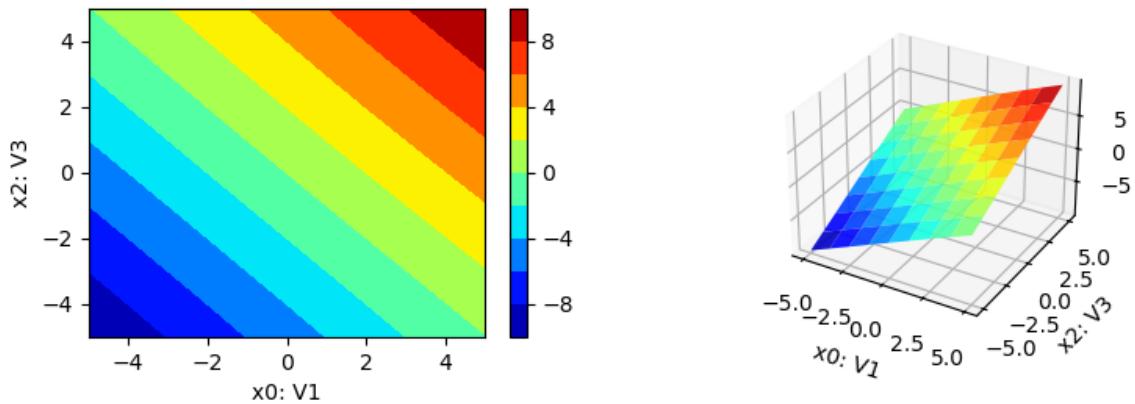
```
Created spot_tensorboard_path: runs/spot_logs/10_c02-b01_2024-05-19_22-5  
3-36 for SummaryWriter()  
spotPython tuning: -15.0 [#####----] 73.33%  
spotPython tuning: -15.0 [#####----] 80.00%  
spotPython tuning: -15.0 [#####----] 86.67%  
spotPython tuning: -15.0 [#####----] 93.33%  
spotPython tuning: -15.0 [#####----] 100.00% Done...  
  
{'CHECKPOINT_PATH': 'runs/saved_models/',  
 'DATASET_PATH': 'data/',  
 'PREFIX': '10',  
 'RESULTS_PATH': 'results/',  
 'TENSORBOARD_PATH': 'runs/',  
 '_L_in': None,  
 '_L_out': None,  
 '_torchmetric': None,  
 'accelerator': 'auto',  
 'converters': None,  
 'core_model': None,  
 'core_model_name': None,  
 'counter': 15,  
 'data': None,  
 'data_dir': './data',  
 'data_module': None,  
 'data_set': None,  
 'data_set_name': None,  
 'db_dict_name': None,  
 'design': None,  
 'device': None,  
 'devices': 1,  
 'enable_progress_bar': False,  
 'eval': None,  
 'fun_evals': 15,  
 'fun_repeats': 1,  
 'horizon': None,  
 'infill_criterion': 'y',  
 'k_folds': 3,  
 'log_graph': False,  
 'log_level': 50,  
 'loss_function': None,  
 'lower': array([-5., -5., -5.]),  
 'max_surrogate_points': 30,  
 'max_time': 1,  
 'metric_params': {},  
 'metric_river': None,  
 'metric_sklearn': None,  
 'metric_sklearn_name': None,  
 'metric_torch': None,  
 'model_dict': {},  
 'n_points': 1,  
 'n_samples': None,  
 'n_total': None,  
 'noise': False,  
 'num_workers': 0,  
 'ocba_delta': 0,  
 'oml_grace_period': None,  
 'optimizer': None,  
 'path': None,  
 'prep_model': None,  
 'prep_model_name': None},
```

```

'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/10_c02-b01_2024-05-19_22-53-3
6',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x148c3ee67910>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([5., 5., 5.]),
'var_name': ['V1', 'V2', 'V3'],
'var_type': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: -15.0
V1: -5.0
V2: -5.0
V3: -5.0

```

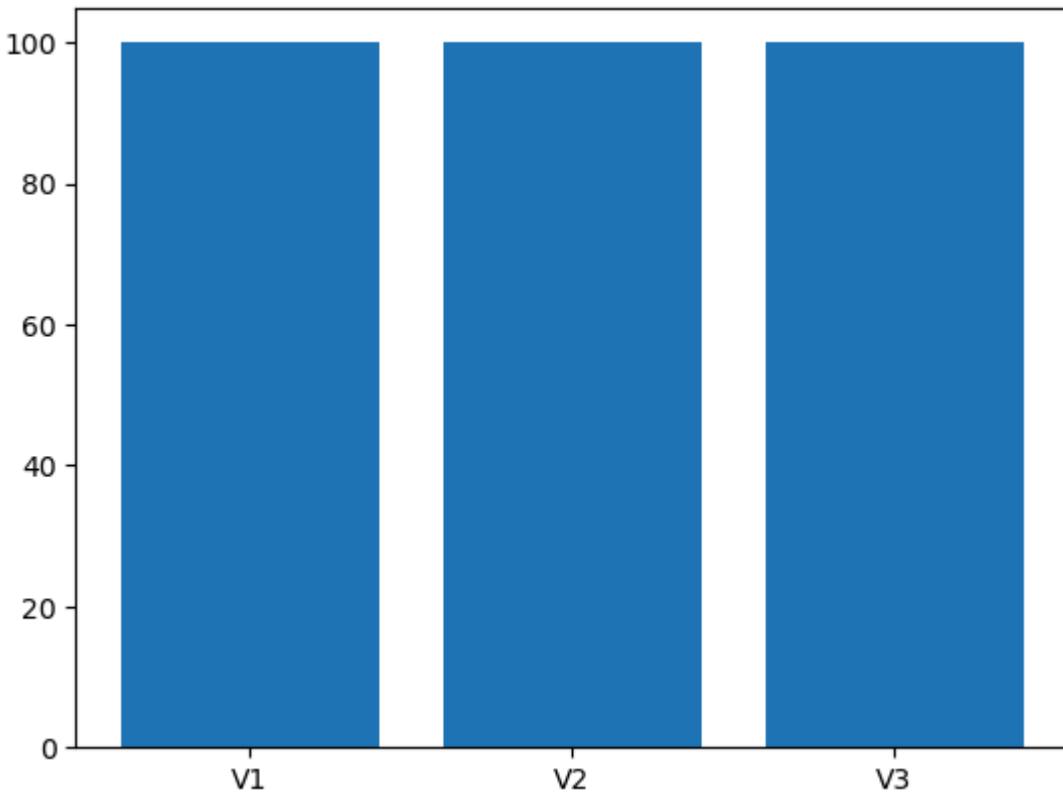




V1: 100.0

V2: 100.0

V3: 100.0



Exercises Chapter 09: Isotropic and Anisotropic Kriging

- The following exercises refer to chapter 09 of the book, see https://sequential-parameter-optimization.github.io/Hyperparameter-Tuning-Cookbook/009_num_spot_anisotropic.html.

1. The Branin Function `fun_branin`

```
y = a * (x2 - b * x1**2 + c * x1 - r)**2 + s * (1 - t) *
np.cos(x1) + s
```

where the values of the parameters are:

- a = 1
- b = 5.1 / (4 * pi**2)

```

- c = 5 / pi
- r = 6
- s = 10
- t = 1 / (8 * pi)

```

It has three global minima:

$f(x) = 0.397887$ at $(-\pi, 12.275)$, $(\pi, 2.275)$, and $(9.42478, 2.475)$.

- Describe the function.
 - The input dimension is 2. The search range is $-5 \leq x_1 \leq 10$ and $0 \leq x_2 \leq 15$.
- Compare the results from `spotPython` run a) with isotropic and b) anisotropic surrogate models.
- Modify the termination criterion: instead of the number of evaluations (which is specified via `fun_evals`), the time should be used as the termination criterion. This can be done as follows (`max_time=1` specifies a run time of one minute):

Isotropic and Anisotropic with time = 1 min

```
In [1]: import numpy as np
from math import inf
from spotPython.fun.objectivefunctions import analytical
from spotPython.spot import spot
from spotPython.utils.init import fun_control_init, surrogate_control_init
from math import inf
#Isotropic Branin Function
PREFIX="003"
fun = analytical().fun_branin
fun_control = fun_control_init(PREFIX=PREFIX,
                               fun_evals=inf,
                               max_time=1,
                               lower = np.array([-5, 0]),
                               upper = np.array([10, 15]))
surrogate_control=surrogate_control_init(n_theta=1)
spot_2 = spot.Spot(fun=fun,
                   fun_control=fun_control,
                   surrogate_control=surrogate_control)

spot_2.run()
spot_2.print_results()
spot_2.plot_progress()
spot_2.surrogate.plot()
spot_2.surrogate.theta
```

```
[rank: 0] Seed set to 123
[rank: 0] Seed set to 123
```

```
Created spot_tensorboard_path: runs/spot_logs/003_c02-b04_2024-05-21_13-23-18 for SummaryWriter()
spotPython tuning: 3.146896546946783 [-----] 0.31%
spotPython tuning: 3.146896546946783 [-----] 0.62%
spotPython tuning: 3.146896546946783 [-----] 0.99%
spotPython tuning: 3.146896546946783 [-----] 1.35%
spotPython tuning: 1.1487866236306008 [-----] 1.73%
spotPython tuning: 1.0240824742350494 [-----] 2.11%
spotPython tuning: 0.4211249770550065 [-----] 2.55%
spotPython tuning: 0.40198655236716263 [-----] 3.03%
spotPython tuning: 0.3991783652663141 [-----] 3.61%
spotPython tuning: 0.3991783652663141 [-----] 4.11%
spotPython tuning: 0.3991783652663141 [-----] 4.57%
spotPython tuning: 0.3991783652663141 [#-----] 5.01%
spotPython tuning: 0.3991783652663141 [#-----] 5.51%
spotPython tuning: 0.3991783652663141 [#-----] 6.01%
spotPython tuning: 0.3991783652663141 [#-----] 6.62%
spotPython tuning: 0.39883116428530485 [#-----] 7.33%
spotPython tuning: 0.39883116428530485 [#-----] 8.06%
spotPython tuning: 0.39830236020791254 [#-----] 8.79%
spotPython tuning: 0.3981492965243003 [#-----] 9.51%
spotPython tuning: 0.3981492965243003 [#-----] 10.25%
spotPython tuning: 0.3981492965243003 [#-----] 10.96%
spotPython tuning: 0.3981492965243003 [#-----] 11.59%
spotPython tuning: 0.3981492965243003 [#-----] 12.16%
spotPython tuning: 0.3981492965243003 [#-----] 12.85%
spotPython tuning: 0.3981492965243003 [#-----] 13.55%
spotPython tuning: 0.3981492965243003 [#-----] 14.20%
spotPython tuning: 0.3981492965243003 [#-----] 14.86%
spotPython tuning: 0.3981492965243003 [##-----] 15.50%
spotPython tuning: 0.3981492965243003 [##-----] 16.24%
spotPython tuning: 0.3981492965243003 [##-----] 16.95%
spotPython tuning: 0.3981492965243003 [##-----] 17.71%
spotPython tuning: 0.3981492965243003 [##-----] 18.50%
spotPython tuning: 0.3981492965243003 [##-----] 19.15%
spotPython tuning: 0.3981492965243003 [##-----] 19.85%
spotPython tuning: 0.3981492965243003 [##-----] 20.47%
spotPython tuning: 0.3981492965243003 [##-----] 21.16%
spotPython tuning: 0.3981492965243003 [##-----] 21.80%
spotPython tuning: 0.3981492965243003 [##-----] 22.43%
spotPython tuning: 0.3981492965243003 [##-----] 23.03%
spotPython tuning: 0.3981492965243003 [##-----] 23.64%
spotPython tuning: 0.3981492965243003 [##-----] 24.25%
spotPython tuning: 0.3981492965243003 [###-----] 25.18%
spotPython tuning: 0.3981492965243003 [###-----] 25.82%
spotPython tuning: 0.3981492965243003 [###-----] 26.38%
spotPython tuning: 0.3981492965243003 [###-----] 27.01%
spotPython tuning: 0.3981492965243003 [###-----] 27.61%
spotPython tuning: 0.3981492965243003 [###-----] 28.34%
spotPython tuning: 0.3981492965243003 [###-----] 29.07%
spotPython tuning: 0.3981492965243003 [###-----] 29.75%
spotPython tuning: 0.3981367922869037 [###-----] 30.38%
spotPython tuning: 0.3981367922869037 [###-----] 31.09%
spotPython tuning: 0.3981272085911698 [###-----] 31.71%
spotPython tuning: 0.3981260363674277 [###-----] 32.36%
spotPython tuning: 0.39812562159481324 [###-----] 32.97%
spotPython tuning: 0.3981243574293405 [###-----] 33.63%
spotPython tuning: 0.3981243574293405 [###-----] 34.29%
spotPython tuning: 0.39812127569033073 [###-----] 34.93%
spotPython tuning: 0.39812127569033073 [###-----] 35.59%
```

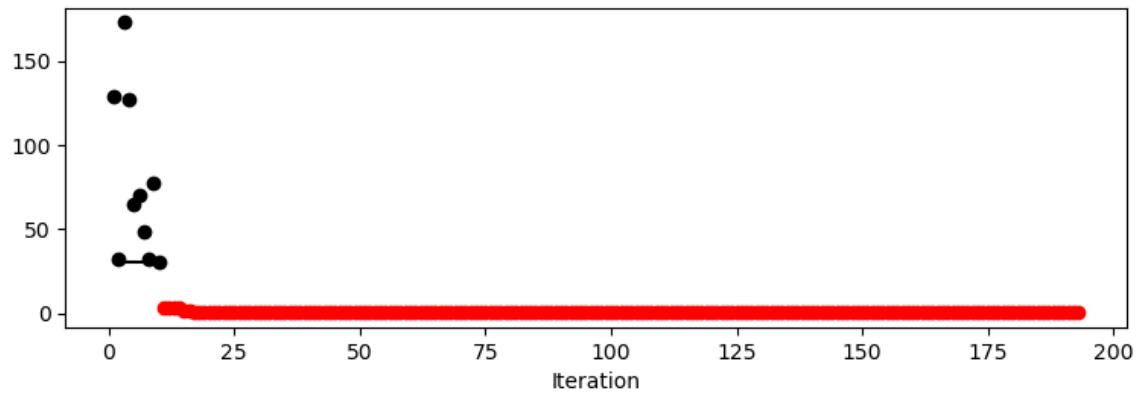
```
spotPython tuning: 0.39812127569033073 [#####-----] 36.26%
spotPython tuning: 0.39812127569033073 [#####-----] 36.86%
spotPython tuning: 0.39812127569033073 [#####-----] 37.47%
spotPython tuning: 0.39812127569033073 [#####-----] 38.06%
spotPython tuning: 0.39812127569033073 [#####-----] 38.61%
spotPython tuning: 0.39812127569033073 [#####-----] 39.32%
spotPython tuning: 0.39812127569033073 [#####-----] 40.07%
spotPython tuning: 0.39812127569033073 [#####-----] 40.74%
spotPython tuning: 0.39801594419739494 [#####-----] 41.33%
spotPython tuning: 0.39801503764417845 [#####-----] 42.01%
spotPython tuning: 0.3980073997081295 [#####-----] 42.67%
spotPython tuning: 0.3980073997081295 [#####-----] 43.42%
spotPython tuning: 0.3980073997081295 [#####-----] 44.03%
spotPython tuning: 0.39800549677289965 [#####-----] 45.03%
spotPython tuning: 0.39800275534149065 [#####-----] 45.61%
spotPython tuning: 0.39800275534149065 [#####-----] 46.19%
spotPython tuning: 0.39800275534149065 [#####-----] 46.69%
spotPython tuning: 0.397957234435637 [#####-----] 47.25%
spotPython tuning: 0.397957234435637 [#####-----] 47.79%
spotPython tuning: 0.397957234435637 [#####-----] 48.30%
spotPython tuning: 0.397957234435637 [#####-----] 48.78%
spotPython tuning: 0.397957234435637 [#####-----] 49.25%
spotPython tuning: 0.397957234435637 [#####-----] 49.74%
spotPython tuning: 0.397957234435637 [#####-----] 50.21%
spotPython tuning: 0.397957234435637 [#####-----] 50.68%
spotPython tuning: 0.397957234435637 [#####-----] 51.14%
spotPython tuning: 0.39795686381840234 [#####-----] 51.62%
spotPython tuning: 0.39795686381840234 [#####-----] 52.16%
spotPython tuning: 0.39794039762593947 [#####-----] 52.65%
spotPython tuning: 0.39794039762593947 [#####-----] 53.13%
spotPython tuning: 0.39794039762593947 [#####-----] 53.61%
spotPython tuning: 0.39790758969199125 [#####-----] 54.08%
spotPython tuning: 0.39790758969199125 [#####-----] 54.55%
spotPython tuning: 0.39790758969199125 [#####-----] 55.01%
spotPython tuning: 0.39790758969199125 [#####-----] 55.48%
spotPython tuning: 0.39790758969199125 [#####-----] 55.94%
spotPython tuning: 0.39790751986817696 [#####-----] 56.40%
spotPython tuning: 0.39790751986817696 [#####-----] 56.89%
spotPython tuning: 0.39790454173192735 [#####-----] 57.37%
spotPython tuning: 0.39790454173192735 [#####-----] 57.85%
spotPython tuning: 0.39790454173192735 [#####-----] 58.32%
spotPython tuning: 0.39790454173192735 [#####-----] 58.78%
spotPython tuning: 0.39790454173192735 [#####-----] 59.35%
spotPython tuning: 0.39790454173192735 [#####-----] 59.81%
spotPython tuning: 0.39790454173192735 [#####-----] 60.30%
spotPython tuning: 0.39790454173192735 [#####-----] 60.82%
spotPython tuning: 0.39790454173192735 [#####-----] 61.36%
spotPython tuning: 0.39790454173192735 [#####-----] 61.86%
spotPython tuning: 0.39790454173192735 [#####-----] 62.50%
spotPython tuning: 0.39790454173192735 [#####-----] 63.05%
spotPython tuning: 0.39790454173192735 [#####-----] 63.59%
spotPython tuning: 0.39790454173192735 [#####-----] 64.16%
spotPython tuning: 0.39790454173192735 [#####-----] 64.59%
spotPython tuning: 0.39790454173192735 [#####-----] 65.14%
spotPython tuning: 0.39790454173192735 [#####-----] 65.68%
spotPython tuning: 0.39790454173192735 [#####-----] 66.21%
spotPython tuning: 0.39790454173192735 [#####-----] 66.71%
spotPython tuning: 0.39790454173192735 [#####-----] 67.27%
spotPython tuning: 0.39790454173192735 [#####-----] 67.79%
spotPython tuning: 0.39790454173192735 [#####-----] 68.29%
```

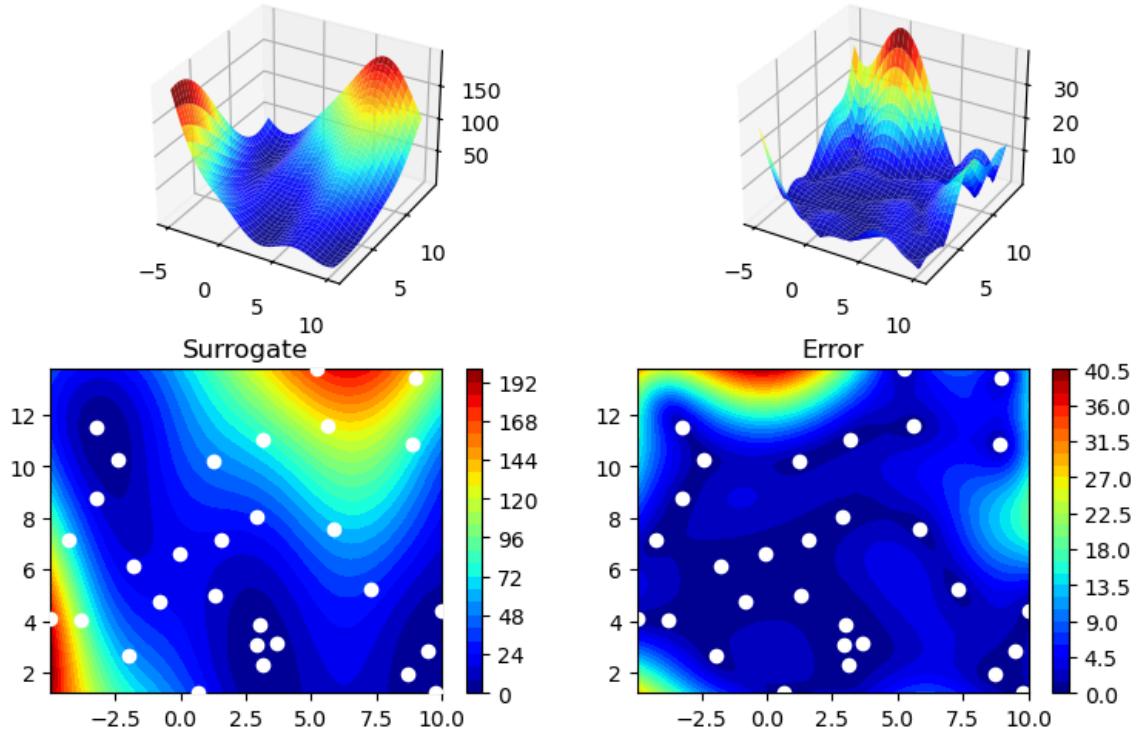
```
spotPython tuning: 0.39790454173192735 [#####----] 68.81%
spotPython tuning: 0.39790454173192735 [#####----] 69.34%
spotPython tuning: 0.39790454173192735 [#####----] 69.80%
spotPython tuning: 0.39790454173192735 [#####----] 70.26%
spotPython tuning: 0.39790454173192735 [#####----] 70.75%
spotPython tuning: 0.39790454173192735 [#####----] 71.20%
spotPython tuning: 0.39790454173192735 [#####----] 71.66%
spotPython tuning: 0.39790454173192735 [#####----] 72.12%
spotPython tuning: 0.39790454173192735 [#####----] 72.58%
spotPython tuning: 0.39790454173192735 [#####----] 73.17%
spotPython tuning: 0.39790454173192735 [#####----] 73.70%
spotPython tuning: 0.39790454173192735 [#####----] 74.27%
spotPython tuning: 0.39790454173192735 [#####----] 74.85%
spotPython tuning: 0.39790454173192735 [#####----] 75.49%
spotPython tuning: 0.39790454173192735 [#####----] 76.04%
spotPython tuning: 0.39790454173192735 [#####----] 76.63%
spotPython tuning: 0.39790454173192735 [#####----] 77.26%
spotPython tuning: 0.39790454173192735 [#####----] 77.74%
spotPython tuning: 0.39790454173192735 [#####----] 78.22%
spotPython tuning: 0.39790454173192735 [#####----] 78.78%
spotPython tuning: 0.39790454173192735 [#####----] 79.41%
spotPython tuning: 0.39790454173192735 [#####----] 79.87%
spotPython tuning: 0.39790454173192735 [#####----] 80.29%
spotPython tuning: 0.39790454173192735 [#####----] 80.74%
spotPython tuning: 0.39790454173192735 [#####----] 81.15%
spotPython tuning: 0.39790454173192735 [#####----] 81.61%
spotPython tuning: 0.39790454173192735 [#####----] 82.16%
spotPython tuning: 0.39790454173192735 [#####----] 82.70%
spotPython tuning: 0.39790454173192735 [#####----] 83.17%
spotPython tuning: 0.39790454173192735 [#####----] 83.57%
spotPython tuning: 0.39790454173192735 [#####----] 84.06%
spotPython tuning: 0.39790454173192735 [#####----] 84.50%
spotPython tuning: 0.39790454173192735 [#####----] 84.95%
spotPython tuning: 0.39790454173192735 [#####----] 85.43%
spotPython tuning: 0.39790454173192735 [#####----] 85.91%
spotPython tuning: 0.39790454173192735 [#####----] 86.36%
spotPython tuning: 0.39790454173192735 [#####----] 86.85%
spotPython tuning: 0.39790454173192735 [#####----] 87.33%
spotPython tuning: 0.39790454173192735 [#####----] 87.79%
spotPython tuning: 0.39790454173192735 [#####----] 88.28%
spotPython tuning: 0.39790454173192735 [#####----] 88.78%
spotPython tuning: 0.39790454173192735 [#####----] 89.27%
spotPython tuning: 0.39790454173192735 [#####----] 89.75%
spotPython tuning: 0.39790454173192735 [#####----] 90.21%
spotPython tuning: 0.39790454173192735 [#####----] 90.72%
spotPython tuning: 0.39790454173192735 [#####----] 91.16%
spotPython tuning: 0.39790454173192735 [#####----] 91.66%
spotPython tuning: 0.39790454173192735 [#####----] 92.14%
spotPython tuning: 0.39790454173192735 [#####----] 92.62%
spotPython tuning: 0.39790454173192735 [#####----] 93.17%
spotPython tuning: 0.39790454173192735 [#####----] 93.67%
spotPython tuning: 0.39790454173192735 [#####----] 94.18%
spotPython tuning: 0.39790454173192735 [#####----] 94.72%
spotPython tuning: 0.39790454173192735 [#####----] 95.27%
spotPython tuning: 0.39790454173192735 [#####----] 95.70%
spotPython tuning: 0.39790454173192735 [#####----] 96.15%
spotPython tuning: 0.39790454173192735 [#####----] 96.69%
spotPython tuning: 0.39790454173192735 [#####----] 97.22%
spotPython tuning: 0.39790454173192735 [#####----] 97.68%
spotPython tuning: 0.39790454173192735 [#####----] 98.14%
```

```
spotPython tuning: 0.39790454173192735 [#####] 98.60%
spotPython tuning: 0.39790454173192735 [#####] 99.03%
spotPython tuning: 0.39790454173192735 [#####] 99.48%
spotPython tuning: 0.39790454173192735 [#####] 99.97%
spotPython tuning: 0.39790454173192735 [#####] 100.00% Done...

{'CHECKPOINT_PATH': 'runs/saved_models/',
 'DATASET_PATH': 'data/',
 'PREFIX': '003',
 'RESULTS_PATH': 'results/',
 'TENSORBOARD_PATH': 'runs/',
 '_L_in': None,
 '_L_out': None,
 '_torchmetric': None,
 'accelerator': 'auto',
 'converters': None,
 'core_model': None,
 'core_model_name': None,
 'counter': 193,
 'data': None,
 'data_dir': './data',
 'data_module': None,
 'data_set': None,
 'data_set_name': None,
 'db_dict_name': None,
 'design': None,
 'device': None,
 'devices': 1,
 'enable_progress_bar': False,
 'eval': None,
 'fun_evals': inf,
 'fun_repeats': 1,
 'horizon': None,
 'infill_criterion': 'y',
 'k_folds': 3,
 'log_graph': False,
 'log_level': 50,
 'loss_function': None,
 'lower': array([-5, 0]),
 'max_surrogate_points': 30,
 'max_time': 1,
 'metric_params': {},
 'metric_river': None,
 'metric_sklearn': None,
 'metric_sklearn_name': None,
 'metric_torch': None,
 'model_dict': {},
 'n_points': 1,
 'n_samples': None,
 'n_total': None,
 'noise': False,
 'num_workers': 0,
 'ocba_delta': 0,
 'oml_grace_period': None,
 'optimizer': None,
 'path': None,
 'prep_model': None,
 'prep_model_name': None,
 'progress_file': None,
 'save_model': False,
```

```
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/003_c02-b04_2024-05-21_13-23-1
8',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x154d40ec5a10>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([10, 15]),
'ver_name': None,
'ver_type': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 0.39790454173192735
x0: 3.140185403945475
x1: 2.2788683124450957
```





Out[1]: array([-1.42962125])

```
In [3]: #Anisotropic Branin Function
surrogate_control = surrogate_control_init(n_theta=2)
spot_2_anisotropic = spot.Spot(fun=fun,
                               fun_control=fun_control,
                               surrogate_control=surrogate_control)
spot_2_anisotropic.run()
spot_2_anisotropic.print_results()
spot_2_anisotropic.surrogate.plot()
spot_2_anisotropic.surrogate.theta
spot_2_anisotropic.plot_progress()
```

```
spotPython tuning: 3.800462807494956 [-----] 0.41%
spotPython tuning: 3.800462807494956 [-----] 0.87%
spotPython tuning: 3.1589357387738524 [-----] 1.35%
spotPython tuning: 3.134400412616152 [-----] 1.82%
spotPython tuning: 2.9066716414049374 [-----] 2.38%
spotPython tuning: 0.4143165308821022 [-----] 3.02%
spotPython tuning: 0.4041224802204155 [-----] 3.61%
spotPython tuning: 0.3987714674290057 [-----] 4.24%
spotPython tuning: 0.3987714674290057 [-----] 4.85%
spotPython tuning: 0.3987714674290057 [#-----] 5.51%
spotPython tuning: 0.3987714674290057 [#-----] 6.14%
spotPython tuning: 0.3987714674290057 [#-----] 6.98%
spotPython tuning: 0.3987714674290057 [#-----] 7.64%
spotPython tuning: 0.3987714674290057 [#-----] 8.48%
spotPython tuning: 0.3987714674290057 [#-----] 9.28%
spotPython tuning: 0.3984374184973323 [#-----] 10.06%
spotPython tuning: 0.39794164248837127 [#-----] 11.10%
spotPython tuning: 0.39794164248837127 [#-----] 12.11%
spotPython tuning: 0.39794164248837127 [#-----] 13.03%
spotPython tuning: 0.39794164248837127 [#-----] 14.03%
spotPython tuning: 0.39794164248837127 [##-----] 15.09%
spotPython tuning: 0.39794164248837127 [##-----] 16.19%
spotPython tuning: 0.39794164248837127 [##-----] 17.18%
spotPython tuning: 0.39794164248837127 [##-----] 18.22%
spotPython tuning: 0.39794164248837127 [##-----] 19.23%
spotPython tuning: 0.39794164248837127 [##-----] 20.22%
spotPython tuning: 0.39794164248837127 [##-----] 21.17%
spotPython tuning: 0.39794164248837127 [##-----] 22.01%
spotPython tuning: 0.39794164248837127 [##-----] 23.07%
spotPython tuning: 0.39794164248837127 [##-----] 24.03%
spotPython tuning: 0.39794164248837127 [##-----] 25.06%
spotPython tuning: 0.39794164248837127 [##-----] 26.06%
spotPython tuning: 0.39794164248837127 [##-----] 26.97%
spotPython tuning: 0.39794164248837127 [##-----] 27.98%
spotPython tuning: 0.39794164248837127 [##-----] 28.96%
spotPython tuning: 0.39794164248837127 [##-----] 29.97%
spotPython tuning: 0.39794164248837127 [##-----] 31.00%
spotPython tuning: 0.39794164248837127 [##-----] 32.12%
spotPython tuning: 0.39794164248837127 [##-----] 33.18%
spotPython tuning: 0.39794164248837127 [##-----] 34.12%
spotPython tuning: 0.39794164248837127 [####-----] 35.10%
spotPython tuning: 0.39794164248837127 [####-----] 36.08%
spotPython tuning: 0.39794164248837127 [####-----] 37.08%
spotPython tuning: 0.39794164248837127 [####-----] 38.01%
spotPython tuning: 0.39794164248837127 [####-----] 38.95%
spotPython tuning: 0.39794164248837127 [####-----] 39.90%
spotPython tuning: 0.39794164248837127 [####-----] 40.83%
spotPython tuning: 0.39794164248837127 [####-----] 41.85%
spotPython tuning: 0.39794164248837127 [####-----] 42.79%
spotPython tuning: 0.39794164248837127 [####-----] 43.72%
spotPython tuning: 0.39794164248837127 [####-----] 44.63%
spotPython tuning: 0.39794164248837127 [#####-----] 45.56%
spotPython tuning: 0.39794164248837127 [#####-----] 46.58%
spotPython tuning: 0.39794164248837127 [#####-----] 47.57%
spotPython tuning: 0.39794164248837127 [#####-----] 48.56%
spotPython tuning: 0.39794164248837127 [#####-----] 49.95%
spotPython tuning: 0.39794164248837127 [#####-----] 51.09%
spotPython tuning: 0.39794164248837127 [#####-----] 52.37%
spotPython tuning: 0.39794164248837127 [#####-----] 53.72%
spotPython tuning: 0.39794164248837127 [#####-----] 54.81%
```

```

spotPython tuning: 0.39794164248837127 [#####----] 56.04%
spotPython tuning: 0.39794164248837127 [#####----] 57.09%
spotPython tuning: 0.39794164248837127 [#####----] 58.21%
spotPython tuning: 0.39794164248837127 [#####----] 59.32%
spotPython tuning: 0.39794164248837127 [#####----] 60.33%
spotPython tuning: 0.39794164248837127 [#####----] 61.34%
spotPython tuning: 0.39794164248837127 [#####----] 62.35%
spotPython tuning: 0.39794164248837127 [#####----] 63.46%
spotPython tuning: 0.39794164248837127 [#####----] 64.52%
spotPython tuning: 0.39794164248837127 [#####----] 65.60%
spotPython tuning: 0.39794164248837127 [#####----] 66.68%
spotPython tuning: 0.39794164248837127 [#####----] 67.98%
spotPython tuning: 0.397926589481596 [#####----] 69.04%
spotPython tuning: 0.397926589481596 [#####----] 70.13%
spotPython tuning: 0.397926589481596 [#####----] 71.22%
spotPython tuning: 0.3979252227079453 [#####----] 72.47%
spotPython tuning: 0.3979252227079453 [#####----] 73.56%
spotPython tuning: 0.3979252227079453 [#####----] 74.64%
spotPython tuning: 0.3979252227079453 [#####----] 75.64%
spotPython tuning: 0.3979252227079453 [#####----] 76.66%
spotPython tuning: 0.3979252227079453 [#####----] 77.62%
spotPython tuning: 0.3979252227079453 [#####----] 78.60%
spotPython tuning: 0.3979252227079453 [#####----] 79.67%
spotPython tuning: 0.3979252227079453 [#####----] 80.62%
spotPython tuning: 0.3979252227079453 [#####----] 81.61%
spotPython tuning: 0.3979252227079453 [#####----] 82.75%
spotPython tuning: 0.3979252227079453 [#####----] 83.79%
spotPython tuning: 0.3979252227079453 [#####----] 84.82%
spotPython tuning: 0.3979252227079453 [#####----] 85.79%
spotPython tuning: 0.3979252227079453 [#####----] 86.74%
spotPython tuning: 0.3979252227079453 [#####----] 87.74%
spotPython tuning: 0.3979252227079453 [#####----] 88.71%
spotPython tuning: 0.3979252227079453 [#####----] 89.67%
spotPython tuning: 0.3979252227079453 [#####----] 90.64%
spotPython tuning: 0.3979252227079453 [#####----] 91.60%
spotPython tuning: 0.3979252227079453 [#####----] 92.60%
spotPython tuning: 0.3979252227079453 [#####----] 93.55%
spotPython tuning: 0.3979252227079453 [#####----] 94.45%
spotPython tuning: 0.3979252227079453 [#####----] 95.35%
spotPython tuning: 0.3979252227079453 [#####----] 96.31%
spotPython tuning: 0.3979252227079453 [#####----] 97.22%
spotPython tuning: 0.3979252227079453 [#####----] 98.17%
spotPython tuning: 0.3979252227079453 [#####----] 99.07%
spotPython tuning: 0.3979252227079453 [#####----] 100.00% Done...

```

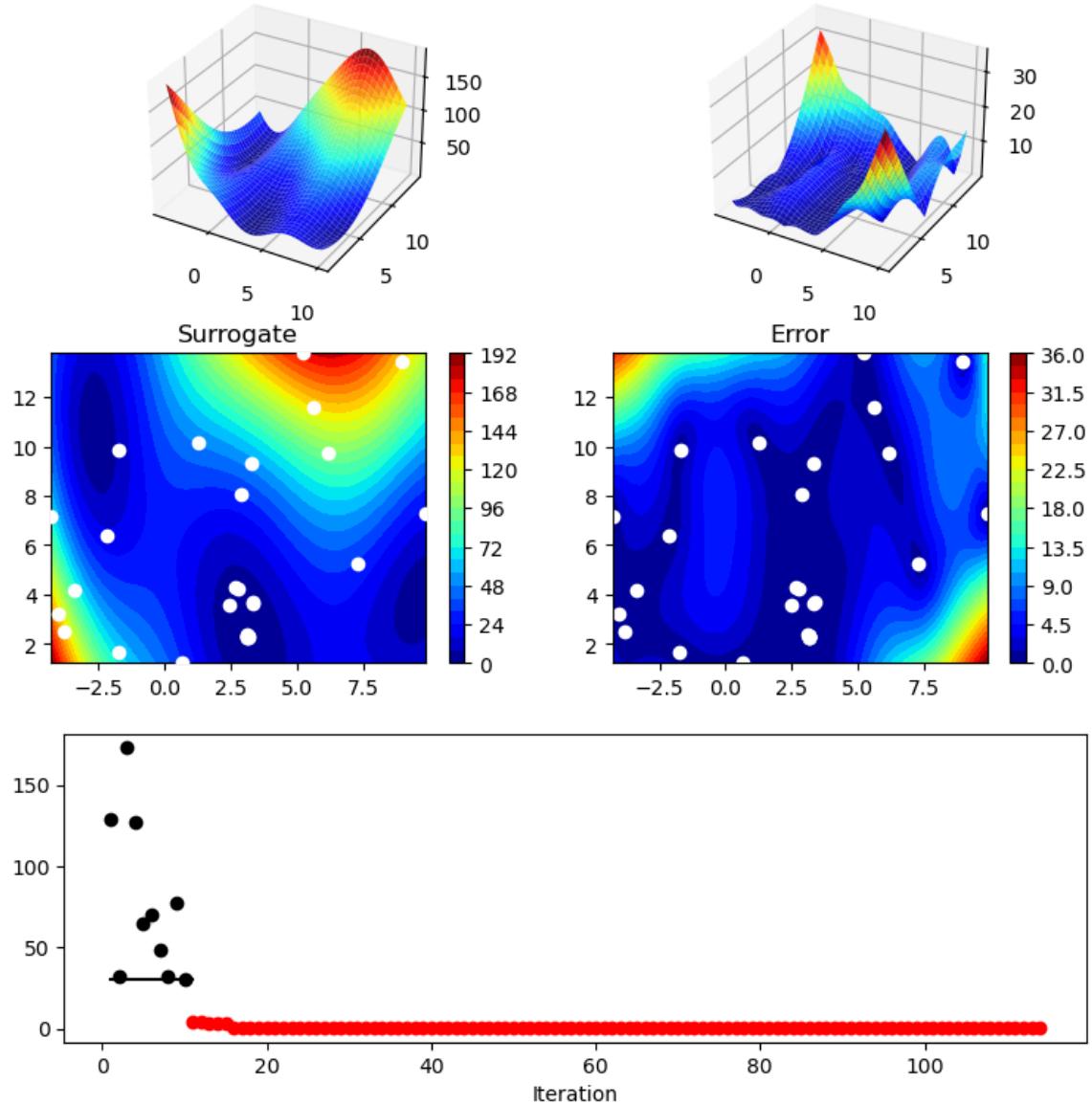
```

{'CHECKPOINT_PATH': 'runs/saved_models/',
'DATASET_PATH': 'data/',
'PREFIX': '003',
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,
'core_model_name': None,
'counter': 114,
'data': None,
'data_dir': './data',

```

```
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': inf,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([-5,  0]),
'max_surrogate_points': 30,
'max_time': 1,
'metric_params': {},
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/003_c02-b04_2024-05-21_13-23-1
8',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x154d40ec5a10>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([10, 15]),
'var_name': None,
'var_type': ['num'],
```

```
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 0.3979252227079453
x0: 3.141966133055075
x1: 2.2808075510609758
```



Isotropic and Anisotropic with time = 0.4 min

```
In [1]: import numpy as np
from math import inf
from spotPython.fun.objectivefunctions import analytical
from spotPython.spot import spot
from spotPython.utils.init import fun_control_init, surrogate_control_init
from math import inf
#Isotropic Branin Function
PREFIX="003"
fun = analytical().fun_branin
fun_control = fun_control_init(PREFIX=PREFIX,
                               fun_evals=inf,
                               max_time=0.4,
                               lower = np.array([-5, 0]),
```

```
upper = np.array([10, 15]))  
surrogate_control=surrogate_control_init(n_theta=1)  
spot_2 = spot.Spot(fun=fun,  
                    fun_control=fun_control,  
                    surrogate_control=surrogate_control)  
  
spot_2.run()  
spot_2.print_results()  
spot_2.plot_progress()  
spot_2.surrogate.plot()  
spot_2.surrogate.theta
```

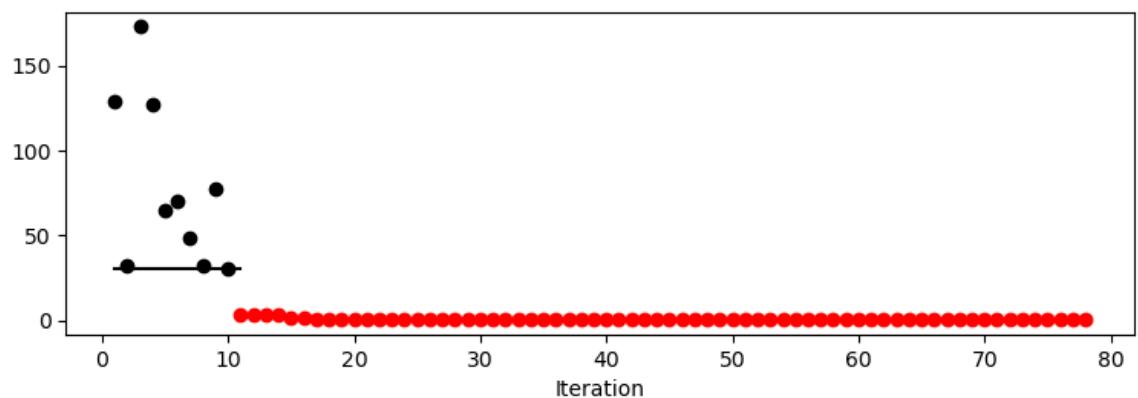
```
[rank: 0] Seed set to 123  
[rank: 0] Seed set to 123
```

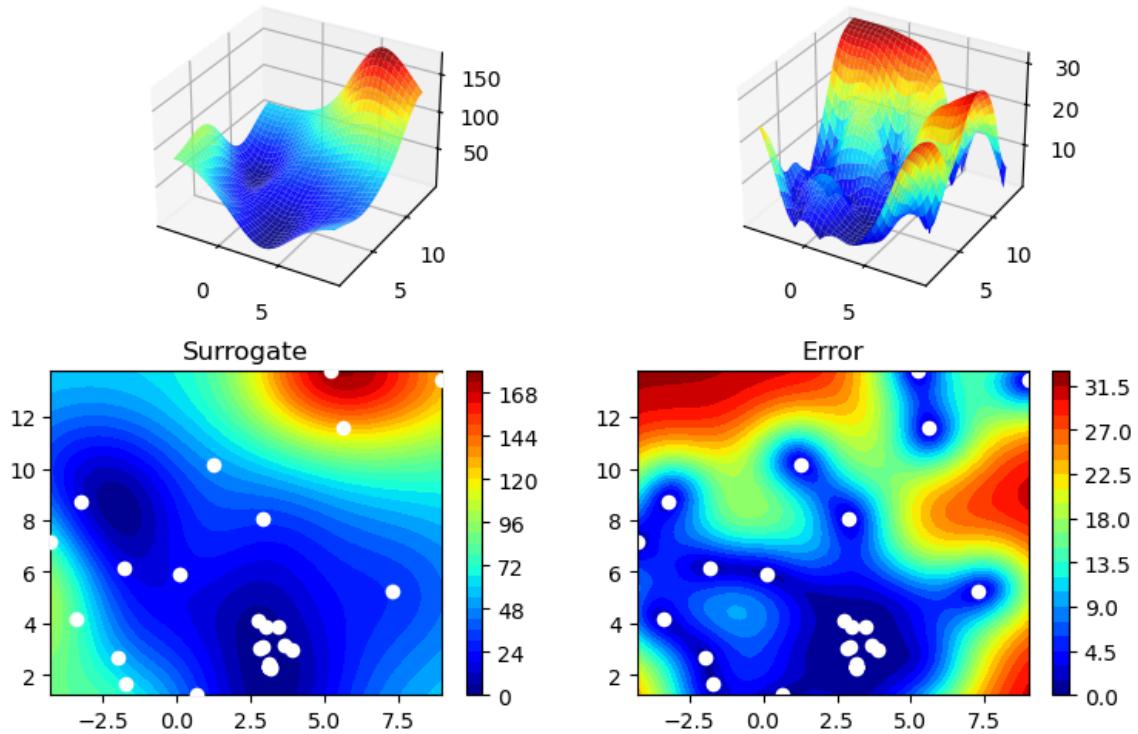
```
Created spot_tensorboard_path: runs/spot_logs/003_c03-b01_2024-05-27_16-10-41 for SummaryWriter()
spotPython tuning: 3.146896546946783 [-----] 0.76%
spotPython tuning: 3.146896546946783 [-----] 1.54%
spotPython tuning: 3.146896546946783 [-----] 2.43%
spotPython tuning: 3.146896546946783 [-----] 3.29%
spotPython tuning: 1.1487866236306008 [-----] 4.22%
spotPython tuning: 1.0240824742350494 [#-----] 5.14%
spotPython tuning: 0.4211249770550065 [#-----] 6.19%
spotPython tuning: 0.40198655236716263 [#-----] 7.34%
spotPython tuning: 0.3991783652663141 [#-----] 8.77%
spotPython tuning: 0.3991783652663141 [#-----] 9.99%
spotPython tuning: 0.3991783652663141 [#-----] 11.08%
spotPython tuning: 0.3991783652663141 [#-----] 12.14%
spotPython tuning: 0.3991783652663141 [#-----] 13.32%
spotPython tuning: 0.3991783652663141 [#-----] 14.52%
spotPython tuning: 0.3991783652663141 [##-----] 15.96%
spotPython tuning: 0.39883116428530485 [##-----] 17.65%
spotPython tuning: 0.39883116428530485 [##-----] 19.39%
spotPython tuning: 0.39830236020791254 [##-----] 20.96%
spotPython tuning: 0.3981492965243003 [##-----] 22.75%
spotPython tuning: 0.3981492965243003 [##-----] 24.52%
spotPython tuning: 0.3981492965243003 [###-----] 26.29%
spotPython tuning: 0.3981492965243003 [###-----] 27.80%
spotPython tuning: 0.3981492965243003 [###-----] 29.16%
spotPython tuning: 0.3981492965243003 [###-----] 30.84%
spotPython tuning: 0.3981492965243003 [###-----] 32.58%
spotPython tuning: 0.3981492965243003 [###-----] 34.14%
spotPython tuning: 0.3981492965243003 [####-----] 35.74%
spotPython tuning: 0.3981492965243003 [####-----] 37.24%
spotPython tuning: 0.3981492965243003 [####-----] 39.01%
spotPython tuning: 0.3981492965243003 [####-----] 40.73%
spotPython tuning: 0.3981492965243003 [####-----] 42.56%
spotPython tuning: 0.3981492965243003 [####-----] 44.47%
spotPython tuning: 0.3981492965243003 [####-----] 46.01%
spotPython tuning: 0.3981492965243003 [####-----] 47.69%
spotPython tuning: 0.3981492965243003 [####-----] 49.23%
spotPython tuning: 0.3981492965243003 [####-----] 50.89%
spotPython tuning: 0.3981492965243003 [####-----] 52.42%
spotPython tuning: 0.3981492965243003 [####-----] 53.93%
spotPython tuning: 0.3981492965243003 [####-----] 55.36%
spotPython tuning: 0.3981492965243003 [####-----] 56.81%
spotPython tuning: 0.3981492965243003 [####-----] 58.28%
spotPython tuning: 0.3981492965243003 [####-----] 60.50%
spotPython tuning: 0.3981492965243003 [####-----] 61.89%
spotPython tuning: 0.3981492965243003 [####-----] 63.23%
spotPython tuning: 0.3981492965243003 [####-----] 64.77%
spotPython tuning: 0.3981492965243003 [#####-----] 66.22%
spotPython tuning: 0.3981492965243003 [#####-----] 67.97%
spotPython tuning: 0.3981492965243003 [#####-----] 69.75%
spotPython tuning: 0.3981492965243003 [#####-----] 71.39%
spotPython tuning: 0.3981367922869037 [#####-----] 72.88%
spotPython tuning: 0.3981367922869037 [#####-----] 74.62%
spotPython tuning: 0.3981272085911698 [#####-----] 76.12%
spotPython tuning: 0.3981260363674277 [#####-----] 77.67%
spotPython tuning: 0.39812562159481324 [#####-----] 79.14%
spotPython tuning: 0.3981243574293405 [#####-----] 80.71%
spotPython tuning: 0.3981243574293405 [#####-----] 82.28%
spotPython tuning: 0.39812127569033073 [#####-----] 83.81%
spotPython tuning: 0.39812127569033073 [#####-----] 85.40%
```

```
spotPython tuning: 0.39812127569033073 [#####] 86.99%
spotPython tuning: 0.39812127569033073 [#####] 88.47%
spotPython tuning: 0.39812127569033073 [#####] 89.92%
spotPython tuning: 0.39812127569033073 [#####] 91.37%
spotPython tuning: 0.39812127569033073 [#####] 92.71%
spotPython tuning: 0.39812127569033073 [#####] 94.42%
spotPython tuning: 0.39812127569033073 [#####] 96.18%
spotPython tuning: 0.39812127569033073 [#####] 97.78%
spotPython tuning: 0.39801594419739494 [#####] 99.20%
spotPython tuning: 0.39801503764417845 [#####] 100.00% Done...

{'CHECKPOINT_PATH': 'runs/saved_models/',
 'DATASET_PATH': 'data/',
 'PREFIX': '003',
 'RESULTS_PATH': 'results/',
 'TENSORBOARD_PATH': 'runs/',
 '_L_in': None,
 '_L_out': None,
 '_torchmetric': None,
 'accelerator': 'auto',
 'converters': None,
 'core_model': None,
 'core_model_name': None,
 'counter': 78,
 'data': None,
 'data_dir': './data',
 'data_module': None,
 'data_set': None,
 'data_set_name': None,
 'db_dict_name': None,
 'design': None,
 'device': None,
 'devices': 1,
 'enable_progress_bar': False,
 'eval': None,
 'fun_evals': inf,
 'fun_repeats': 1,
 'horizon': None,
 'infill_criterion': 'y',
 'k_folds': 3,
 'log_graph': False,
 'log_level': 50,
 'loss_function': None,
 'lower': array([-5, 0]),
 'max_surrogate_points': 30,
 'max_time': 0.4,
 'metric_params': {},
 'metric_river': None,
 'metric_sklearn': None,
 'metric_sklearn_name': None,
 'metric_torch': None,
 'model_dict': {},
 'n_points': 1,
 'n_samples': None,
 'n_total': None,
 'noise': False,
 'num_workers': 0,
 'ocba_delta': 0,
 'oml_grace_period': None,
 'optimizer': None,
```

```
'path': None,  
'prep_model': None,  
'prep_model_name': None,  
'progress_file': None,  
'save_model': False,  
'seed': 123,  
'show_batch_interval': 1000000,  
'show_models': False,  
'show_progress': True,  
'shuffle': None,  
'sigma': 0.0,  
'spot_tensorboard_path': 'runs/spot_logs/003_c03-b01_2024-05-27_16-10-4  
1',  
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at  
0x148a702295d0>,  
'target_column': None,  
'target_type': None,  
'task': None,  
'test': None,  
'test_seed': 1234,  
'test_size': 0.4,  
'tolerance_x': 0,  
'train': None,  
'upper': array([10, 15]),  
'var_name': None,  
'var_type': ['num'],  
'verbosity': 0,  
'weight_coeff': 0.0,  
'weights': 1.0,  
'weights_entry': None}  
min y: 0.39801503764417845  
x0: 3.1404168943160635  
x1: 2.2869190545593483
```





Out[1]: array([-0.99852039])

```
In [2]: #Anisotropic Branin Function
surrogate_control = surrogate_control_init(n_theta=2)
spot_2_anisotropic = spot.Spot(fun=fun,
                               fun_control=fun_control,
                               surrogate_control=surrogate_control)
spot_2_anisotropic.run()
spot_2_anisotropic.print_results()
spot_2_anisotropic.surrogate.plot()
spot_2_anisotropic.surrogate.theta
spot_2_anisotropic.plot_progress()
```

```

spotPython tuning: 3.800462807494956 [-----] 1.00%
spotPython tuning: 3.800462807494956 [-----] 2.12%
spotPython tuning: 3.1589357387738524 [-----] 3.27%
spotPython tuning: 3.134400412616152 [-----] 4.41%
spotPython tuning: 2.9066716414049374 [#-----] 5.74%
spotPython tuning: 0.4143165308821022 [#-----] 7.29%
spotPython tuning: 0.4041224802204155 [#-----] 8.77%
spotPython tuning: 0.3987714674290057 [#-----] 10.27%
spotPython tuning: 0.3987714674290057 [#-----] 11.75%
spotPython tuning: 0.3987714674290057 [#-----] 13.33%
spotPython tuning: 0.3987714674290057 [#-----] 14.83%
spotPython tuning: 0.3987714674290057 [##-----] 16.67%
spotPython tuning: 0.3987714674290057 [##-----] 18.26%
spotPython tuning: 0.3987714674290057 [##-----] 20.27%
spotPython tuning: 0.3987714674290057 [##-----] 22.18%
spotPython tuning: 0.3984374184973323 [##-----] 24.08%
spotPython tuning: 0.39794164248837127 [###-----] 26.59%
spotPython tuning: 0.39794164248837127 [###-----] 29.00%
spotPython tuning: 0.39794164248837127 [###-----] 31.22%
spotPython tuning: 0.39794164248837127 [###-----] 33.63%
spotPython tuning: 0.39794164248837127 [####-----] 36.17%
spotPython tuning: 0.39794164248837127 [####-----] 38.60%
spotPython tuning: 0.39794164248837127 [####-----] 41.02%
spotPython tuning: 0.39794164248837127 [####-----] 43.51%
spotPython tuning: 0.39794164248837127 [####-----] 45.92%
spotPython tuning: 0.39794164248837127 [####-----] 48.28%
spotPython tuning: 0.39794164248837127 [####-----] 50.60%
spotPython tuning: 0.39794164248837127 [####-----] 52.60%
spotPython tuning: 0.39794164248837127 [#####-----] 55.11%
spotPython tuning: 0.39794164248837127 [#####-----] 57.38%
spotPython tuning: 0.39794164248837127 [#####-----] 59.86%
spotPython tuning: 0.39794164248837127 [#####-----] 62.26%
spotPython tuning: 0.39794164248837127 [#####-----] 64.48%
spotPython tuning: 0.39794164248837127 [#####-----] 66.97%
spotPython tuning: 0.39794164248837127 [#####-----] 69.31%
spotPython tuning: 0.39794164248837127 [#####-----] 71.74%
spotPython tuning: 0.39794164248837127 [#####-----] 74.23%
spotPython tuning: 0.39794164248837127 [#####-----] 76.95%
spotPython tuning: 0.39794164248837127 [#####-----] 79.33%
spotPython tuning: 0.39794164248837127 [#####-----] 81.64%
spotPython tuning: 0.39794164248837127 [#####-----] 84.04%
spotPython tuning: 0.39794164248837127 [#####-----] 86.52%
spotPython tuning: 0.39794164248837127 [#####-----] 88.97%
spotPython tuning: 0.39794164248837127 [#####-----] 91.21%
spotPython tuning: 0.39794164248837127 [#####-----] 93.53%
spotPython tuning: 0.39794164248837127 [#####-----] 95.80%
spotPython tuning: 0.39794164248837127 [#####-----] 98.01%
spotPython tuning: 0.39794164248837127 [#####-----] 100.00% Done...

```

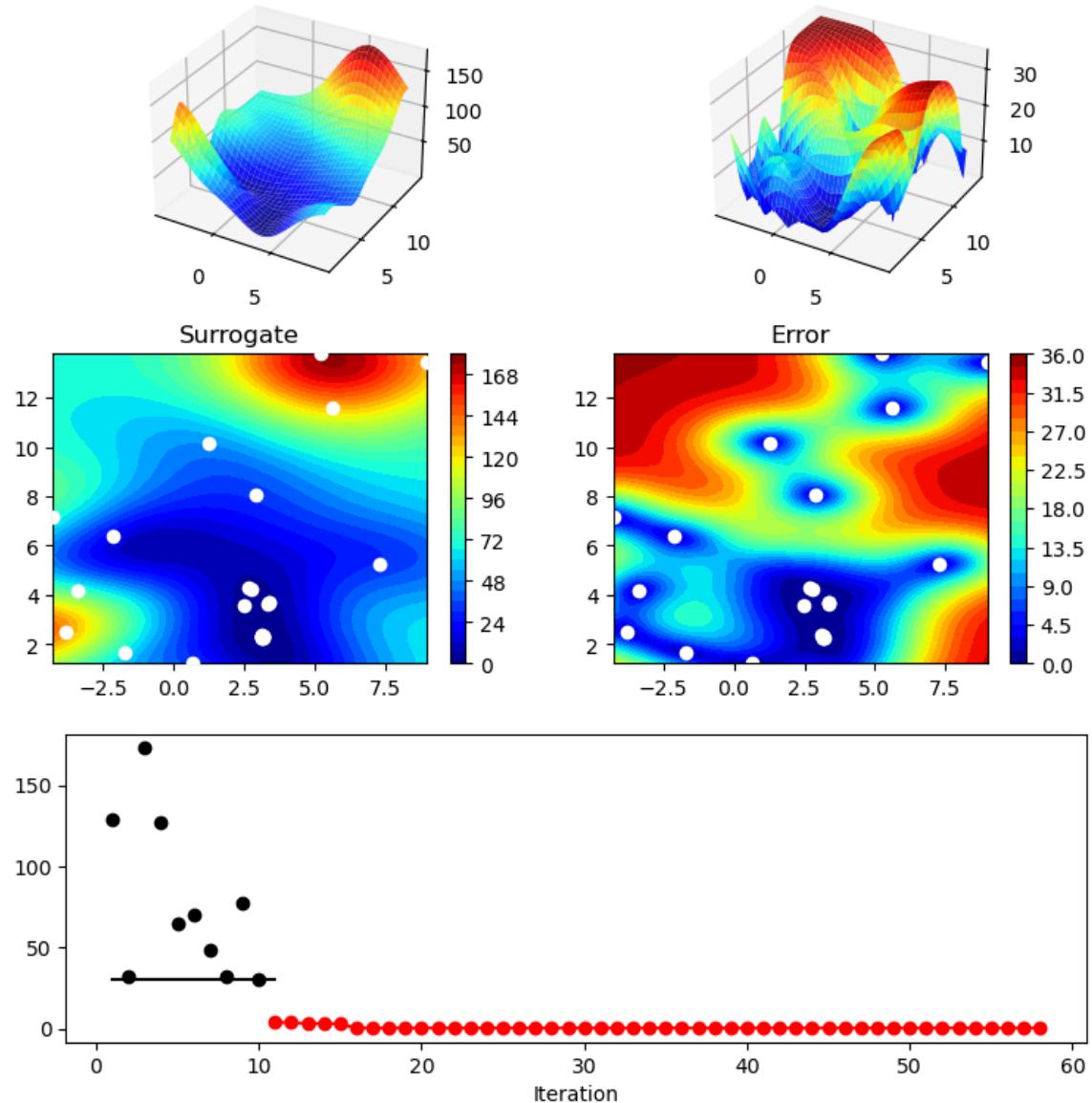
```

{'CHECKPOINT_PATH': 'runs/saved_models/',
'DATASET_PATH': 'data/',
'PREFIX': '003',
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,

```

```
'core_model_name': None,
'counter': 58,
'data': None,
'data_dir': './data',
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': inf,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([-5,  0]),
'max_surrogate_points': 30,
'max_time': 0.4,
'metric_params': {},
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/003_c03-b01_2024-05-27_16-10-4
1',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x148a702295d0>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
```

```
'train': None,
'upper': array([10, 15]),
'var_name': None,
'var_type': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 0.39794164248837127
x0: 3.1426690711947476
x1: 2.2811408026436806
```



Result of Branin Function

Function	n_theta	max_time(min)	Number of Evaluations	Number of Iterations	Objective function Value
Branin Function	1	1	inf	194	0.39790454173192735
Branin Function	2	1	inf	114	0.3979252227079453
Branin	1	0.4	inf	78	0.39801503764417845

Function	n_theta	max_time(min)	Number of Evaluations	Number of Iterations	Objective function Value
Function					
Branin Function	2	0.4	inf	58	0.39794164248837127

Increasing `n_theta` from 1 to 2 generally improves optimization efficiency by reducing the number of iterations required, as seen with iterations dropping from 194 to 114 for `max_time` set to 1 minute and from 78 to 58 for `max_time` set to 0.4 minutes. However, reducing `max_time` from 1 minute to 0.4 minutes results in a slight increase in the objective function value, indicating a trade-off between optimization performance and convergence speed.

2. The Two-dimensional Sin-Cos Function `fun_sin_cos`

- Describe the function.

* Sinusoidal function.

Args:

X (array):

input

fun_control (dict):

dict with entries `sigma` (noise level) and `seed` (random seed).

Returns:

(np.ndarray): A 1D numpy array with shape (n,) containing the calculated values.

* """Sinusoidal function.

Args:

X (array):

input

fun_control (dict):

dict with entries `sigma` (noise level) and `seed` (random seed).

Returns:

(np.ndarray): A 1D numpy array with shape (n,) containing the calculated values.

```
 `` `` ``
```

- * The input dimension is `2`. The search range is $-2\pi \leq x_1 \leq 2\pi$ and $-2\pi \leq x_2 \leq 2\pi$.
- Compare the results from `spotPython` run a) with isotropic and b) anisotropic surrogate models.
- Modify the termination criterion (`max_time` instead of `fun_evals`) as described for `fun_branin`

Isotropic and Anisotropic with time = 1 min

```
In [4]: #Isotropic Sin-Cos Function
PREFIX="003"
fun = analytical().fun_sin_cos
a = 2* np.pi
#print(a)
fun_control = fun_control_init(PREFIX=PREFIX,
                                fun_evals=inf,
                                max_time=1,
                                lower = np.array([-a,-a]),
                                upper = np.array([a,a]))
surrogate_control=surrogate_control_init(n_theta=1)
spot_2 = spot.Spot(fun=fun,
                    fun_control=fun_control,
                    surrogate_control=surrogate_control)

spot_2.run()
spot_2.print_results()
spot_2.plot_progress()
spot_2.surrogate.plot()
spot_2.surrogate.theta
```

```
[rank: 0] Seed set to 123
```

```
Created spot_tensorboard_path: runs/spot_logs/003_c02-b04_2024-05-21_13-27-49 for SummaryWriter()
spotPython tuning: -2.032916688789623 [-----] 0.38%
spotPython tuning: -2.388464187850052 [-----] 0.79%
spotPython tuning: -2.4732239160328326 [-----] 1.13%
spotPython tuning: -2.4732239160328326 [-----] 1.52%
spotPython tuning: -2.4892152901553843 [-----] 1.84%
spotPython tuning: -2.4991257784759258 [-----] 2.19%
spotPython tuning: -2.4991257784759258 [-----] 2.54%
spotPython tuning: -2.4999988700971447 [-----] 2.88%
spotPython tuning: -2.499999950865807 [-----] 3.25%
spotPython tuning: -2.4999999509691717 [-----] 3.58%
spotPython tuning: -2.499999956610016 [-----] 3.89%
spotPython tuning: -2.499999956610016 [-----] 4.21%
spotPython tuning: -2.4999999618068043 [-----] 4.55%
spotPython tuning: -2.4999999618068043 [-----] 4.87%
spotPython tuning: -2.4999999675987525 [#-----] 5.26%
spotPython tuning: -2.4999999675987525 [#-----] 5.61%
spotPython tuning: -2.4999999675987525 [#-----] 6.12%
spotPython tuning: -2.4999999675987525 [#-----] 6.58%
spotPython tuning: -2.4999999675987525 [#-----] 7.21%
spotPython tuning: -2.4999999675987525 [#-----] 7.75%
spotPython tuning: -2.4999999675987525 [#-----] 8.38%
spotPython tuning: -2.4999999675987525 [#-----] 8.92%
spotPython tuning: -2.4999999675987525 [#-----] 9.55%
spotPython tuning: -2.4999999675987525 [#-----] 10.10%
spotPython tuning: -2.4999999675987525 [#-----] 10.75%
spotPython tuning: -2.4999999675987525 [#-----] 11.36%
spotPython tuning: -2.4999999675987525 [#-----] 12.07%
spotPython tuning: -2.4999999675987525 [#-----] 12.68%
spotPython tuning: -2.4999999675987525 [#-----] 13.32%
spotPython tuning: -2.4999999675987525 [#-----] 13.85%
spotPython tuning: -2.4999999675987525 [#-----] 14.50%
spotPython tuning: -2.4999999675987525 [##-----] 15.04%
spotPython tuning: -2.4999999675987525 [##-----] 15.82%
spotPython tuning: -2.4999999675987525 [##-----] 16.39%
spotPython tuning: -2.4999999675987525 [##-----] 17.03%
spotPython tuning: -2.4999999675987525 [##-----] 17.59%
spotPython tuning: -2.4999999675987525 [##-----] 18.14%
spotPython tuning: -2.4999999675987525 [##-----] 18.69%
spotPython tuning: -2.4999999675987525 [##-----] 19.22%
spotPython tuning: -2.4999999675987525 [##-----] 19.86%
spotPython tuning: -2.4999999675987525 [##-----] 20.41%
spotPython tuning: -2.4999999675987525 [##-----] 20.91%
spotPython tuning: -2.4999999675987525 [##-----] 21.46%
spotPython tuning: -2.4999999675987525 [##-----] 22.00%
spotPython tuning: -2.4999999675987525 [##-----] 22.56%
spotPython tuning: -2.4999999675987525 [##-----] 23.14%
spotPython tuning: -2.4999999675987525 [##-----] 23.74%
spotPython tuning: -2.4999999675987525 [##-----] 24.28%
spotPython tuning: -2.4999999675987525 [##-----] 24.96%
spotPython tuning: -2.4999999675987525 [###-----] 25.62%
spotPython tuning: -2.4999999675987525 [###-----] 26.14%
spotPython tuning: -2.4999999675987525 [###-----] 26.67%
spotPython tuning: -2.4999999675987525 [###-----] 27.21%
spotPython tuning: -2.4999999675987525 [###-----] 27.73%
spotPython tuning: -2.4999999675987525 [###-----] 28.25%
spotPython tuning: -2.4999999675987525 [###-----] 28.78%
spotPython tuning: -2.4999999675987525 [###-----] 29.32%
spotPython tuning: -2.4999999675987525 [###-----] 29.95%
```

```
spotPython tuning: -2.4999999675987525 [###-----] 30.61%
spotPython tuning: -2.4999999675987525 [###-----] 31.18%
spotPython tuning: -2.4999999675987525 [###-----] 31.80%
spotPython tuning: -2.4999999675987525 [###-----] 32.52%
spotPython tuning: -2.4999999675987525 [###-----] 33.30%
spotPython tuning: -2.4999999675987525 [###-----] 33.92%
spotPython tuning: -2.4999999675987525 [###-----] 34.51%
spotPython tuning: -2.4999999675987525 [####-----] 35.15%
spotPython tuning: -2.4999999675987525 [####-----] 35.62%
spotPython tuning: -2.4999999675987525 [####-----] 36.33%
spotPython tuning: -2.4999999675987525 [####-----] 37.03%
spotPython tuning: -2.4999999675987525 [####-----] 37.54%
spotPython tuning: -2.4999999675987525 [####-----] 38.20%
spotPython tuning: -2.4999999675987525 [####-----] 38.74%
spotPython tuning: -2.4999999675987525 [####-----] 39.43%
spotPython tuning: -2.4999999675987525 [####-----] 39.93%
spotPython tuning: -2.4999999675987525 [####-----] 40.36%
spotPython tuning: -2.4999999675987525 [####-----] 40.88%
spotPython tuning: -2.4999999675987525 [####-----] 41.40%
spotPython tuning: -2.4999999675987525 [####-----] 41.81%
spotPython tuning: -2.4999999675987525 [####-----] 42.21%
spotPython tuning: -2.4999999675987525 [####-----] 42.62%
spotPython tuning: -2.4999999675987525 [####-----] 43.03%
spotPython tuning: -2.4999999675987525 [####-----] 43.43%
spotPython tuning: -2.4999999675987525 [####-----] 43.82%
spotPython tuning: -2.4999999675987525 [####-----] 44.21%
spotPython tuning: -2.4999999675987525 [####-----] 44.59%
spotPython tuning: -2.4999999675987525 [#####-----] 45.03%
spotPython tuning: -2.4999999675987525 [#####-----] 45.52%
spotPython tuning: -2.4999999675987525 [#####-----] 45.97%
spotPython tuning: -2.4999999675987525 [#####-----] 46.40%
spotPython tuning: -2.4999999675987525 [#####-----] 46.82%
spotPython tuning: -2.4999999675987525 [#####-----] 47.23%
spotPython tuning: -2.4999999675987525 [#####-----] 47.63%
spotPython tuning: -2.4999999675987525 [#####-----] 48.03%
spotPython tuning: -2.4999999675987525 [#####-----] 48.51%
spotPython tuning: -2.4999999675987525 [#####-----] 48.91%
spotPython tuning: -2.4999999675987525 [#####-----] 49.42%
spotPython tuning: -2.4999999675987525 [#####-----] 49.77%
spotPython tuning: -2.4999999675987525 [#####-----] 50.31%
spotPython tuning: -2.4999999675987525 [#####-----] 50.72%
spotPython tuning: -2.4999999675987525 [#####-----] 51.16%
spotPython tuning: -2.4999999675987525 [#####-----] 51.61%
spotPython tuning: -2.4999999675987525 [#####-----] 51.98%
spotPython tuning: -2.4999999675987525 [#####-----] 52.38%
spotPython tuning: -2.4999999675987525 [#####-----] 52.78%
spotPython tuning: -2.4999999675987525 [#####-----] 53.19%
spotPython tuning: -2.4999999675987525 [#####-----] 53.61%
spotPython tuning: -2.4999999675987525 [#####-----] 54.09%
spotPython tuning: -2.4999999675987525 [#####-----] 54.52%
spotPython tuning: -2.4999999675987525 [#####-----] 54.94%
spotPython tuning: -2.4999999675987525 [#####-----] 55.35%
spotPython tuning: -2.4999999675987525 [#####-----] 55.74%
spotPython tuning: -2.4999999675987525 [#####-----] 56.07%
spotPython tuning: -2.4999999675987525 [#####-----] 56.42%
spotPython tuning: -2.4999999675987525 [#####-----] 56.80%
spotPython tuning: -2.4999999675987525 [#####-----] 57.13%
spotPython tuning: -2.4999999675987525 [#####-----] 57.44%
spotPython tuning: -2.4999999675987525 [#####-----] 57.89%
spotPython tuning: -2.4999999675987525 [#####-----] 58.34%
```

```
spotPython tuning: -2.4999999675987525 [#####----] 58.78%
spotPython tuning: -2.4999999675987525 [#####----] 59.26%
spotPython tuning: -2.4999999675987525 [#####----] 59.70%
spotPython tuning: -2.4999999675987525 [#####----] 60.07%
spotPython tuning: -2.4999999675987525 [#####----] 60.51%
spotPython tuning: -2.4999999675987525 [#####----] 60.91%
spotPython tuning: -2.4999999675987525 [#####----] 61.36%
spotPython tuning: -2.4999999675987525 [#####----] 61.77%
spotPython tuning: -2.4999999675987525 [#####----] 62.15%
spotPython tuning: -2.4999999675987525 [#####----] 62.57%
spotPython tuning: -2.4999999675987525 [#####----] 62.94%
spotPython tuning: -2.4999999675987525 [#####----] 63.31%
spotPython tuning: -2.4999999675987525 [#####----] 63.97%
spotPython tuning: -2.4999999675987525 [#####----] 64.38%
spotPython tuning: -2.4999999675987525 [#####----] 64.79%
spotPython tuning: -2.4999999675987525 [#####----] 65.19%
spotPython tuning: -2.4999999675987525 [#####----] 65.60%
spotPython tuning: -2.4999999675987525 [#####----] 66.00%
spotPython tuning: -2.4999999675987525 [#####----] 66.38%
spotPython tuning: -2.4999999675987525 [#####----] 66.83%
spotPython tuning: -2.4999999675987525 [#####----] 67.28%
spotPython tuning: -2.4999999675987525 [#####----] 67.67%
spotPython tuning: -2.4999999675987525 [#####----] 68.07%
spotPython tuning: -2.4999999675987525 [#####----] 68.44%
spotPython tuning: -2.4999999675987525 [#####----] 68.86%
spotPython tuning: -2.4999999675987525 [#####----] 69.27%
spotPython tuning: -2.4999999675987525 [#####----] 69.71%
spotPython tuning: -2.4999999675987525 [#####----] 70.14%
spotPython tuning: -2.4999999675987525 [#####----] 70.58%
spotPython tuning: -2.4999999675987525 [#####----] 71.02%
spotPython tuning: -2.4999999675987525 [#####----] 71.46%
spotPython tuning: -2.4999999675987525 [#####----] 71.92%
spotPython tuning: -2.4999999675987525 [#####----] 72.32%
spotPython tuning: -2.4999999675987525 [#####----] 72.76%
spotPython tuning: -2.4999999675987525 [#####----] 73.18%
spotPython tuning: -2.4999999675987525 [#####----] 73.59%
spotPython tuning: -2.4999999675987525 [#####----] 74.01%
spotPython tuning: -2.4999999675987525 [#####----] 74.44%
spotPython tuning: -2.4999999675987525 [#####----] 74.80%
spotPython tuning: -2.4999999675987525 [#####----] 75.24%
spotPython tuning: -2.4999999675987525 [#####----] 75.66%
spotPython tuning: -2.4999999675987525 [#####----] 76.10%
spotPython tuning: -2.4999999675987525 [#####----] 76.54%
spotPython tuning: -2.4999999675987525 [#####----] 76.93%
spotPython tuning: -2.4999999675987525 [#####----] 77.32%
spotPython tuning: -2.4999999675987525 [#####----] 77.75%
spotPython tuning: -2.4999999675987525 [#####----] 78.21%
spotPython tuning: -2.4999999675987525 [#####----] 78.63%
spotPython tuning: -2.4999999675987525 [#####----] 79.04%
spotPython tuning: -2.4999999675987525 [#####----] 79.44%
spotPython tuning: -2.4999999675987525 [#####----] 79.90%
spotPython tuning: -2.4999999675987525 [#####----] 80.29%
spotPython tuning: -2.4999999675987525 [#####----] 80.78%
spotPython tuning: -2.4999999675987525 [#####----] 81.20%
spotPython tuning: -2.4999999675987525 [#####----] 81.59%
spotPython tuning: -2.4999999675987525 [#####----] 82.03%
spotPython tuning: -2.4999999675987525 [#####----] 82.40%
spotPython tuning: -2.4999999675987525 [#####----] 82.82%
spotPython tuning: -2.4999999675987525 [#####----] 83.21%
spotPython tuning: -2.4999999675987525 [#####----] 83.63%
```

```

spotPython tuning: -2.4999999675987525 [#####] 84.02%
spotPython tuning: -2.4999999675987525 [#####] 84.46%
spotPython tuning: -2.4999999675987525 [#####] 84.88%
spotPython tuning: -2.4999999675987525 [#####] 85.29%
spotPython tuning: -2.4999999675987525 [#####] 85.72%
spotPython tuning: -2.4999999675987525 [#####] 86.16%
spotPython tuning: -2.4999999675987525 [#####] 86.60%
spotPython tuning: -2.4999999675987525 [#####] 87.04%
spotPython tuning: -2.4999999675987525 [#####] 87.46%
spotPython tuning: -2.4999999675987525 [#####] 87.88%
spotPython tuning: -2.4999999675987525 [#####] 88.31%
spotPython tuning: -2.4999999675987525 [#####] 88.73%
spotPython tuning: -2.4999999675987525 [#####] 89.17%
spotPython tuning: -2.4999999675987525 [#####] 89.53%
spotPython tuning: -2.4999999675987525 [#####] 89.95%
spotPython tuning: -2.4999999675987525 [#####] 90.37%
spotPython tuning: -2.4999999675987525 [#####] 90.76%
spotPython tuning: -2.4999999675987525 [#####] 91.16%
spotPython tuning: -2.4999999675987525 [#####] 91.60%
spotPython tuning: -2.4999999675987525 [#####] 92.03%
spotPython tuning: -2.4999999675987525 [#####] 92.43%
spotPython tuning: -2.4999999675987525 [#####] 92.81%
spotPython tuning: -2.4999999675987525 [#####] 93.19%
spotPython tuning: -2.4999999675987525 [#####] 93.58%
spotPython tuning: -2.4999999675987525 [#####] 94.02%
spotPython tuning: -2.4999999675987525 [#####] 94.41%
spotPython tuning: -2.4999999675987525 [#####] 94.82%
spotPython tuning: -2.4999999675987525 [#####] 95.25%
spotPython tuning: -2.4999999675987525 [#####] 95.67%
spotPython tuning: -2.4999999675987525 [#####] 96.10%
spotPython tuning: -2.4999999675987525 [#####] 96.49%
spotPython tuning: -2.4999999675987525 [#####] 96.92%
spotPython tuning: -2.4999999675987525 [#####] 97.31%
spotPython tuning: -2.4999999675987525 [#####] 97.78%
spotPython tuning: -2.4999999675987525 [#####] 98.19%
spotPython tuning: -2.4999999675987525 [#####] 98.58%
spotPython tuning: -2.4999999675987525 [#####] 98.96%
spotPython tuning: -2.4999999675987525 [#####] 99.42%
spotPython tuning: -2.4999999675987525 [#####] 99.81%
spotPython tuning: -2.4999999675987525 [#####] 100.00% Done...

```

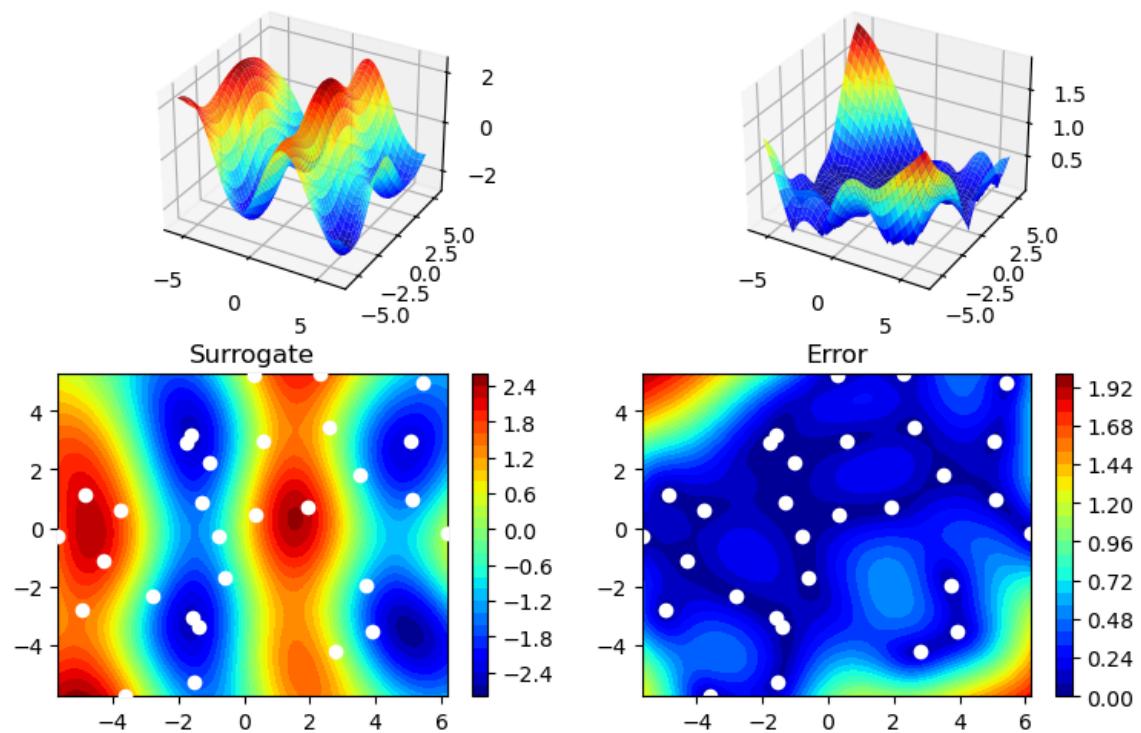
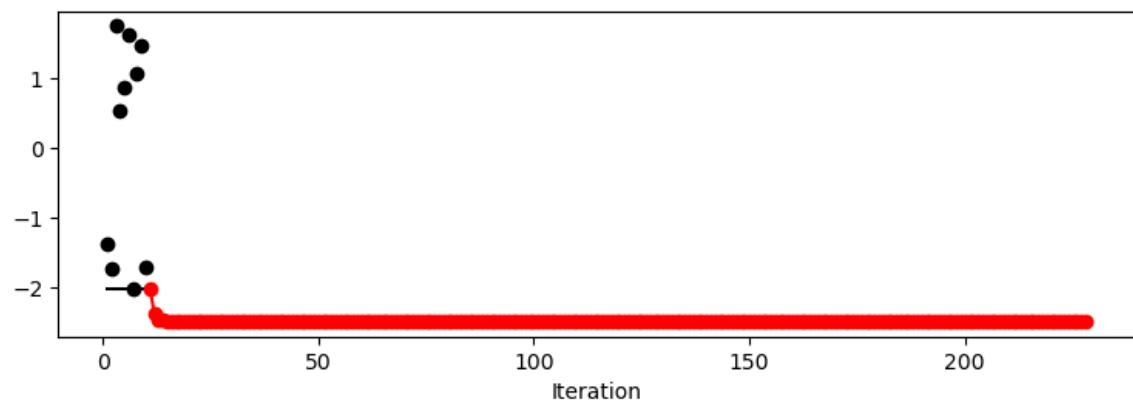
```

{'CHECKPOINT_PATH': 'runs/saved_models/',
'DATASET_PATH': 'data/',
'PREFIX': '003',
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,
'core_model_name': None,
'counter': 228,
'data': None,
'data_dir': './data',
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
}

```

```
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': inf,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([-6.28318531, -6.28318531]),
'max_surrogate_points': 30,
'max_time': 1,
'metric_params': {},
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/003_c02-b04_2024-05-21_13-27-4
9',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x154d38ae3d50>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([6.28318531, 6.28318531]),
'vername': None,
'vertype': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
```

```
min y: -2.4999999675987525
x0: -1.5706722620612161
x1: 3.1418534921552315
```



Out[4]: array([-1.01591397])

In [5]: #Anisotropic Sin-Cos Function
 surrogate_control = surrogate_control_init(n_theta=2)
 spot_2_anisotropic = spot.Spot(fun=fun,
 fun_control=fun_control,
 surrogate_control=surrogate_control)
 spot_2_anisotropic.run()
 spot_2_anisotropic.print_results()
 spot_2_anisotropic.surrogate.plot()
 spot_2_anisotropic.surrogate.theta

```
spotPython tuning: -2.2653112883046314 [-----] 0.54%
spotPython tuning: -2.2653112883046314 [-----] 1.11%
spotPython tuning: -2.2653112883046314 [-----] 1.62%
spotPython tuning: -2.454202160223451 [-----] 2.20%
spotPython tuning: -2.458521375229338 [-----] 2.71%
spotPython tuning: -2.482462560531697 [-----] 3.30%
spotPython tuning: -2.482925901019679 [-----] 3.76%
spotPython tuning: -2.4904698041359308 [-----] 4.31%
spotPython tuning: -2.4991596489166508 [-----] 4.80%
spotPython tuning: -2.499962076699469 [#-----] 5.41%
spotPython tuning: -2.4999776386398036 [#-----] 6.06%
spotPython tuning: -2.4999799363486916 [#-----] 6.60%
spotPython tuning: -2.4999804518304902 [#-----] 7.29%
spotPython tuning: -2.499981664757109 [#-----] 7.88%
spotPython tuning: -2.499981664757109 [#-----] 8.47%
spotPython tuning: -2.49998242826458 [#-----] 9.11%
spotPython tuning: -2.4999825326495873 [#-----] 9.86%
spotPython tuning: -2.4999928232338777 [#-----] 10.46%
spotPython tuning: -2.4999928232338777 [#-----] 11.07%
spotPython tuning: -2.4999995278811955 [#-----] 11.70%
spotPython tuning: -2.4999995278811955 [#-----] 12.55%
spotPython tuning: -2.49999998300882 [#-----] 13.26%
spotPython tuning: -2.49999998739451 [#-----] 13.89%
spotPython tuning: -2.4999999996677 [#-----] 14.62%
spotPython tuning: -2.4999999996677 [##-----] 15.51%
spotPython tuning: -2.4999999996677 [##-----] 16.35%
spotPython tuning: -2.4999999996677 [##-----] 17.23%
spotPython tuning: -2.4999999996677 [##-----] 18.11%
spotPython tuning: -2.4999999996677 [##-----] 18.83%
spotPython tuning: -2.4999999996677 [##-----] 19.67%
spotPython tuning: -2.4999999996677 [##-----] 20.57%
spotPython tuning: -2.4999999996677 [##-----] 21.55%
spotPython tuning: -2.4999999996677 [##-----] 22.42%
spotPython tuning: -2.4999999996677 [##-----] 23.33%
spotPython tuning: -2.4999999996677 [##-----] 24.11%
spotPython tuning: -2.4999999996677 [###-----] 25.09%
spotPython tuning: -2.4999999996677 [###-----] 26.00%
spotPython tuning: -2.4999999996677 [###-----] 26.91%
spotPython tuning: -2.4999999996677 [###-----] 27.71%
spotPython tuning: -2.4999999996677 [###-----] 28.50%
spotPython tuning: -2.4999999996677 [###-----] 29.22%
spotPython tuning: -2.4999999996677 [###-----] 30.02%
spotPython tuning: -2.4999999996677 [###-----] 30.74%
spotPython tuning: -2.4999999996677 [###-----] 31.52%
spotPython tuning: -2.4999999996677 [###-----] 32.23%
spotPython tuning: -2.4999999996677 [###-----] 32.95%
spotPython tuning: -2.4999999996677 [###-----] 33.73%
spotPython tuning: -2.4999999996677 [###-----] 34.53%
spotPython tuning: -2.4999999996677 [####-----] 35.41%
spotPython tuning: -2.4999999996677 [####-----] 36.14%
spotPython tuning: -2.4999999996677 [####-----] 36.99%
spotPython tuning: -2.4999999996677 [####-----] 37.81%
spotPython tuning: -2.4999999996677 [####-----] 38.54%
spotPython tuning: -2.4999999996677 [####-----] 39.28%
spotPython tuning: -2.4999999996677 [####-----] 40.01%
spotPython tuning: -2.4999999996677 [####-----] 40.89%
spotPython tuning: -2.4999999996677 [####-----] 41.71%
spotPython tuning: -2.4999999996677 [####-----] 42.49%
spotPython tuning: -2.4999999996677 [####-----] 43.25%
spotPython tuning: -2.4999999996677 [####-----] 44.00%
```

```
spotPython tuning: -2.49999999996677 [#####----] 44.85%
spotPython tuning: -2.49999999996677 [#####----] 45.62%
spotPython tuning: -2.49999999996677 [#####----] 46.39%
spotPython tuning: -2.49999999996677 [#####----] 47.15%
spotPython tuning: -2.49999999996677 [#####----] 47.78%
spotPython tuning: -2.49999999996677 [#####----] 48.42%
spotPython tuning: -2.49999999996677 [#####----] 49.15%
spotPython tuning: -2.49999999996677 [#####----] 49.78%
spotPython tuning: -2.49999999996677 [#####----] 50.46%
spotPython tuning: -2.49999999996677 [#####----] 51.10%
spotPython tuning: -2.49999999996677 [#####----] 51.73%
spotPython tuning: -2.49999999996677 [#####----] 52.35%
spotPython tuning: -2.49999999996677 [#####----] 52.97%
spotPython tuning: -2.49999999996677 [#####----] 53.66%
spotPython tuning: -2.49999999996677 [#####----] 54.26%
spotPython tuning: -2.49999999996677 [#####----] 54.87%
spotPython tuning: -2.49999999996677 [#####----] 55.52%
spotPython tuning: -2.49999999996677 [#####----] 56.15%
spotPython tuning: -2.49999999996677 [#####----] 56.71%
spotPython tuning: -2.49999999996677 [#####----] 57.34%
spotPython tuning: -2.49999999996677 [#####----] 57.99%
spotPython tuning: -2.49999999996677 [#####----] 58.71%
spotPython tuning: -2.49999999996677 [#####----] 59.39%
spotPython tuning: -2.49999999996677 [#####----] 59.97%
spotPython tuning: -2.49999999996677 [#####----] 60.54%
spotPython tuning: -2.49999999996677 [#####----] 61.10%
spotPython tuning: -2.49999999996677 [#####----] 61.67%
spotPython tuning: -2.49999999996677 [#####----] 62.27%
spotPython tuning: -2.49999999996677 [#####----] 62.88%
spotPython tuning: -2.49999999996677 [#####----] 63.47%
spotPython tuning: -2.49999999996677 [#####----] 64.10%
spotPython tuning: -2.49999999996677 [#####----] 64.67%
spotPython tuning: -2.49999999996677 [#####----] 65.31%
spotPython tuning: -2.49999999996677 [#####----] 66.08%
spotPython tuning: -2.49999999996677 [#####----] 66.82%
spotPython tuning: -2.49999999996677 [#####----] 67.44%
spotPython tuning: -2.49999999996677 [#####----] 68.07%
spotPython tuning: -2.49999999996677 [#####----] 68.75%
spotPython tuning: -2.49999999996677 [#####----] 69.40%
spotPython tuning: -2.49999999996677 [#####----] 70.03%
spotPython tuning: -2.49999999996677 [#####----] 70.74%
spotPython tuning: -2.49999999996677 [#####----] 71.41%
spotPython tuning: -2.49999999996677 [#####----] 72.03%
spotPython tuning: -2.49999999996677 [#####----] 72.77%
spotPython tuning: -2.49999999996677 [#####----] 73.41%
spotPython tuning: -2.49999999996677 [#####----] 74.15%
spotPython tuning: -2.49999999996677 [#####----] 74.85%
spotPython tuning: -2.49999999996677 [#####----] 75.61%
spotPython tuning: -2.49999999996677 [#####----] 76.22%
spotPython tuning: -2.49999999996677 [#####----] 76.79%
spotPython tuning: -2.49999999996677 [#####----] 77.36%
spotPython tuning: -2.49999999996677 [#####----] 78.07%
spotPython tuning: -2.49999999996677 [#####----] 78.79%
spotPython tuning: -2.49999999996677 [#####----] 79.53%
spotPython tuning: -2.49999999996677 [#####----] 80.16%
spotPython tuning: -2.49999999996677 [#####----] 80.83%
spotPython tuning: -2.49999999996677 [#####----] 81.53%
spotPython tuning: -2.49999999996677 [#####----] 82.19%
spotPython tuning: -2.49999999996677 [#####----] 82.92%
spotPython tuning: -2.49999999996677 [#####----] 83.53%
```

```

spotPython tuning: -2.49999999996677 [#####--] 84.18%
spotPython tuning: -2.49999999996677 [#####--] 84.79%
spotPython tuning: -2.49999999996677 [#####--] 85.39%
spotPython tuning: -2.49999999996677 [#####--] 86.00%
spotPython tuning: -2.49999999996677 [#####--] 86.63%
spotPython tuning: -2.49999999996677 [#####--] 87.29%
spotPython tuning: -2.49999999996677 [#####--] 87.94%
spotPython tuning: -2.49999999996677 [#####--] 88.56%
spotPython tuning: -2.49999999996677 [#####--] 89.18%
spotPython tuning: -2.49999999996677 [#####--] 89.82%
spotPython tuning: -2.49999999996677 [#####--] 90.47%
spotPython tuning: -2.49999999996677 [#####--] 91.26%
spotPython tuning: -2.49999999996677 [#####--] 92.03%
spotPython tuning: -2.49999999996677 [#####--] 92.72%
spotPython tuning: -2.49999999996677 [#####--] 93.35%
spotPython tuning: -2.49999999996677 [#####--] 93.96%
spotPython tuning: -2.49999999996677 [#####--] 94.65%
spotPython tuning: -2.49999999996677 [#####--] 95.34%
spotPython tuning: -2.49999999996677 [#####--] 96.00%
spotPython tuning: -2.49999999996677 [#####--] 96.67%
spotPython tuning: -2.49999999996677 [#####--] 97.37%
spotPython tuning: -2.49999999996677 [#####--] 98.06%
spotPython tuning: -2.49999999996677 [#####--] 98.81%
spotPython tuning: -2.49999999996677 [#####--] 99.44%
spotPython tuning: -2.49999999996677 [#####--] 100.00% Done...

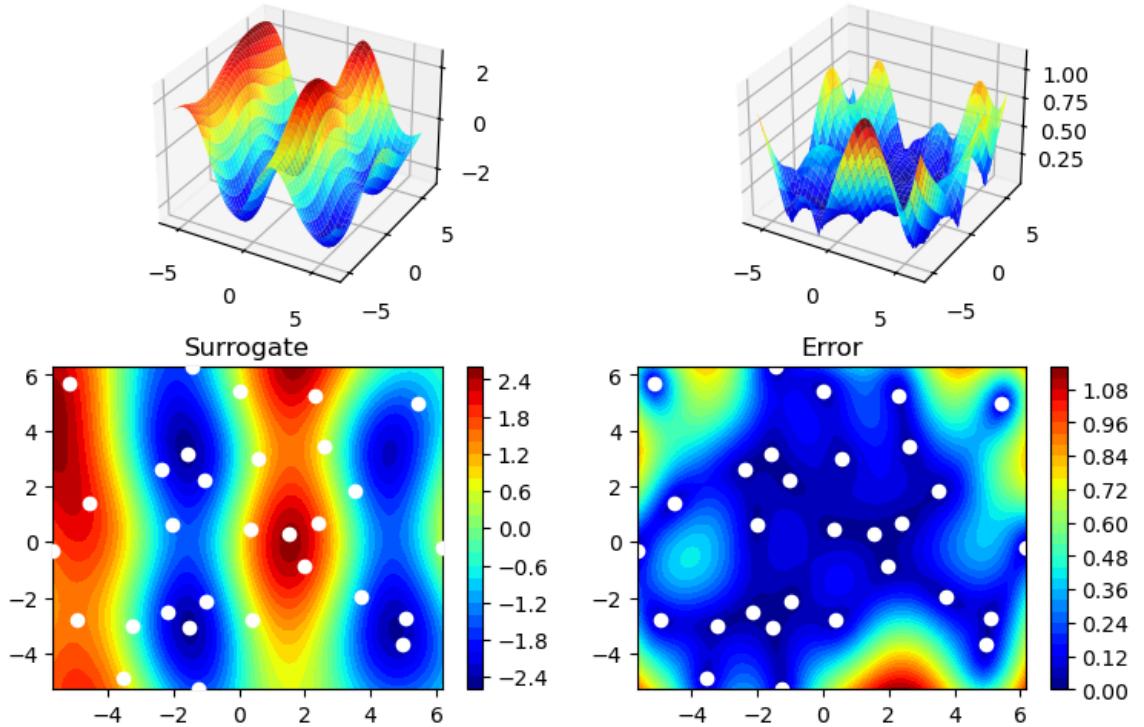
```

```

{'CHECKPOINT_PATH': 'runs/saved_models/',
'DATASET_PATH': 'data/',
'PREFIX': '003',
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,
'core_model_name': None,
'counter': 155,
'data': None,
'data_dir': './data',
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': inf,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([-6.28318531, -6.28318531]),
'max_surrogate_points': 30,
}

```

```
'max_time': 1,
'metric_params': {},
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/003_c02-b04_2024-05-21_13-27-4
9',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x154d38ae3d50>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([6.28318531, 6.28318531]),
'vername': None,
'vertype': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: -2.4999999996677
x0: -1.5707818548046566
x1: 3.14161482215979
```



Out[5]: array([-0.84830163, -1.22843865])

Isotropic and Anisotropic with Time = 0.4 min

```
In [3]: #Isotropic Sin-Cos Function
PREFIX="003"
fun = analytical().fun_sin_cos
a = 2* np.pi
#print(a)
fun_control = fun_control_init(PREFIX=PREFIX,
                                fun_evals=inf,
                                max_time=0.4,
                                lower = np.array([-a,-a]),
                                upper = np.array([a,a]))
surrogate_control=surrogate_control_init(n_theta=1)
spot_2 = spot.Spot(fun=fun,
                    fun_control=fun_control,
                    surrogate_control=surrogate_control)

spot_2.run()
spot_2.print_results()
spot_2.plot_progress()
spot_2.surrogate.plot()
spot_2.surrogate.theta
```

[rank: 0] Seed set to 123

```
Created spot_tensorboard_path: runs/spot_logs/003_c03-b01_2024-05-27_16-26-32 for SummaryWriter()
spotPython tuning: -2.032916688789623 [-----] 1.09%
spotPython tuning: -2.388464187850052 [-----] 2.06%
spotPython tuning: -2.4732239160328326 [-----] 2.86%
spotPython tuning: -2.4732239160328326 [-----] 3.81%
spotPython tuning: -2.4892152901553843 [-----] 4.59%
spotPython tuning: -2.4991257784759258 [#-----] 5.42%
spotPython tuning: -2.4991257784759258 [#-----] 6.26%
spotPython tuning: -2.4999988700971447 [#-----] 7.07%
spotPython tuning: -2.499999950865807 [#-----] 7.95%
spotPython tuning: -2.4999999509691717 [#-----] 8.74%
spotPython tuning: -2.499999956610016 [#-----] 9.49%
spotPython tuning: -2.499999956610016 [#-----] 10.26%
spotPython tuning: -2.4999999618068043 [#-----] 11.07%
spotPython tuning: -2.4999999618068043 [#-----] 11.82%
spotPython tuning: -2.4999999675987525 [#-----] 12.74%
spotPython tuning: -2.4999999675987525 [#-----] 13.57%
spotPython tuning: -2.4999999675987525 [#-----] 14.80%
spotPython tuning: -2.4999999675987525 [##-----] 15.89%
spotPython tuning: -2.4999999675987525 [##-----] 17.43%
spotPython tuning: -2.4999999675987525 [##-----] 18.71%
spotPython tuning: -2.4999999675987525 [##-----] 20.27%
spotPython tuning: -2.4999999675987525 [##-----] 21.55%
spotPython tuning: -2.4999999675987525 [##-----] 23.05%
spotPython tuning: -2.4999999675987525 [##-----] 24.38%
spotPython tuning: -2.4999999675987525 [##-----] 25.92%
spotPython tuning: -2.4999999675987525 [##-----] 27.42%
spotPython tuning: -2.4999999675987525 [##-----] 29.11%
spotPython tuning: -2.4999999675987525 [##-----] 30.60%
spotPython tuning: -2.4999999675987525 [##-----] 32.11%
spotPython tuning: -2.4999999675987525 [##-----] 33.39%
spotPython tuning: -2.4999999675987525 [##-----] 34.93%
spotPython tuning: -2.4999999675987525 [##-----] 36.23%
spotPython tuning: -2.4999999675987525 [##-----] 37.95%
spotPython tuning: -2.4999999675987525 [##-----] 39.31%
spotPython tuning: -2.4999999675987525 [##-----] 40.83%
spotPython tuning: -2.4999999675987525 [##-----] 42.15%
spotPython tuning: -2.4999999675987525 [##-----] 43.59%
spotPython tuning: -2.4999999675987525 [##-----] 44.89%
spotPython tuning: -2.4999999675987525 [##-----] 46.14%
spotPython tuning: -2.4999999675987525 [##-----] 47.69%
spotPython tuning: -2.4999999675987525 [##-----] 49.01%
spotPython tuning: -2.4999999675987525 [##-----] 50.19%
spotPython tuning: -2.4999999675987525 [##-----] 51.51%
spotPython tuning: -2.4999999675987525 [##-----] 52.78%
spotPython tuning: -2.4999999675987525 [##-----] 54.11%
spotPython tuning: -2.4999999675987525 [##-----] 55.48%
spotPython tuning: -2.4999999675987525 [##-----] 56.90%
spotPython tuning: -2.4999999675987525 [##-----] 58.18%
spotPython tuning: -2.4999999675987525 [##-----] 59.85%
spotPython tuning: -2.4999999675987525 [##-----] 61.43%
spotPython tuning: -2.4999999675987525 [##-----] 62.72%
spotPython tuning: -2.4999999675987525 [##-----] 63.99%
spotPython tuning: -2.4999999675987525 [##-----] 65.27%
spotPython tuning: -2.4999999675987525 [##-----] 66.49%
spotPython tuning: -2.4999999675987525 [##-----] 67.75%
spotPython tuning: -2.4999999675987525 [##-----] 69.02%
spotPython tuning: -2.4999999675987525 [##-----] 70.32%
spotPython tuning: -2.4999999675987525 [##-----] 71.84%
```

```

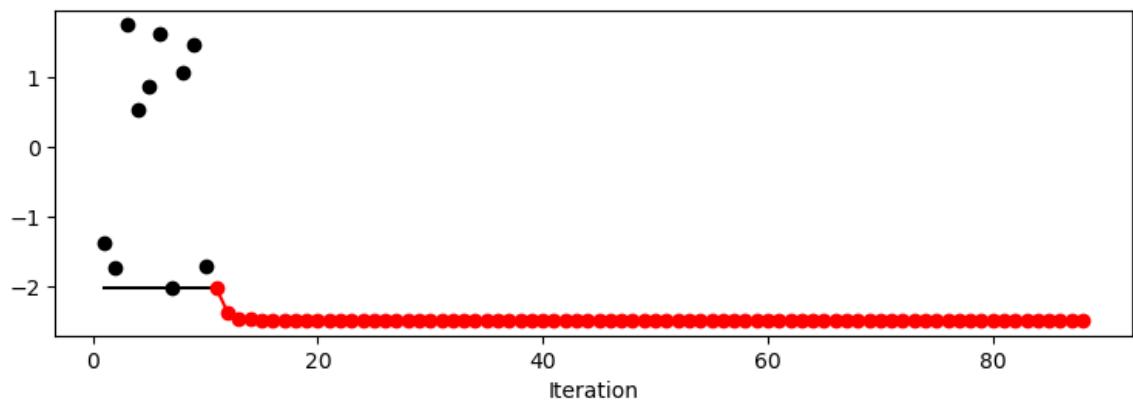
spotPython tuning: -2.4999999675987525 [#####---] 73.43%
spotPython tuning: -2.4999999675987525 [#####---] 74.78%
spotPython tuning: -2.4999999675987525 [#####---] 76.28%
spotPython tuning: -2.4999999675987525 [#####---] 77.97%
spotPython tuning: -2.4999999675987525 [#####---] 79.69%
spotPython tuning: -2.4999999675987525 [#####---] 81.17%
spotPython tuning: -2.4999999675987525 [#####---] 82.57%
spotPython tuning: -2.4999999675987525 [#####---] 84.06%
spotPython tuning: -2.4999999675987525 [#####---] 85.35%
spotPython tuning: -2.4999999675987525 [#####---] 87.07%
spotPython tuning: -2.4999999675987525 [#####---] 88.74%
spotPython tuning: -2.4999999675987525 [#####---] 89.97%
spotPython tuning: -2.4999999675987525 [#####---] 91.56%
spotPython tuning: -2.4999999675987525 [#####---] 92.83%
spotPython tuning: -2.4999999675987525 [#####---] 94.34%
spotPython tuning: -2.4999999675987525 [#####---] 95.53%
spotPython tuning: -2.4999999675987525 [#####---] 96.54%
spotPython tuning: -2.4999999675987525 [#####---] 97.79%
spotPython tuning: -2.4999999675987525 [#####---] 99.04%
spotPython tuning: -2.4999999675987525 [#####---] 100.00%
{'CHECKPOINT_PATH': 'runs/saved_models/',
 'DATASET_PATH': 'data/',
 'PREFIX': '003',
 'RESULTS_PATH': 'results/',
 'TENSORBOARD_PATH': 'runs/',
 '_L_in': None,
 '_L_out': None,
 '_torchmetric': None,
 'accelerator': 'auto',
 'converters': None,
 'core_model': None,
 'core_model_name': None,
 'counter': 88,
 'data': None,
 'data_dir': './data',
 'data_module': None,
 'data_set': None,
 'data_set_name': None,
 'db_dict_name': None,
 'design': None,
 'device': None,
 'devices': 1,
 'enable_progress_bar': False,
 'eval': None,
 'fun_evals': inf,
 'fun_repeats': 1,
 'horizon': None,
 'infill_criterion': 'y',
 'k_folds': 3,
 'log_graph': False,
 'log_level': 50,
 'loss_function': None,
 'lower': array([-6.28318531, -6.28318531]),
 'max_surrogate_points': 30,
 'max_time': 0.4,
 'metric_params': {},
 'metric_river': None,
 'metric_sklearn': None,
 'metric_sklearn_name': None,
 'metric_torch': None,
}

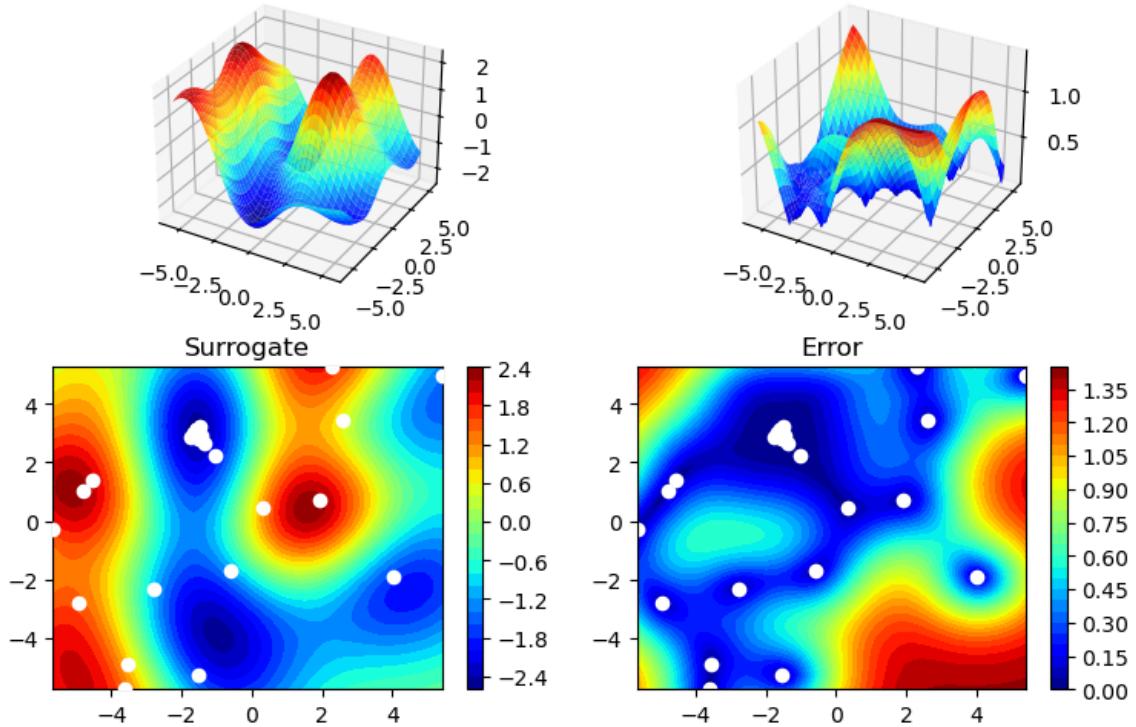
```

```

'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/003_c03-b01_2024-05-27_16-26-3
2',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x148996592f50>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([6.28318531, 6.28318531]),
'vername': None,
'ver_type': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: -2.499999675987525
x0: -1.5706722620612161
x1: 3.141534921552315

```





Out[3]: array([-0.88803075])

```
In [4]: #Anisotropic Sin-Cos Function
surrogate_control = surrogate_control_init(n_theta=2)
spot_2_anisotropic = spot.Spot(fun=fun,
                               fun_control=fun_control,
                               surrogate_control=surrogate_control)
spot_2_anisotropic.run()
spot_2_anisotropic.print_results()
spot_2_anisotropic.surrogate.plot()
spot_2_anisotropic.surrogate.theta
```

```

spotPython tuning: -2.2653112883046314 [-----] 1.32%
spotPython tuning: -2.2653112883046314 [-----] 2.71%
spotPython tuning: -2.2653112883046314 [-----] 3.92%
spotPython tuning: -2.454202160223451 [#-----] 5.37%
spotPython tuning: -2.458521375229338 [#-----] 6.58%
spotPython tuning: -2.482462560531697 [#-----] 8.00%
spotPython tuning: -2.482925901019679 [#-----] 9.12%
spotPython tuning: -2.4904698041359308 [#-----] 10.42%
spotPython tuning: -2.4991596489166508 [#-----] 11.60%
spotPython tuning: -2.499962076699469 [#-----] 13.05%
spotPython tuning: -2.4999776386398036 [#-----] 14.59%
spotPython tuning: -2.4999799363486916 [##-----] 15.86%
spotPython tuning: -2.4999804518304902 [##-----] 17.51%
spotPython tuning: -2.499981664757109 [##-----] 18.93%
spotPython tuning: -2.499981664757109 [##-----] 20.18%
spotPython tuning: -2.49998242826458 [##-----] 21.72%
spotPython tuning: -2.4999825326495873 [##-----] 23.52%
spotPython tuning: -2.4999928232338777 [##-----] 24.96%
spotPython tuning: -2.4999928232338777 [###-----] 26.40%
spotPython tuning: -2.4999995278811955 [###-----] 27.92%
spotPython tuning: -2.4999995278811955 [###-----] 29.91%
spotPython tuning: -2.49999998300882 [###-----] 31.65%
spotPython tuning: -2.49999998739451 [###-----] 33.15%
spotPython tuning: -2.499999996677 [###-----] 34.90%
spotPython tuning: -2.499999996677 [####-----] 37.04%
spotPython tuning: -2.499999996677 [####-----] 39.07%
spotPython tuning: -2.499999996677 [####-----] 41.18%
spotPython tuning: -2.4999999996677 [####-----] 43.45%
spotPython tuning: -2.4999999996677 [#####-----] 45.17%
spotPython tuning: -2.4999999996677 [#####-----] 47.10%
spotPython tuning: -2.4999999996677 [#####-----] 49.20%
spotPython tuning: -2.4999999996677 [#####-----] 51.47%
spotPython tuning: -2.4999999996677 [#####-----] 53.55%
spotPython tuning: -2.4999999996677 [#####-----] 55.74%
spotPython tuning: -2.4999999996677 [#####-----] 57.62%
spotPython tuning: -2.4999999996677 [#####-----] 59.80%
spotPython tuning: -2.4999999996677 [#####-----] 61.99%
spotPython tuning: -2.4999999996677 [#####-----] 64.27%
spotPython tuning: -2.4999999996677 [#####-----] 66.16%
spotPython tuning: -2.4999999996677 [#####-----] 68.03%
spotPython tuning: -2.4999999996677 [#####-----] 69.78%
spotPython tuning: -2.4999999996677 [#####-----] 71.68%
spotPython tuning: -2.4999999996677 [#####-----] 73.40%
spotPython tuning: -2.4999999996677 [#####-----] 75.31%
spotPython tuning: -2.4999999996677 [#####-----] 76.99%
spotPython tuning: -2.4999999996677 [#####-----] 78.67%
spotPython tuning: -2.4999999996677 [#####-----] 80.58%
spotPython tuning: -2.4999999996677 [#####-----] 82.52%
spotPython tuning: -2.4999999996677 [#####-----] 84.79%
spotPython tuning: -2.4999999996677 [#####-----] 86.53%
spotPython tuning: -2.4999999996677 [#####-----] 88.56%
spotPython tuning: -2.4999999996677 [#####-----] 90.55%
spotPython tuning: -2.4999999996677 [#####-----] 92.30%
spotPython tuning: -2.4999999996677 [#####-----] 94.03%
spotPython tuning: -2.4999999996677 [#####-----] 95.77%
spotPython tuning: -2.4999999996677 [#####-----] 97.89%
spotPython tuning: -2.4999999996677 [#####-----] 99.77%
spotPython tuning: -2.4999999996677 [#####-----] 100.00% Done...

```

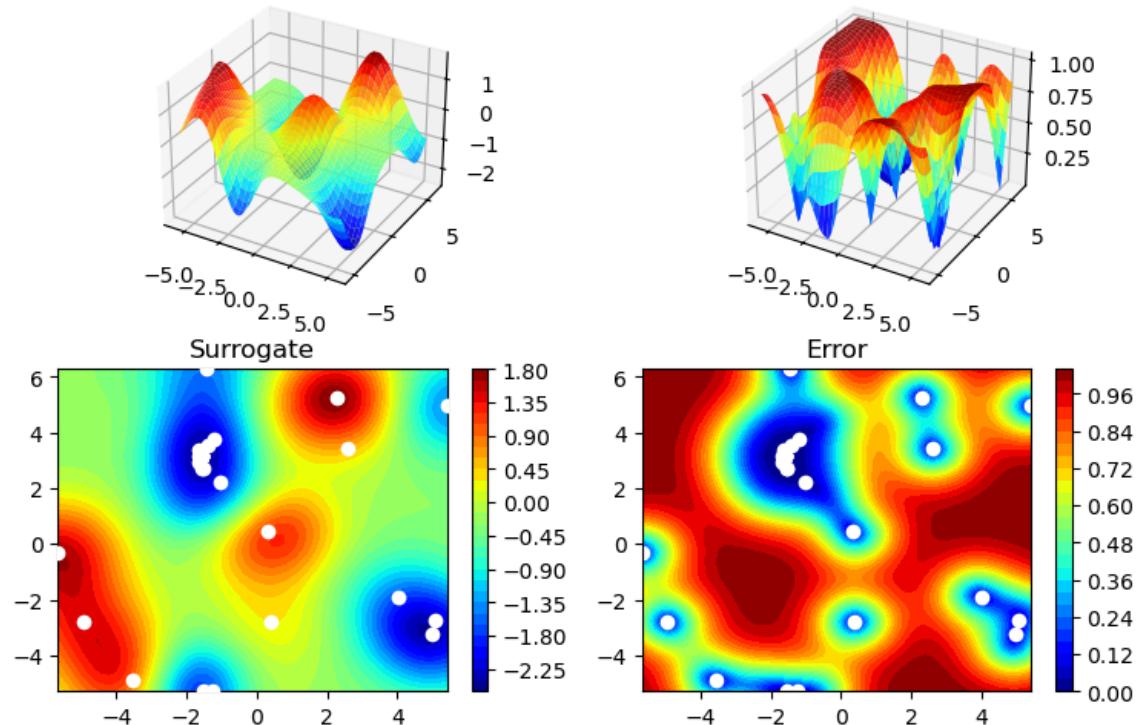
```
{'CHECKPOINT_PATH': 'runs/saved_models/>,
```

```
'DATASET_PATH': 'data/',
'PREFIX': '003',
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,
'core_model_name': None,
'counter': 68,
'data': None,
'data_dir': './data',
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': inf,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([-6.28318531, -6.28318531]),
'max_surrogate_points': 30,
'max_time': 0.4,
'metric_params': {},
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/003_c03-b01_2024-05-27_16-26-3
```

```

2',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x148996592f50>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([6.28318531, 6.28318531]),
'ver_name': None,
'ver_type': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: -2.4999999996677
x0: -1.5707818548046566
x1: 3.14161482215979

```



Out[4]: array([-0.50815466, -0.51579636])

Result of Sin-Cos Function

Function	n_theta	max_time(min)	Number of Evaluations	Number of Iterations	Objective function Value
Sin-Cos Function	1	1	inf	228	-2.4999999675987525
Sin-Cos Function	2	1	inf	155	-2.4999999996677
Sin-Cos Function	1	0.4	inf	88	-2.4999999675987525

Function	n_theta	max_time(min)	Number of Evaluations	Number of Iterations	Objective function Value
Sin-Cos Function	2	0.4	inf	68	-2.49999999996677

Increasing `n_theta` from 1 to 2 consistently reduces the number of iterations needed to reach the optimal objective function value for the Sin-Cos function, from 228 to 155 iterations for `max_time` set to 1 minute and from 88 to 68 iterations for `max_time` set to 0.4 minutes, while maintaining the same objective function value. This suggests that a higher `n_theta` improves optimization efficiency without compromising the quality of the solution.

3. The Two-dimensional Runge Function `fun_runge`

- Describe the function.
 - * Formula: $f(x) = 1 / (1 + \sum(x_i) - \text{offset})^2$
 Dim: $k \geq 1$
 Interval: $-5 \leq x \leq 5$
 - The input dimension is 2. The search range is $-5 \leq x_1 \leq 5$ and $-5 \leq x_2 \leq 5$.
- Compare the results from `spotPython` run a) with isotropic and b) anisotropic surrogate models.
- Modify the termination criterion (`max_time` instead of `fun_evals`) as described for `fun_branin`.### 3. The Two-dimensional Runge Function `fun_runge`
- Describe the function.
 - * Formula: $f(x) = 1 / (1 + \sum(x_i) - \text{offset})^2$
 Dim: $k \geq 1$
 Interval: $-5 \leq x \leq 5$
 - The input dimension is 2. The search range is $-5 \leq x_1 \leq 5$ and $-5 \leq x_2 \leq 5$.
- Compare the results from `spotPython` run a) with isotropic and b) anisotropic surrogate models.
- Modify the termination criterion (`max_time` instead of `fun_evals`) as described for `fun_branin`.

Isotropic and Anisotropic with time = 1 min

```
In [6]: #Isotropic Runge Function
PREFIX="003"
fun = analytical().fun_runge

fun_control = fun_control_init(PREFIX=PREFIX,
                               fun_evals=inf,
                               max_time=1,
```

```
lower = np.array([-5,-5]),
upper = np.array([5,5]))
surrogate_control=surrogate_control_init(n_theta=1)
spot_2 = spot.Spot(fun=fun,
                    fun_control=fun_control,
                    surrogate_control=surrogate_control)

spot_2.run()
spot_2.print_results()
spot_2.plot_progress()
spot_2.surrogate.plot()
spot_2.surrogate.theta
```

```
[rank: 0] Seed set to 123
```

```
Created spot_tensorboard_path: runs/spot_logs/003_c02-b04_2024-05-21_13-29-56 for SummaryWriter()
spotPython tuning: 0.028397462483452193 [-----] 0.30%
spotPython tuning: 0.027658261922815508 [-----] 0.55%
spotPython tuning: 0.027658261922815508 [-----] 0.90%
spotPython tuning: 0.027391176020978375 [-----] 1.18%
spotPython tuning: 0.02381296964249503 [-----] 1.57%
spotPython tuning: 0.02381296964249503 [-----] 1.86%
spotPython tuning: 0.02381296964249503 [-----] 2.22%
spotPython tuning: 0.022058326824035742 [-----] 2.71%
spotPython tuning: 0.022058326824035742 [-----] 3.03%
spotPython tuning: 0.020829601115955484 [-----] 3.51%
spotPython tuning: 0.020348168886124868 [-----] 3.94%
spotPython tuning: 0.020348168886124868 [-----] 4.28%
spotPython tuning: 0.0196078431372549 [-----] 4.93%
spotPython tuning: 0.0196078431372549 [#-----] 5.47%
spotPython tuning: 0.0196078431372549 [#-----] 6.01%
spotPython tuning: 0.0196078431372549 [#-----] 6.46%
spotPython tuning: 0.0196078431372549 [#-----] 6.84%
spotPython tuning: 0.0196078431372549 [#-----] 7.41%
spotPython tuning: 0.0196078431372549 [#-----] 7.85%
spotPython tuning: 0.0196078431372549 [#-----] 8.69%
spotPython tuning: 0.0196078431372549 [#-----] 9.30%
spotPython tuning: 0.0196078431372549 [#-----] 9.79%
spotPython tuning: 0.0196078431372549 [#-----] 10.31%
spotPython tuning: 0.0196078431372549 [#-----] 10.81%
spotPython tuning: 0.0196078431372549 [#-----] 11.30%
spotPython tuning: 0.0196078431372549 [#-----] 11.79%
spotPython tuning: 0.0196078431372549 [#-----] 12.34%
spotPython tuning: 0.0196078431372549 [#-----] 12.83%
spotPython tuning: 0.0196078431372549 [#-----] 13.48%
spotPython tuning: 0.0196078431372549 [#-----] 14.14%
spotPython tuning: 0.0196078431372549 [#-----] 14.80%
spotPython tuning: 0.0196078431372549 [##-----] 15.45%
spotPython tuning: 0.0196078431372549 [##-----] 16.11%
spotPython tuning: 0.0196078431372549 [##-----] 16.80%
spotPython tuning: 0.0196078431372549 [##-----] 17.45%
spotPython tuning: 0.0196078431372549 [##-----] 18.11%
spotPython tuning: 0.0196078431372549 [##-----] 18.76%
spotPython tuning: 0.0196078431372549 [##-----] 19.42%
spotPython tuning: 0.0196078431372549 [##-----] 20.08%
spotPython tuning: 0.0196078431372549 [##-----] 20.81%
spotPython tuning: 0.0196078431372549 [##-----] 21.47%
spotPython tuning: 0.0196078431372549 [##-----] 22.13%
spotPython tuning: 0.0196078431372549 [##-----] 22.78%
spotPython tuning: 0.0196078431372549 [##-----] 23.43%
spotPython tuning: 0.0196078431372549 [##-----] 24.07%
spotPython tuning: 0.0196078431372549 [##-----] 24.71%
spotPython tuning: 0.0196078431372549 [###-----] 25.33%
spotPython tuning: 0.0196078431372549 [###-----] 25.86%
spotPython tuning: 0.0196078431372549 [###-----] 26.41%
spotPython tuning: 0.0196078431372549 [###-----] 27.02%
spotPython tuning: 0.0196078431372549 [###-----] 27.62%
spotPython tuning: 0.0196078431372549 [###-----] 28.25%
spotPython tuning: 0.0196078431372549 [###-----] 28.71%
spotPython tuning: 0.0196078431372549 [###-----] 29.22%
spotPython tuning: 0.0196078431372549 [###-----] 29.67%
spotPython tuning: 0.0196078431372549 [###-----] 30.12%
spotPython tuning: 0.0196078431372549 [###-----] 30.74%
spotPython tuning: 0.0196078431372549 [###-----] 31.17%
```

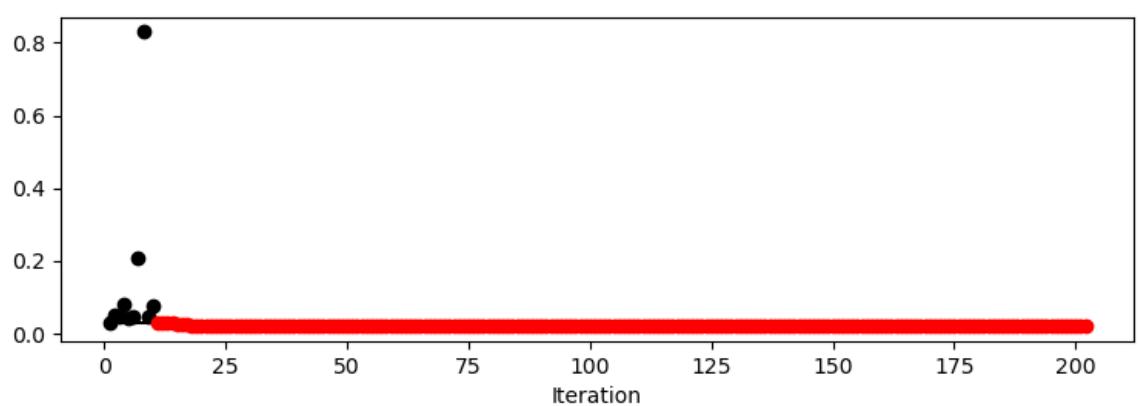
```
spotPython tuning: 0.0196078431372549 [#####-----] 31.80%
spotPython tuning: 0.0196078431372549 [#####-----] 32.28%
spotPython tuning: 0.0196078431372549 [#####-----] 32.73%
spotPython tuning: 0.0196078431372549 [#####-----] 33.19%
spotPython tuning: 0.0196078431372549 [#####-----] 33.70%
spotPython tuning: 0.0196078431372549 [#####-----] 34.14%
spotPython tuning: 0.0196078431372549 [#####-----] 34.76%
spotPython tuning: 0.0196078431372549 [#####-----] 35.18%
spotPython tuning: 0.0196078431372549 [#####-----] 35.65%
spotPython tuning: 0.0196078431372549 [#####-----] 36.19%
spotPython tuning: 0.0196078431372549 [#####-----] 36.75%
spotPython tuning: 0.0196078431372549 [#####-----] 37.41%
spotPython tuning: 0.0196078431372549 [#####-----] 37.96%
spotPython tuning: 0.0196078431372549 [#####-----] 38.53%
spotPython tuning: 0.0196078431372549 [#####-----] 39.14%
spotPython tuning: 0.0196078431372549 [#####-----] 39.72%
spotPython tuning: 0.0196078431372549 [#####-----] 40.36%
spotPython tuning: 0.0196078431372549 [#####-----] 40.93%
spotPython tuning: 0.0196078431372549 [#####-----] 41.45%
spotPython tuning: 0.0196078431372549 [#####-----] 41.96%
spotPython tuning: 0.0196078431372549 [#####-----] 42.60%
spotPython tuning: 0.0196078431372549 [#####-----] 43.07%
spotPython tuning: 0.0196078431372549 [#####-----] 43.47%
spotPython tuning: 0.0196078431372549 [#####-----] 44.07%
spotPython tuning: 0.0196078431372549 [#####-----] 44.57%
spotPython tuning: 0.0196078431372549 [#####-----] 45.12%
spotPython tuning: 0.0196078431372549 [#####-----] 45.72%
spotPython tuning: 0.0196078431372549 [#####-----] 46.32%
spotPython tuning: 0.0196078431372549 [#####-----] 46.75%
spotPython tuning: 0.0196078431372549 [#####-----] 47.32%
spotPython tuning: 0.0196078431372549 [#####-----] 47.94%
spotPython tuning: 0.0196078431372549 [#####-----] 48.48%
spotPython tuning: 0.0196078431372549 [#####-----] 49.11%
spotPython tuning: 0.0196078431372549 [#####-----] 49.65%
spotPython tuning: 0.0196078431372549 [#####-----] 50.29%
spotPython tuning: 0.0196078431372549 [#####-----] 50.93%
spotPython tuning: 0.0196078431372549 [#####-----] 51.41%
spotPython tuning: 0.0196078431372549 [#####-----] 52.50%
spotPython tuning: 0.0196078431372549 [#####-----] 53.21%
spotPython tuning: 0.0196078431372549 [#####-----] 53.83%
spotPython tuning: 0.0196078431372549 [#####-----] 54.36%
spotPython tuning: 0.0196078431372549 [#####-----] 54.87%
spotPython tuning: 0.0196078431372549 [#####-----] 55.26%
spotPython tuning: 0.0196078431372549 [#####-----] 55.67%
spotPython tuning: 0.0196078431372549 [#####-----] 56.16%
spotPython tuning: 0.0196078431372549 [#####-----] 56.55%
spotPython tuning: 0.0196078431372549 [#####-----] 56.94%
spotPython tuning: 0.0196078431372549 [#####-----] 57.34%
spotPython tuning: 0.0196078431372549 [#####-----] 57.91%
spotPython tuning: 0.0196078431372549 [#####-----] 58.32%
spotPython tuning: 0.0196078431372549 [#####-----] 58.74%
spotPython tuning: 0.0196078431372549 [#####-----] 59.18%
spotPython tuning: 0.0196078431372549 [#####-----] 59.67%
spotPython tuning: 0.0196078431372549 [#####-----] 60.29%
spotPython tuning: 0.0196078431372549 [#####-----] 60.92%
spotPython tuning: 0.0196078431372549 [#####-----] 61.53%
spotPython tuning: 0.0196078431372549 [#####-----] 62.18%
spotPython tuning: 0.0196078431372549 [#####-----] 62.75%
spotPython tuning: 0.0196078431372549 [#####-----] 63.18%
spotPython tuning: 0.0196078431372549 [#####-----] 63.57%
```

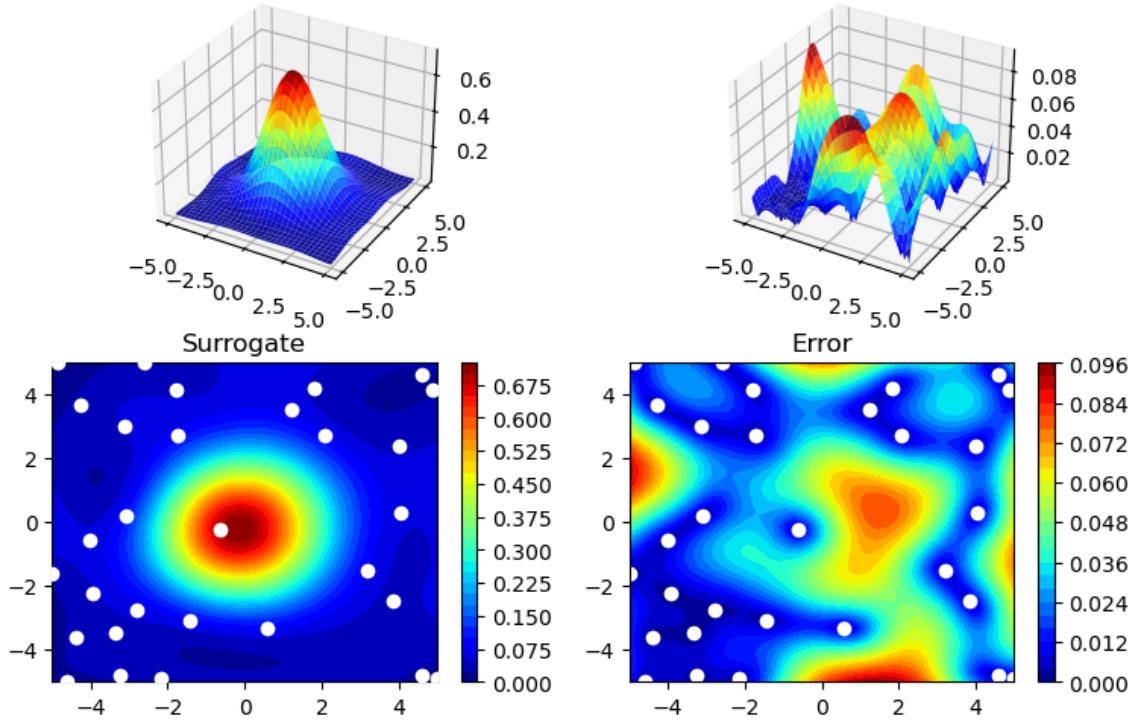
```
spotPython tuning: 0.0196078431372549 [#####----] 63.95%
spotPython tuning: 0.0196078431372549 [#####----] 64.48%
spotPython tuning: 0.0196078431372549 [#####----] 64.82%
spotPython tuning: 0.0196078431372549 [#####----] 65.25%
spotPython tuning: 0.0196078431372549 [#####----] 65.87%
spotPython tuning: 0.0196078431372549 [#####----] 66.46%
spotPython tuning: 0.0196078431372549 [#####----] 66.91%
spotPython tuning: 0.0196078431372549 [#####----] 67.27%
spotPython tuning: 0.0196078431372549 [#####----] 67.90%
spotPython tuning: 0.0196078431372549 [#####----] 68.36%
spotPython tuning: 0.0196078431372549 [#####----] 68.90%
spotPython tuning: 0.0196078431372549 [#####----] 69.44%
spotPython tuning: 0.0196078431372549 [#####----] 69.96%
spotPython tuning: 0.0196078431372549 [#####----] 70.40%
spotPython tuning: 0.0196078431372549 [#####----] 70.87%
spotPython tuning: 0.0196078431372549 [#####----] 71.33%
spotPython tuning: 0.0196078431372549 [#####----] 71.81%
spotPython tuning: 0.0196078431372549 [#####----] 72.28%
spotPython tuning: 0.0196078431372549 [#####----] 72.71%
spotPython tuning: 0.0196078431372549 [#####----] 73.28%
spotPython tuning: 0.0196078431372549 [#####----] 74.03%
spotPython tuning: 0.0196078431372549 [#####----] 74.43%
spotPython tuning: 0.0196078431372549 [#####----] 74.78%
spotPython tuning: 0.0196078431372549 [#####----] 75.42%
spotPython tuning: 0.0196078431372549 [#####----] 75.78%
spotPython tuning: 0.0196078431372549 [#####----] 76.26%
spotPython tuning: 0.0196078431372549 [#####----] 76.73%
spotPython tuning: 0.0196078431372549 [#####----] 77.38%
spotPython tuning: 0.0196078431372549 [#####----] 77.94%
spotPython tuning: 0.0196078431372549 [#####----] 78.37%
spotPython tuning: 0.0196078431372549 [#####----] 78.73%
spotPython tuning: 0.0196078431372549 [#####----] 79.19%
spotPython tuning: 0.0196078431372549 [#####----] 79.60%
spotPython tuning: 0.0196078431372549 [#####----] 79.99%
spotPython tuning: 0.0196078431372549 [#####----] 80.50%
spotPython tuning: 0.0196078431372549 [#####----] 80.90%
spotPython tuning: 0.0196078431372549 [#####----] 81.30%
spotPython tuning: 0.0196078431372549 [#####----] 81.71%
spotPython tuning: 0.0196078431372549 [#####----] 82.36%
spotPython tuning: 0.0196078431372549 [#####----] 82.97%
spotPython tuning: 0.0196078431372549 [#####----] 83.36%
spotPython tuning: 0.0196078431372549 [#####----] 83.86%
spotPython tuning: 0.0196078431372549 [#####----] 84.23%
spotPython tuning: 0.0196078431372549 [#####----] 84.73%
spotPython tuning: 0.0196078431372549 [#####----] 85.10%
spotPython tuning: 0.0196078431372549 [#####----] 85.92%
spotPython tuning: 0.0196078431372549 [#####----] 86.53%
spotPython tuning: 0.0196078431372549 [#####----] 87.00%
spotPython tuning: 0.0196078431372549 [#####----] 87.53%
spotPython tuning: 0.0196078431372549 [#####----] 87.98%
spotPython tuning: 0.0196078431372549 [#####----] 88.45%
spotPython tuning: 0.0196078431372549 [#####----] 89.02%
spotPython tuning: 0.0196078431372549 [#####----] 89.37%
spotPython tuning: 0.0196078431372549 [#####----] 89.80%
spotPython tuning: 0.0196078431372549 [#####----] 90.35%
spotPython tuning: 0.0196078431372549 [#####----] 90.74%
spotPython tuning: 0.0196078431372549 [#####----] 91.17%
spotPython tuning: 0.0196078431372549 [#####----] 91.60%
spotPython tuning: 0.0196078431372549 [#####----] 92.18%
spotPython tuning: 0.0196078431372549 [#####----] 92.84%
```

```
spotPython tuning: 0.0196078431372549 [#####] 93.28%
spotPython tuning: 0.0196078431372549 [#####] 93.93%
spotPython tuning: 0.0196078431372549 [#####] 94.51%
spotPython tuning: 0.0196078431372549 [#####] 95.07%
spotPython tuning: 0.0196078431372549 [#####] 95.46%
spotPython tuning: 0.0196078431372549 [#####] 95.78%
spotPython tuning: 0.0196078431372549 [#####] 96.25%
spotPython tuning: 0.0196078431372549 [#####] 96.74%
spotPython tuning: 0.0196078431372549 [#####] 97.22%
spotPython tuning: 0.0196078431372549 [#####] 97.98%
spotPython tuning: 0.0196078431372549 [#####] 98.67%
spotPython tuning: 0.0196078431372549 [#####] 99.28%
spotPython tuning: 0.0196078431372549 [#####] 99.65%
spotPython tuning: 0.0196078431372549 [#####] 100.00% Done...

{'CHECKPOINT_PATH': 'runs/saved_models/',
 'DATASET_PATH': 'data/',
 'PREFIX': '003',
 'RESULTS_PATH': 'results/',
 'TENSORBOARD_PATH': 'runs/',
 '_L_in': None,
 '_L_out': None,
 '_torchmetric': None,
 'accelerator': 'auto',
 'converters': None,
 'core_model': None,
 'core_model_name': None,
 'counter': 202,
 'data': None,
 'data_dir': './data',
 'data_module': None,
 'data_set': None,
 'data_set_name': None,
 'db_dict_name': None,
 'design': None,
 'device': None,
 'devices': 1,
 'enable_progress_bar': False,
 'eval': None,
 'fun_evals': inf,
 'fun_repeats': 1,
 'horizon': None,
 'infill_criterion': 'y',
 'k_folds': 3,
 'log_graph': False,
 'log_level': 50,
 'loss_function': None,
 'lower': array([-5, -5]),
 'max_surrogate_points': 30,
 'max_time': 1,
 'metric_params': {},
 'metric_river': None,
 'metric_sklearn': None,
 'metric_sklearn_name': None,
 'metric_torch': None,
 'model_dict': {},
 'n_points': 1,
 'n_samples': None,
 'n_total': None,
 'noise': False,
```

```
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/003_c02-b04_2024-05-21_13-29-5
6',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x154d3847db50>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([5, 5]),
'vername': None,
'ver_type': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 0.0196078431372549
x0: 5.0
x1: 5.0
```





Out[6]: array([-0.83843666])

In [7]: #Anisotropic Runge Function
 surrogate_control = surrogate_control_init(n_theta=2)
 spot_2_anisotropic = spot.Spot(fun=fun,
 fun_control=fun_control,
 surrogate_control=surrogate_control)
 spot_2_anisotropic.run()
 spot_2_anisotropic.print_results()
 spot_2_anisotropic.surrogate.plot()
 spot_2_anisotropic.surrogate.theta

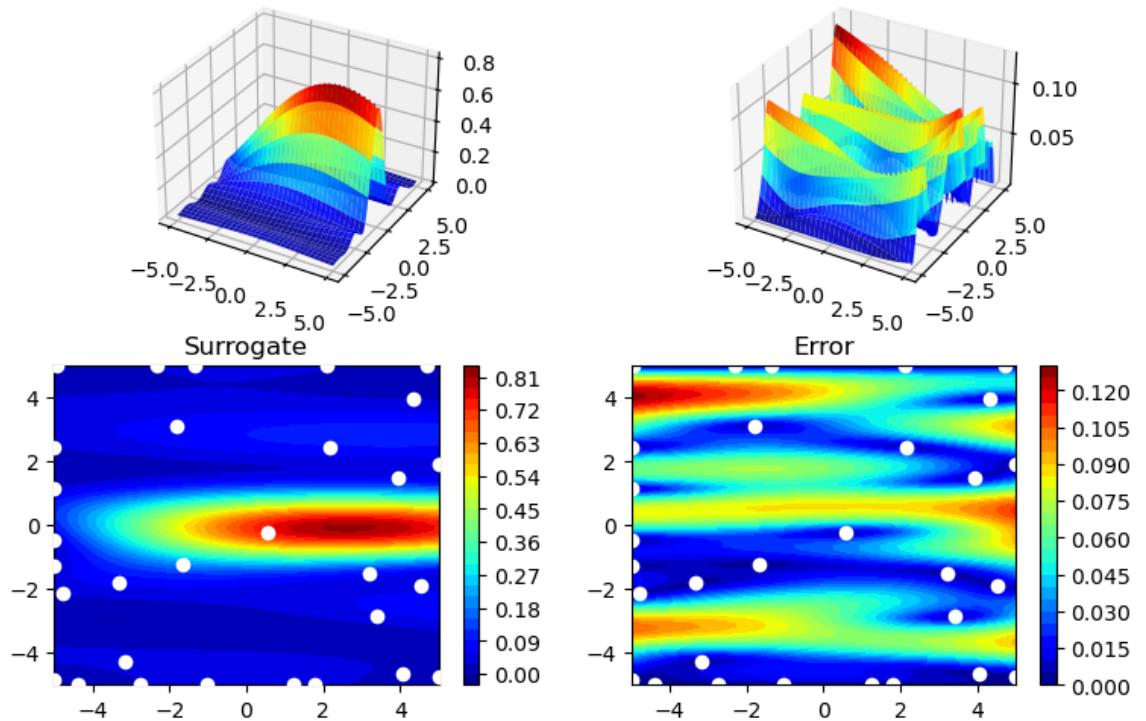
```
spotPython tuning: 0.028397462483452193 [-----] 0.45%
spotPython tuning: 0.028397462483452193 [-----] 0.88%
spotPython tuning: 0.028397462483452193 [-----] 1.32%
spotPython tuning: 0.021438173065031263 [-----] 1.84%
spotPython tuning: 0.021438173065031263 [-----] 2.37%
spotPython tuning: 0.021438173065031263 [-----] 2.96%
spotPython tuning: 0.0196078431372549 [-----] 3.91%
spotPython tuning: 0.0196078431372549 [-----] 4.65%
spotPython tuning: 0.0196078431372549 [#-----] 5.39%
spotPython tuning: 0.0196078431372549 [#-----] 6.04%
spotPython tuning: 0.0196078431372549 [#-----] 6.83%
spotPython tuning: 0.0196078431372549 [#-----] 7.39%
spotPython tuning: 0.0196078431372549 [#-----] 8.13%
spotPython tuning: 0.0196078431372549 [#-----] 8.68%
spotPython tuning: 0.0196078431372549 [#-----] 9.22%
spotPython tuning: 0.0196078431372549 [#-----] 10.05%
spotPython tuning: 0.0196078431372549 [#-----] 10.75%
spotPython tuning: 0.0196078431372549 [#-----] 11.51%
spotPython tuning: 0.0196078431372549 [#-----] 11.99%
spotPython tuning: 0.0196078431372549 [#-----] 12.58%
spotPython tuning: 0.0196078431372549 [#-----] 13.13%
spotPython tuning: 0.0196078431372549 [#-----] 14.00%
spotPython tuning: 0.0196078431372549 [#-----] 14.82%
spotPython tuning: 0.0196078431372549 [##-----] 15.52%
spotPython tuning: 0.0196078431372549 [##-----] 16.20%
spotPython tuning: 0.0196078431372549 [##-----] 16.71%
spotPython tuning: 0.0196078431372549 [##-----] 17.45%
spotPython tuning: 0.0196078431372549 [##-----] 18.35%
spotPython tuning: 0.0196078431372549 [##-----] 19.01%
spotPython tuning: 0.0196078431372549 [##-----] 19.86%
spotPython tuning: 0.0196078431372549 [##-----] 20.93%
spotPython tuning: 0.0196078431372549 [##-----] 21.66%
spotPython tuning: 0.0196078431372549 [##-----] 22.36%
spotPython tuning: 0.0196078431372549 [##-----] 23.11%
spotPython tuning: 0.0196078431372549 [##-----] 23.82%
spotPython tuning: 0.0196078431372549 [##-----] 24.68%
spotPython tuning: 0.0196078431372549 [###-----] 25.90%
spotPython tuning: 0.0196078431372549 [###-----] 26.79%
spotPython tuning: 0.0196078431372549 [###-----] 27.91%
spotPython tuning: 0.0196078431372549 [###-----] 28.73%
spotPython tuning: 0.0196078431372549 [###-----] 29.75%
spotPython tuning: 0.0196078431372549 [###-----] 30.63%
spotPython tuning: 0.0196078431372549 [###-----] 31.39%
spotPython tuning: 0.0196078431372549 [###-----] 32.06%
spotPython tuning: 0.0196078431372549 [###-----] 32.98%
spotPython tuning: 0.0196078431372549 [###-----] 33.68%
spotPython tuning: 0.0196078431372549 [###-----] 34.37%
spotPython tuning: 0.0196078431372549 [###-----] 34.98%
spotPython tuning: 0.0196078431372549 [####-----] 35.75%
spotPython tuning: 0.0196078431372549 [####-----] 36.60%
spotPython tuning: 0.0196078431372549 [####-----] 37.42%
spotPython tuning: 0.0196078431372549 [####-----] 38.26%
spotPython tuning: 0.0196078431372549 [####-----] 39.17%
spotPython tuning: 0.0196078431372549 [####-----] 39.87%
spotPython tuning: 0.0196078431372549 [####-----] 40.75%
spotPython tuning: 0.0196078431372549 [####-----] 41.39%
spotPython tuning: 0.0196078431372549 [####-----] 42.33%
spotPython tuning: 0.0196078431372549 [####-----] 42.99%
spotPython tuning: 0.0196078431372549 [####-----] 43.87%
spotPython tuning: 0.0196078431372549 [####-----] 45.02%
```

```
spotPython tuning: 0.0196078431372549 [#####-----] 45.85%
spotPython tuning: 0.0196078431372549 [#####-----] 46.51%
spotPython tuning: 0.0196078431372549 [#####-----] 47.20%
spotPython tuning: 0.0196078431372549 [#####-----] 48.23%
spotPython tuning: 0.0196078431372549 [#####-----] 49.69%
spotPython tuning: 0.0196078431372549 [#####-----] 50.41%
spotPython tuning: 0.0196078431372549 [#####-----] 51.10%
spotPython tuning: 0.0196078431372549 [#####-----] 51.79%
spotPython tuning: 0.0196078431372549 [#####-----] 52.50%
spotPython tuning: 0.0196078431372549 [#####-----] 53.06%
spotPython tuning: 0.0196078431372549 [#####-----] 54.39%
spotPython tuning: 0.0196078431372549 [#####-----] 55.03%
spotPython tuning: 0.0196078431372549 [#####-----] 56.03%
spotPython tuning: 0.0196078431372549 [#####-----] 56.88%
spotPython tuning: 0.0196078431372549 [#####-----] 57.73%
spotPython tuning: 0.0196078431372549 [#####-----] 58.78%
spotPython tuning: 0.0196078431372549 [#####-----] 59.45%
spotPython tuning: 0.0196078431372549 [#####-----] 60.14%
spotPython tuning: 0.0196078431372549 [#####-----] 60.92%
spotPython tuning: 0.0196078431372549 [#####-----] 62.00%
spotPython tuning: 0.0196078431372549 [#####-----] 62.84%
spotPython tuning: 0.0196078431372549 [#####-----] 63.69%
spotPython tuning: 0.0196078431372549 [#####-----] 64.47%
spotPython tuning: 0.0196078431372549 [#####-----] 65.22%
spotPython tuning: 0.0196078431372549 [#####-----] 65.93%
spotPython tuning: 0.0196078431372549 [#####-----] 66.78%
spotPython tuning: 0.0196078431372549 [#####-----] 67.57%
spotPython tuning: 0.0196078431372549 [#####-----] 68.37%
spotPython tuning: 0.0196078431372549 [#####-----] 69.25%
spotPython tuning: 0.0196078431372549 [#####-----] 70.15%
spotPython tuning: 0.0196078431372549 [#####-----] 71.12%
spotPython tuning: 0.0196078431372549 [#####-----] 71.88%
spotPython tuning: 0.0196078431372549 [#####-----] 72.70%
spotPython tuning: 0.0196078431372549 [#####-----] 73.61%
spotPython tuning: 0.0196078431372549 [#####-----] 74.57%
spotPython tuning: 0.0196078431372549 [#####-----] 75.54%
spotPython tuning: 0.0196078431372549 [#####-----] 76.43%
spotPython tuning: 0.0196078431372549 [#####-----] 77.06%
spotPython tuning: 0.0196078431372549 [#####-----] 77.69%
spotPython tuning: 0.0196078431372549 [#####-----] 78.41%
spotPython tuning: 0.0196078431372549 [#####-----] 79.08%
spotPython tuning: 0.0196078431372549 [#####-----] 79.73%
spotPython tuning: 0.0196078431372549 [#####-----] 80.38%
spotPython tuning: 0.0196078431372549 [#####-----] 81.11%
spotPython tuning: 0.0196078431372549 [#####-----] 81.81%
spotPython tuning: 0.0196078431372549 [#####-----] 82.50%
spotPython tuning: 0.0196078431372549 [#####-----] 83.27%
spotPython tuning: 0.0196078431372549 [#####-----] 84.09%
spotPython tuning: 0.0196078431372549 [#####-----] 84.85%
spotPython tuning: 0.0196078431372549 [#####-----] 85.73%
spotPython tuning: 0.0196078431372549 [#####-----] 86.66%
spotPython tuning: 0.0196078431372549 [#####-----] 87.46%
spotPython tuning: 0.0196078431372549 [#####-----] 88.22%
spotPython tuning: 0.0196078431372549 [#####-----] 88.92%
spotPython tuning: 0.0196078431372549 [#####-----] 89.89%
spotPython tuning: 0.0196078431372549 [#####-----] 90.65%
spotPython tuning: 0.0196078431372549 [#####-----] 91.61%
spotPython tuning: 0.0196078431372549 [#####-----] 92.39%
spotPython tuning: 0.0196078431372549 [#####-----] 93.24%
spotPython tuning: 0.0196078431372549 [#####-----] 94.18%
```

```
spotPython tuning: 0.0196078431372549 [#####] 94.95%
spotPython tuning: 0.0196078431372549 [#####] 95.87%
spotPython tuning: 0.0196078431372549 [#####] 96.64%
spotPython tuning: 0.0196078431372549 [#####] 97.24%
spotPython tuning: 0.0196078431372549 [#####] 98.18%
spotPython tuning: 0.0196078431372549 [#####] 98.82%
spotPython tuning: 0.0196078431372549 [#####] 99.66%
spotPython tuning: 0.0196078431372549 [#####] 100.00% Done...
```

```
{'CHECKPOINT_PATH': 'runs/saved_models/',
'DATASET_PATH': 'data/',
'PREFIX': '003',
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,
'core_model_name': None,
'counter': 138,
'data': None,
'data_dir': './data',
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': inf,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([-5, -5]),
'max_surrogate_points': 30,
'max_time': 1,
'metric_params': {},
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
```

```
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/003_c02-b04_2024-05-21_13-29-5
6',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x154d3847db50>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([5, 5]),
'vername': None,
'vertype': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 0.0196078431372549
x0: 5.0
x1: 5.0
```



Out[7]: array([-1.90286797, -0.13144966])

Isotropic and Anisotropic with time = 0.4 min

```
In [6]: #Isotropic Runge Function
PREFIX="003"
fun = analytical().fun_runge

fun_control = fun_control_init(PREFIX=PREFIX,
                                fun_evals=inf,
                                max_time=0.4,
                                lower = np.array([-5,-5]),
                                upper = np.array([5,5]))
surrogate_control=surrogate_control_init(n_theta=1)
spot_2 = spot.Spot(fun=fun,
                    fun_control=fun_control,
                    surrogate_control=surrogate_control)

spot_2.run()
spot_2.print_results()
spot_2.plot_progress()
spot_2.surrogate.plot()
spot_2.surrogate.theta
```

```
[rank: 0] Seed set to 123
```

```
Created spot_tensorboard_path: runs/spot_logs/003_c03-b01_2024-05-27_16-42-01 for SummaryWriter()
spotPython tuning: 0.028397462483452193 [-----] 0.74%
spotPython tuning: 0.027658261922815508 [-----] 1.36%
spotPython tuning: 0.027658261922815508 [-----] 2.20%
spotPython tuning: 0.027391176020978375 [-----] 2.86%
spotPython tuning: 0.02381296964249503 [-----] 3.80%
spotPython tuning: 0.02381296964249503 [-----] 4.51%
spotPython tuning: 0.02381296964249503 [#-----] 5.37%
spotPython tuning: 0.022058326824035742 [#-----] 6.52%
spotPython tuning: 0.022058326824035742 [#-----] 7.29%
spotPython tuning: 0.020829601115955484 [#-----] 8.30%
spotPython tuning: 0.020348168886124868 [#-----] 9.33%
spotPython tuning: 0.020348168886124868 [#-----] 10.20%
spotPython tuning: 0.0196078431372549 [#-----] 11.73%
spotPython tuning: 0.0196078431372549 [#-----] 13.05%
spotPython tuning: 0.0196078431372549 [#-----] 14.35%
spotPython tuning: 0.0196078431372549 [##-----] 15.46%
spotPython tuning: 0.0196078431372549 [##-----] 16.37%
spotPython tuning: 0.0196078431372549 [##-----] 17.69%
spotPython tuning: 0.0196078431372549 [##-----] 18.75%
spotPython tuning: 0.0196078431372549 [##-----] 20.73%
spotPython tuning: 0.0196078431372549 [##-----] 22.17%
spotPython tuning: 0.0196078431372549 [##-----] 23.37%
spotPython tuning: 0.0196078431372549 [##-----] 24.58%
spotPython tuning: 0.0196078431372549 [###-----] 25.80%
spotPython tuning: 0.0196078431372549 [###-----] 27.00%
spotPython tuning: 0.0196078431372549 [###-----] 28.17%
spotPython tuning: 0.0196078431372549 [###-----] 29.47%
spotPython tuning: 0.0196078431372549 [###-----] 30.65%
spotPython tuning: 0.0196078431372549 [###-----] 32.24%
spotPython tuning: 0.0196078431372549 [###-----] 33.80%
spotPython tuning: 0.0196078431372549 [###-----] 35.36%
spotPython tuning: 0.0196078431372549 [###-----] 36.96%
spotPython tuning: 0.0196078431372549 [###-----] 38.52%
spotPython tuning: 0.0196078431372549 [###-----] 40.15%
spotPython tuning: 0.0196078431372549 [###-----] 41.71%
spotPython tuning: 0.0196078431372549 [###-----] 43.44%
spotPython tuning: 0.0196078431372549 [#####----] 45.04%
spotPython tuning: 0.0196078431372549 [#####----] 46.62%
spotPython tuning: 0.0196078431372549 [#####----] 48.19%
spotPython tuning: 0.0196078431372549 [#####----] 49.75%
spotPython tuning: 0.0196078431372549 [#####----] 51.37%
spotPython tuning: 0.0196078431372549 [#####----] 52.97%
spotPython tuning: 0.0196078431372549 [#####----] 54.56%
spotPython tuning: 0.0196078431372549 [#####----] 56.06%
spotPython tuning: 0.0196078431372549 [#####----] 57.57%
spotPython tuning: 0.0196078431372549 [#####----] 59.10%
spotPython tuning: 0.0196078431372549 [#####----] 60.58%
spotPython tuning: 0.0196078431372549 [#####----] 61.83%
spotPython tuning: 0.0196078431372549 [#####----] 63.15%
spotPython tuning: 0.0196078431372549 [#####----] 64.59%
spotPython tuning: 0.0196078431372549 [#####----] 66.09%
spotPython tuning: 0.0196078431372549 [#####----] 67.62%
spotPython tuning: 0.0196078431372549 [#####----] 68.77%
spotPython tuning: 0.0196078431372549 [#####----] 69.97%
spotPython tuning: 0.0196078431372549 [#####----] 71.04%
spotPython tuning: 0.0196078431372549 [#####----] 72.11%
spotPython tuning: 0.0196078431372549 [#####----] 73.60%
spotPython tuning: 0.0196078431372549 [#####----] 74.62%
```

```

spotPython tuning: 0.0196078431372549 [#####---] 76.18%
spotPython tuning: 0.0196078431372549 [#####---] 77.32%
spotPython tuning: 0.0196078431372549 [#####---] 78.39%
spotPython tuning: 0.0196078431372549 [#####---] 79.49%
spotPython tuning: 0.0196078431372549 [#####---] 80.68%
spotPython tuning: 0.0196078431372549 [#####---] 81.75%
spotPython tuning: 0.0196078431372549 [#####---] 83.27%
spotPython tuning: 0.0196078431372549 [#####---] 84.29%
spotPython tuning: 0.0196078431372549 [#####---] 85.58%
spotPython tuning: 0.0196078431372549 [#####---] 86.92%
spotPython tuning: 0.0196078431372549 [#####---] 88.21%
spotPython tuning: 0.0196078431372549 [#####---] 89.65%
spotPython tuning: 0.0196078431372549 [#####---] 90.96%
spotPython tuning: 0.0196078431372549 [#####---] 92.34%
spotPython tuning: 0.0196078431372549 [#####---] 93.87%
spotPython tuning: 0.0196078431372549 [#####---] 95.25%
spotPython tuning: 0.0196078431372549 [#####---] 96.79%
spotPython tuning: 0.0196078431372549 [#####---] 98.12%
spotPython tuning: 0.0196078431372549 [#####---] 99.38%
spotPython tuning: 0.0196078431372549 [#####---] 100.00% Done...

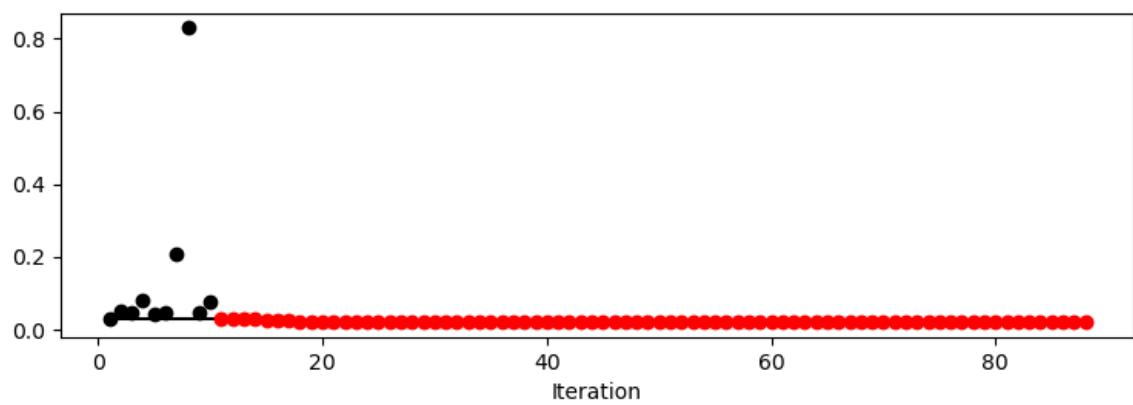
```

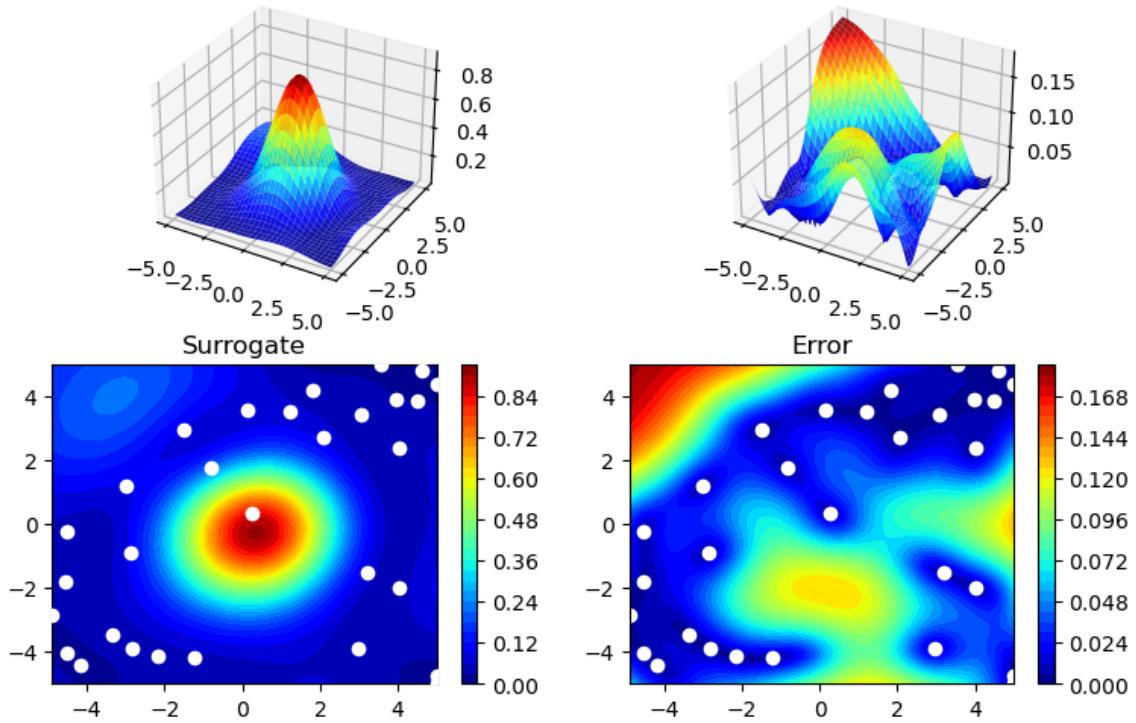
```

{'CHECKPOINT_PATH': 'runs/saved_models/',
'DATASET_PATH': 'data/',
'PREFIX': '003',
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,
'core_model_name': None,
'counter': 88,
'data': None,
'data_dir': './data',
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': inf,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([-5, -5]),
'max_surrogate_points': 30,
'max_time': 0.4,
'metric_params': {},
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
}

```

```
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/003_c03-b01_2024-05-27_16-42-0
1',
'spot_writer': <torch.utils.tensorboard.SummaryWriter object at
0x148995ef64d0>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([5, 5]),
'vername': None,
'vertype': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 0.0196078431372549
x0: 5.0
x1: 5.0
```





Out[6]: array([-0.80725198])

```
In [7]: #Anisotropic Runge Function
surrogate_control = surrogate_control_init(n_theta=2)
spot_2_anisotropic = spot.Spot(fun=fun,
                               fun_control=fun_control,
                               surrogate_control=surrogate_control)
spot_2_anisotropic.run()
spot_2_anisotropic.print_results()
spot_2_anisotropic.surrogate.plot()
spot_2_anisotropic.surrogate.theta
```

```

spotPython tuning: 0.028397462483452193 [-----] 1.05%
spotPython tuning: 0.028397462483452193 [-----] 2.10%
spotPython tuning: 0.028397462483452193 [-----] 3.16%
spotPython tuning: 0.021438173065031263 [-----] 4.39%
spotPython tuning: 0.021438173065031263 [#-----] 5.63%
spotPython tuning: 0.021438173065031263 [#-----] 7.01%
spotPython tuning: 0.0196078431372549 [#-----] 9.30%
spotPython tuning: 0.0196078431372549 [#-----] 11.05%
spotPython tuning: 0.0196078431372549 [#-----] 12.81%
spotPython tuning: 0.0196078431372549 [#-----] 14.37%
spotPython tuning: 0.0196078431372549 [##-----] 16.24%
spotPython tuning: 0.0196078431372549 [##-----] 17.56%
spotPython tuning: 0.0196078431372549 [##-----] 19.35%
spotPython tuning: 0.0196078431372549 [##-----] 20.70%
spotPython tuning: 0.0196078431372549 [##-----] 21.99%
spotPython tuning: 0.0196078431372549 [##-----] 23.94%
spotPython tuning: 0.0196078431372549 [###-----] 25.60%
spotPython tuning: 0.0196078431372549 [###-----] 27.38%
spotPython tuning: 0.0196078431372549 [###-----] 28.55%
spotPython tuning: 0.0196078431372549 [###-----] 29.98%
spotPython tuning: 0.0196078431372549 [###-----] 31.32%
spotPython tuning: 0.0196078431372549 [###-----] 33.27%
spotPython tuning: 0.0196078431372549 [####-----] 35.30%
spotPython tuning: 0.0196078431372549 [####-----] 36.95%
spotPython tuning: 0.0196078431372549 [####-----] 38.58%
spotPython tuning: 0.0196078431372549 [####-----] 39.82%
spotPython tuning: 0.0196078431372549 [####-----] 41.59%
spotPython tuning: 0.0196078431372549 [####-----] 43.85%
spotPython tuning: 0.0196078431372549 [#####-----] 45.40%
spotPython tuning: 0.0196078431372549 [#####-----] 47.43%
spotPython tuning: 0.0196078431372549 [#####-----] 49.94%
spotPython tuning: 0.0196078431372549 [#####-----] 51.68%
spotPython tuning: 0.0196078431372549 [#####-----] 53.38%
spotPython tuning: 0.0196078431372549 [#####-----] 55.18%
spotPython tuning: 0.0196078431372549 [#####-----] 56.87%
spotPython tuning: 0.0196078431372549 [#####-----] 58.90%
spotPython tuning: 0.0196078431372549 [#####-----] 61.85%
spotPython tuning: 0.0196078431372549 [#####-----] 64.00%
spotPython tuning: 0.0196078431372549 [#####-----] 66.66%
spotPython tuning: 0.0196078431372549 [#####-----] 68.60%
spotPython tuning: 0.0196078431372549 [#####-----] 71.04%
spotPython tuning: 0.0196078431372549 [#####-----] 73.00%
spotPython tuning: 0.0196078431372549 [#####-----] 74.88%
spotPython tuning: 0.0196078431372549 [#####-----] 76.46%
spotPython tuning: 0.0196078431372549 [#####-----] 78.68%
spotPython tuning: 0.0196078431372549 [#####-----] 80.34%
spotPython tuning: 0.0196078431372549 [#####-----] 81.98%
spotPython tuning: 0.0196078431372549 [#####-----] 83.47%
spotPython tuning: 0.0196078431372549 [#####-----] 85.47%
spotPython tuning: 0.0196078431372549 [#####-----] 87.50%
spotPython tuning: 0.0196078431372549 [#####-----] 89.45%
spotPython tuning: 0.0196078431372549 [#####-----] 91.51%
spotPython tuning: 0.0196078431372549 [#####-----] 93.68%
spotPython tuning: 0.0196078431372549 [#####-----] 95.37%
spotPython tuning: 0.0196078431372549 [#####-----] 97.48%
spotPython tuning: 0.0196078431372549 [#####-----] 99.02%
spotPython tuning: 0.0196078431372549 [#####-----] 100.00% Done...

```

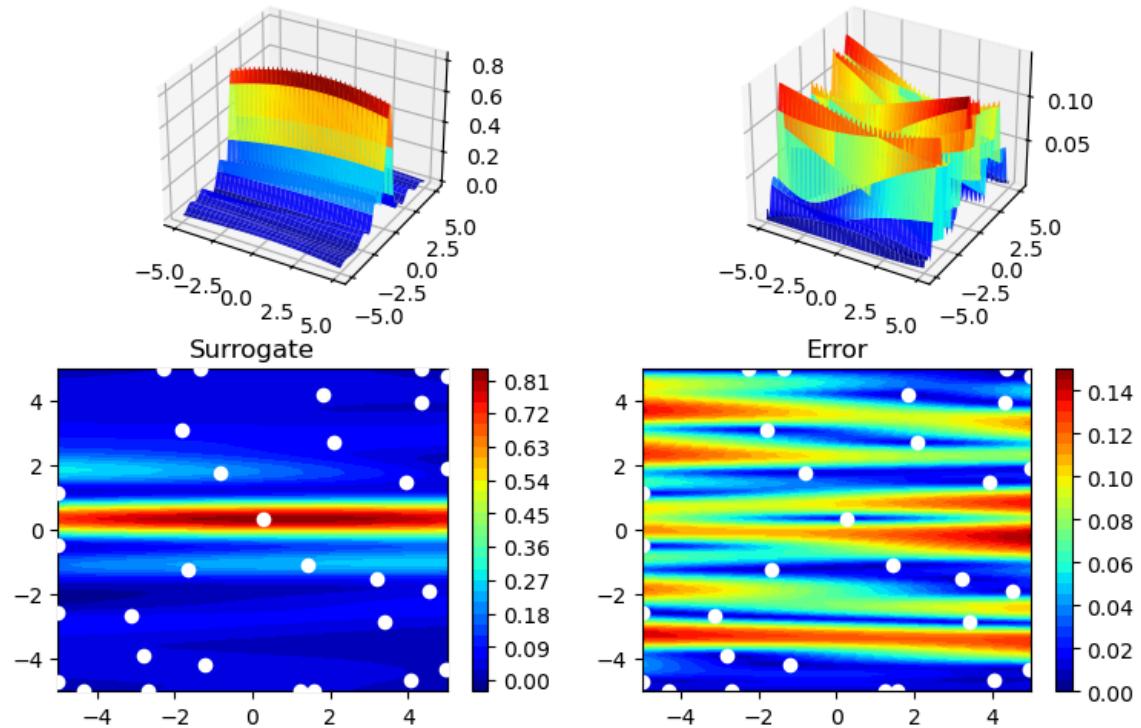
```

{'CHECKPOINT_PATH': 'runs/saved_models/',
'DATASET_PATH': 'data',

```

```
'PREFIX': '003',
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,
'core_model_name': None,
'counter': 67,
'data': None,
'data_dir': './data',
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': inf,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([-5, -5]),
'max_surrogate_points': 30,
'max_time': 0.4,
'metric_params': {},
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/003_c03-b01_2024-05-27_16-42-0
1',
```

```
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x148995ef64d0>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([5, 5]),
'var_name': None,
'var_type': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 0.0196078431372549
x0: 5.0
x1: 5.0
```



Out[7]: array([-2.31090925, 0.43464023])

Result

Function	n_theta	max_time(min)	Number of Evaluations	Number of Iterations	Objective function Value
Runge Function	1	1	inf	202	0.0196078431372549
Runge Function	2	1	inf	138	0.0196078431372549
Runge Function	1	0.4	inf	88	0.0196078431372549

Function	n_theta	max_time(min)	Number of Evaluations	Number of Iterations	Objective function Value
Runge Function	2	0.4	inf	67	0.0196078431372549

Increasing `n_theta` from 1 to 2 reduces the number of iterations required to reach the optimal objective function value for the Runge function, from 202 to 138 iterations for `max_time` set to 1 minute and from 88 to 67 iterations for `max_time` set to 0.4 minutes. This indicates that a higher `n_theta` improves optimization efficiency without affecting the quality of the solution.

4. The Ten-dimensional Wing-Weight Function `fun_wingwt`

- Describe the function.
 - * Example from Forrester et al. to understand the weight of an unpainted light aircraft wing as a function of nine design and operational parameters:

$$W = 0.036 S \cdot 0.758 \cdot W_{fw} \cdot 0.0035 (A / (\cos^2 \Lambda))^{0.6} \cdot q \cdot 0.006 \cdot \lambda^{0.04} \cdot ((100 R_{tc}) / (\cos \Lambda)) - 0.3 \cdot (N_z W_{dg})^{0.49}$$
 - The input dimension is `10`. The search ranges are between 0 and 1 (values are mapped internally to their natural bounds).
- Compare the results from `spotPython` run a) with isotropic and b) anisotropic surrogate models.
- Modify the termination criterion (`max_time` instead of `fun_evals`) as described for `fun_branin`.

Isotropic and Anisotropic with Time = 1 min

```
In [8]: #Isotropic Wing-Weight Function
PREFIX="003"
fun = analytical().fun_wingwt

fun_control = fun_control_init(PREFIX=PREFIX,
                               fun_evals=inf,
                               max_time=1,
                               lower = np.array([0,0,0,0,0,0,0,0,0,0]),
                               upper = np.array([1,1,1,1,1,1,1,1,1,1]))
surrogate_control=surrogate_control_init(n_theta=1)
spot_2 = spot.Spot(fun=fun,
                   fun_control=fun_control,
                   surrogate_control=surrogate_control)

spot_2.run()
spot_2.print_results()
spot_2.plot_progress()
spot_2.surrogate.plot()
spot_2.surrogate.theta
```

[rank: 0] Seed set to 123

```

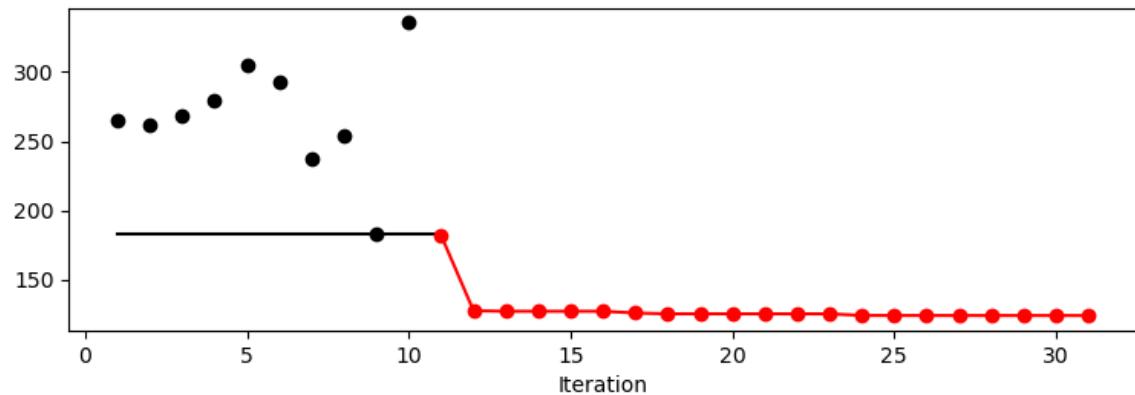
Created spot_tensorboard_path: runs/spot_logs/003_c02-b04_2024-05-21_13-32-03 for SummaryWriter()
spotPython tuning: 181.85467170304693 [-----] 2.05%
spotPython tuning: 127.42305098146872 [#-----] 7.06%
spotPython tuning: 126.99585019636925 [#-----] 11.14%
spotPython tuning: 126.99582536475548 [#-----] 14.76%
spotPython tuning: 126.99582536475548 [##-----] 19.23%
spotPython tuning: 126.99582536475548 [##-----] 23.80%
spotPython tuning: 125.75754761859794 [###-----] 28.11%
spotPython tuning: 125.19270696347755 [###-----] 33.05%
spotPython tuning: 125.19270696347755 [####-----] 38.41%
spotPython tuning: 125.16786252715022 [####-----] 43.93%
spotPython tuning: 125.15079569005242 [#####-----] 48.92%
spotPython tuning: 125.14790023857168 [#####-----] 54.75%
spotPython tuning: 125.14790023857168 [#####-----] 60.09%
spotPython tuning: 123.97787541186656 [#####-----] 65.95%
spotPython tuning: 123.95017737468233 [#####-----] 70.81%
spotPython tuning: 123.95017737468233 [#####-----] 76.00%
spotPython tuning: 123.9249382999715 [#####-----] 81.00%
spotPython tuning: 123.9249382999715 [#####-----] 85.63%
spotPython tuning: 123.9249382999715 [#####-----] 90.99%
spotPython tuning: 123.9249382999715 [#####-----] 97.50%
spotPython tuning: 123.89321921573641 [#####-----] 100.00% Done...

```

```
{
'CHECKPOINT_PATH': 'runs/saved_models/',
'DATASET_PATH': 'data/',
'PREFIX': '003',
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,
'core_model_name': None,
'counter': 31,
'data': None,
'data_dir': './data',
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': inf,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0]),
'max_surrogate_points': 30,
'max_time': 1,
'metric_params': {}}

```

```
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/003_c02-b04_2024-05-21_13-32-0
3',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x154d38795e10>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1]),
'vername': None,
'ver_type': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 123.89321921573641
x0: 0.0
x1: 0.7896824235413803
x2: 0.0
x3: 0.5996766165807106
x4: 0.5067161564417725
x5: 0.0
x6: 1.0
x7: 0.0
x8: 0.0
x9: 0.0
```



Out [8]: array([-0.5573158])

In [9]: *#Anisotropic Wing-Weight Function*
surrogate_control = surrogate_control_init(n_theta=2)
spot_2_anisotropic = spot.Spot(fun=fun,
 fun_control=fun_control,
 surrogate_control=surrogate_control)
spot_2_anisotropic.run()
spot_2_anisotropic.print_results()
spot_2_anisotropic.surrogate.plot()
spot_2_anisotropic.surrogate.theta

```

spotPython tuning: 130.98167265774384 [-----] 4.62%
spotPython tuning: 124.96150746789587 [#-----] 14.06%
spotPython tuning: 123.38367978743302 [##-----] 23.02%
spotPython tuning: 123.38367978743302 [###-----] 29.88%
spotPython tuning: 123.38367978743302 [####-----] 35.69%
spotPython tuning: 123.38346849453022 [#####-----] 44.55%
spotPython tuning: 123.38346849423637 [#####-----] 52.47%
spotPython tuning: 123.38346849423637 [#####-----] 60.56%
spotPython tuning: 123.383468440801 [#####-----] 66.06%
spotPython tuning: 123.38346844053387 [#####-----] 72.13%
spotPython tuning: 123.38346842534199 [#####-----] 78.46%
spotPython tuning: 123.38346842534199 [#####-----] 85.51%
spotPython tuning: 123.38346842534199 [#####-----] 93.36%
spotPython tuning: 123.38346842534199 [#####-----] 100.00% Done...

```

```

{'CHECKPOINT_PATH': 'runs/saved_models/',
 'DATASET_PATH': 'data/',
 'PREFIX': '003',
 'RESULTS_PATH': 'results/',
 'TENSORBOARD_PATH': 'runs/',
 '_L_in': None,
 '_L_out': None,
 '_torchmetric': None,
 'accelerator': 'auto',
 'converters': None,
 'core_model': None,
 'core_model_name': None,
 'counter': 24,
 'data': None,
 'data_dir': './data',
 'data_module': None,
 'data_set': None,
 'data_set_name': None,
 'db_dict_name': None,
 'design': None,
 'device': None,
 'devices': 1,
 'enable_progress_bar': False,
 'eval': None,
 'fun_evals': inf,
 'fun_repeats': 1,
 'horizon': None,
 'infill_criterion': 'y',
 'k_folds': 3,
 'log_graph': False,
 'log_level': 50,
 'loss_function': None,
 'lower': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0]),
 'max_surrogate_points': 30,
 'max_time': 1,
 'metric_params': {},
 'metric_river': None,
 'metric_sklearn': None,
 'metric_sklearn_name': None,
 'metric_torch': None,
 'model_dict': {},
 'n_points': 1,
 'n_samples': None,
 'n_total': None,
 'noise': False,
}
```

```

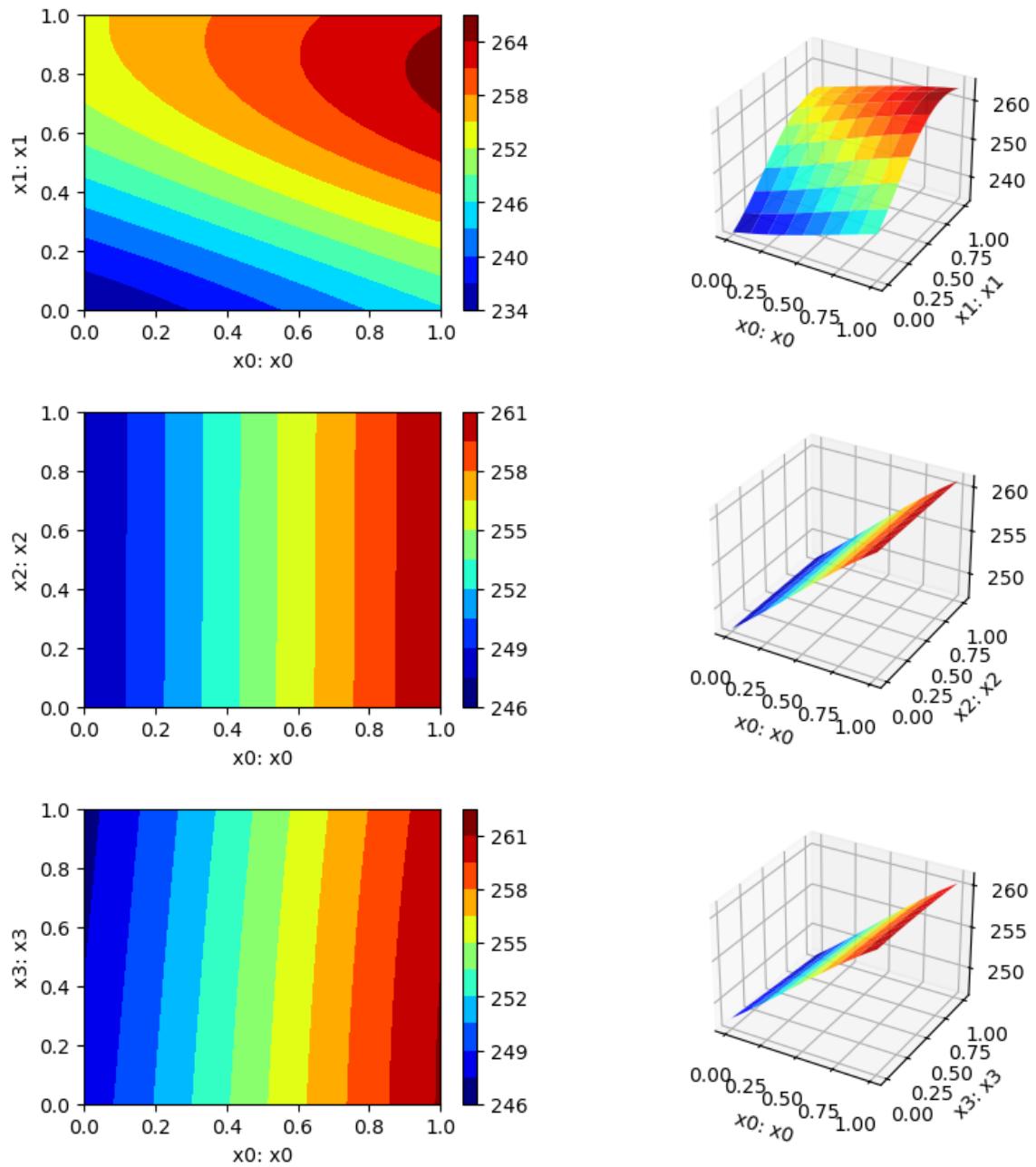
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/003_c02-b04_2024-05-21_13-32-0
3',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x154d38795e10>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1]),
'ver': None,
'ver_type': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 123.38346842534199
x0: 0.0
x1: 1.0
x2: 0.0
x3: 0.49997968170323087
x4: 0.0
x5: 0.0
x6: 1.0
x7: 0.0
x8: 0.0
x9: 0.0
Out[9]: array([-0.64145565,  0.12836716, -2.4279085 , -1.35010903, -2.99706243,
   -1.74992045, -0.89388034,  1.3560256 , -2.59696482, -2.99331943])

```

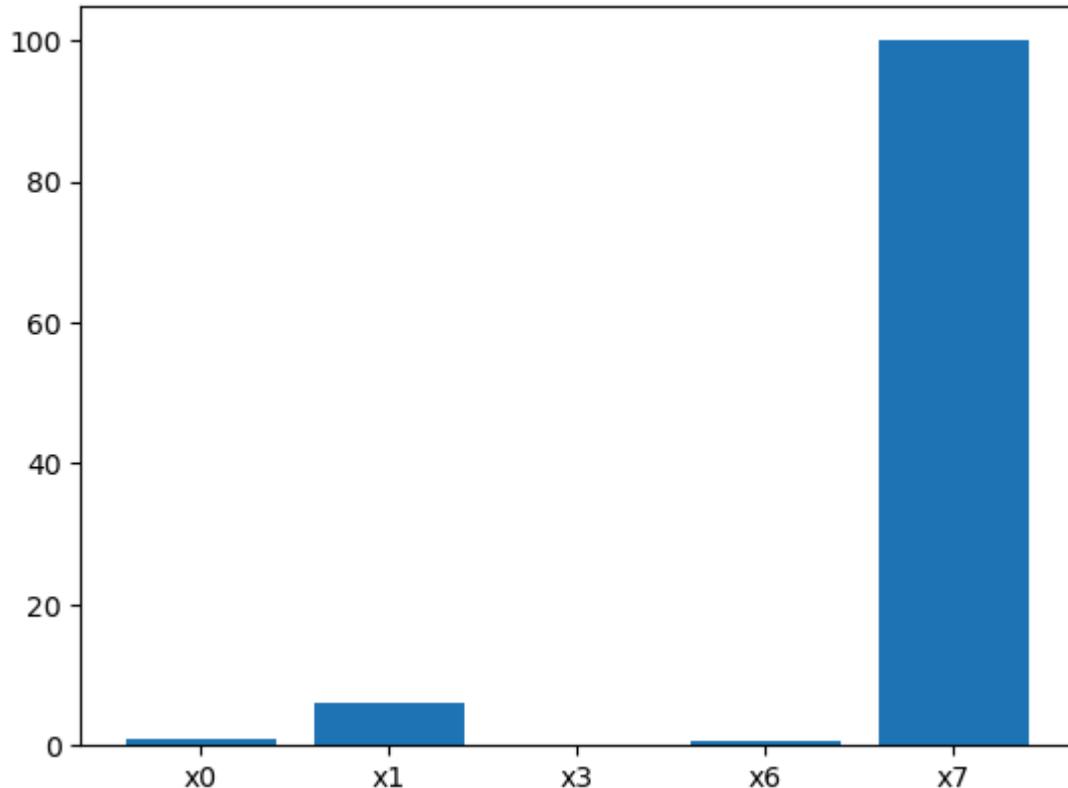
```

In [10]: min_z = None
max_z = None
spot_2_anisotropic.plot_contour(i=0, j=1, min_z=min_z, max_z=max_z)
spot_2_anisotropic.plot_contour(i=0, j=2, min_z=min_z, max_z=max_z)
spot_2_anisotropic.plot_contour(i=0, j=3, min_z=min_z, max_z=max_z)
_= spot_2_anisotropic.print_importance()
spot_2_anisotropic.plot_importance()

```



$x_0: 1.0058164788818371$
 $x_1: 5.920270576451609$
 $x_3: 0.19672763310138394$
 $x_6: 0.5624631299436303$
 $x_7: 99.99999999999999$



Isotropic and Anisotropic with time = 0.4 min

```
In [8]: #Isotropic Wing-Weight Function
PREFIX="003"
fun = analytical().fun_wingwt

fun_control = fun_control_init(PREFIX=PREFIX,
                                fun_evals=inf,
                                max_time=0.4,
                                lower = np.array([0,0,0,0,0,0,0,0,0,0]),
                                upper = np.array([1,1,1,1,1,1,1,1,1,1]))
surrogate_control=surrogate_control_init(n_theta=1)
spot_2 = spot.Spot(fun=fun,
                    fun_control=fun_control,
                    surrogate_control=surrogate_control)

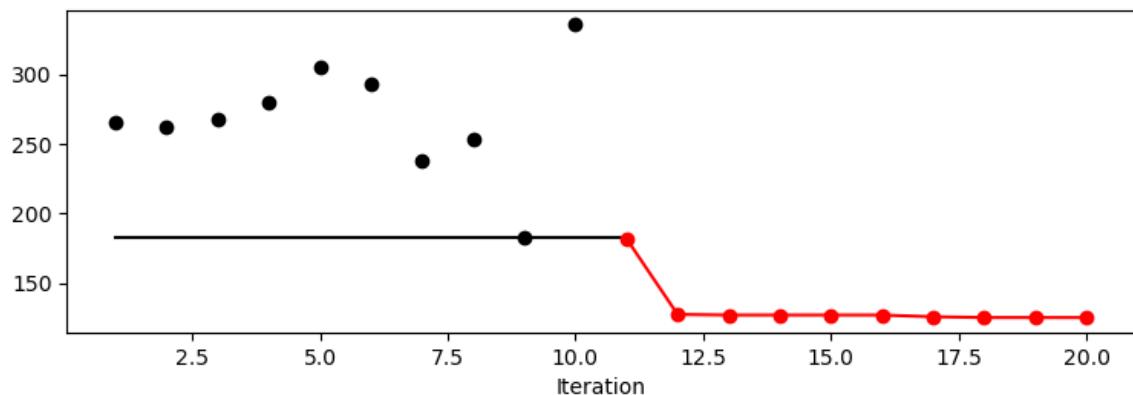
spot_2.run()
spot_2.print_results()
spot_2.plot_progress()
spot_2.surrogate.plot()
spot_2.surrogate.theta
```

[rank: 0] Seed set to 123

```
Created spot_tensorboard_path: runs/spot_logs/003_c03-b01_2024-05-27_16-55-18 for SummaryWriter()
spotPython tuning: 181.85467170304693 [-----] 5.17%
spotPython tuning: 127.42305098146872 [##-----] 17.09%
spotPython tuning: 126.99585019636925 [###-----] 26.74%
spotPython tuning: 126.99582536475548 [####-----] 35.28%
spotPython tuning: 126.99582536475548 [#####-----] 46.21%
spotPython tuning: 126.99582536475548 [#####-----] 57.06%
spotPython tuning: 125.75754761859794 [#####-----] 67.23%
spotPython tuning: 125.19270696347755 [#####-----] 78.71%
spotPython tuning: 125.19270696347755 [#####-----] 90.92%
spotPython tuning: 125.16786252715022 [#####-----] 100.00% Done...
```

```
{'CHECKPOINT_PATH': 'runs/saved_models/',
'DATASET_PATH': 'data/',
'PREFIX': '003',
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,
'core_model_name': None,
'counter': 20,
'data': None,
'data_dir': './data',
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': inf,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0]),
'max_surrogate_points': 30,
'max_time': 0.4,
'metric_params': {},
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
```

```
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/003_c03-b01_2024-05-27_16-55-18',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x148996581190>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1]),
'vername': None,
'vertype': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 125.16786252715022
x0: 0.0
x1: 0.7142565792017924
x2: 0.0
x3: 0.5738303252720693
x4: 0.7068045840056159
x5: 0.27877019687544374
x6: 1.0
x7: 0.0
x8: 0.0
x9: 0.0
```



Out[8]: array([-1.54038052])

In [9]: #Anisotropic Wing-Weight Function
surrogate_control = surrogate_control_init(n_theta=2)

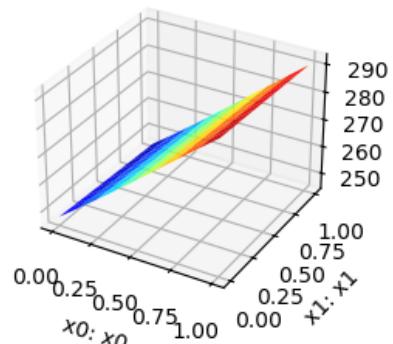
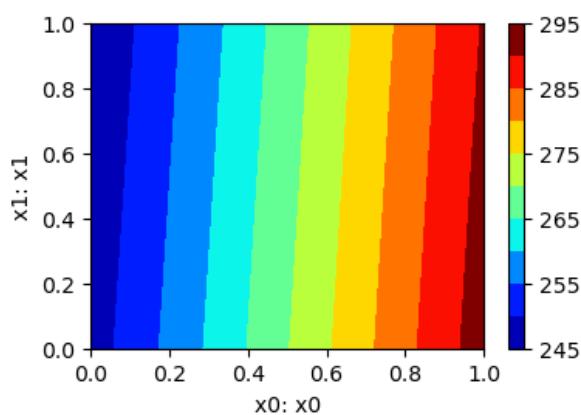
```
spot_2_anisotropic = spot.Spot(fun=fun,
                               fun_control=fun_control,
                               surrogate_control=surrogate_control)
spot_2_anisotropic.run()
spot_2_anisotropic.print_results()
spot_2_anisotropic.surrogate.plot()
spot_2_anisotropic.surrogate.theta

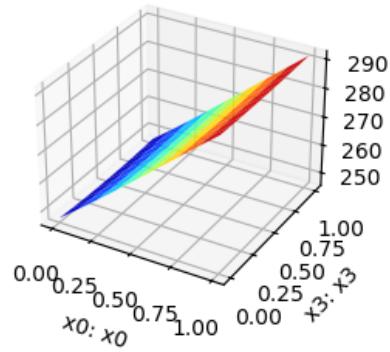
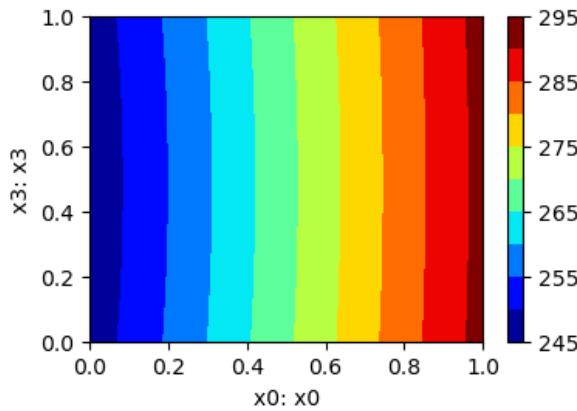
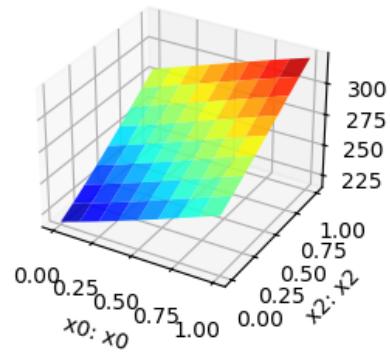
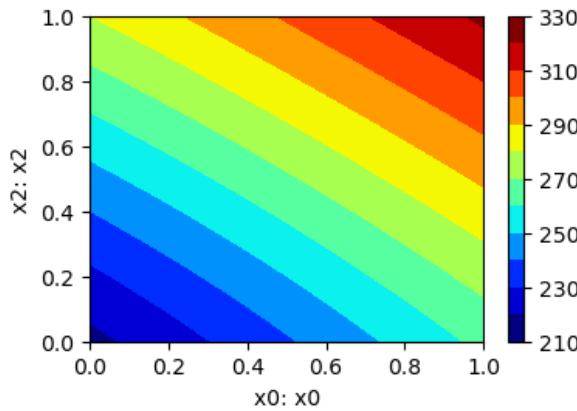
min_z = None
max_z = None
spot_2_anisotropic.plot_contour(i=0, j=1, min_z=min_z, max_z=max_z)
spot_2_anisotropic.plot_contour(i=0, j=2, min_z=min_z, max_z=max_z)
spot_2_anisotropic.plot_contour(i=0, j=3, min_z=min_z, max_z=max_z)
_= spot_2_anisotropic.print_importance()
spot_2_anisotropic.plot_importance()
```

```
spotPython tuning: 130.98167265774384 [-----] 10.92%
spotPython tuning: 124.96150746789587 [####----] 35.09%
spotPython tuning: 123.38367978743302 [#####---] 56.34%
spotPython tuning: 123.38367978743302 [#####---] 72.84%
spotPython tuning: 123.38367978743302 [#####---] 86.74%
spotPython tuning: 123.38346849453022 [#####---] 100.00% Done...

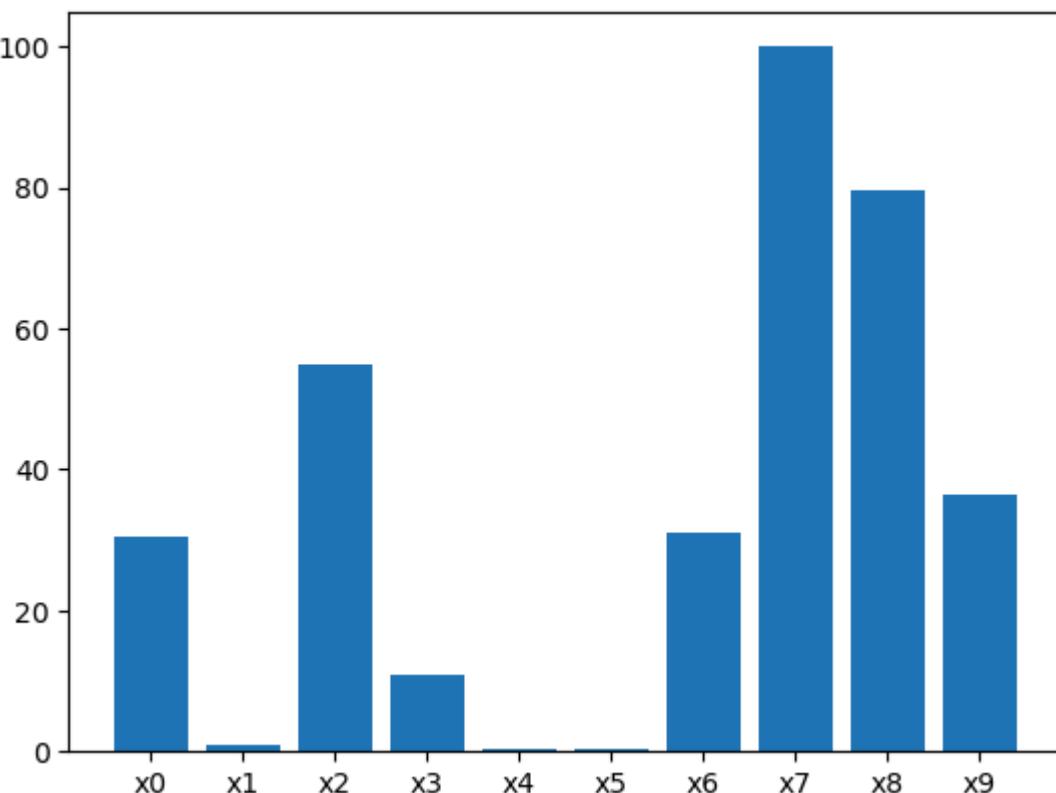
{'CHECKPOINT_PATH': 'runs/saved_models/',
'DATASET_PATH': 'data/',
'PREFIX': '003',
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,
'core_model_name': None,
'counter': 16,
'data': None,
'data_dir': './data',
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': inf,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0]),
'max_surrogate_points': 30,
'max_time': 0.4,
'metric_params': {},
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
```

```
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/003_c03-b01_2024-05-27_16-55-1
8',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x148996581190>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1]),
'vername': None,
'vertype': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 123.38346849453022
x0: 0.0
x1: 1.0
x2: 0.0
x3: 0.49989530805729265
x4: 0.0
x5: 0.0
x6: 1.0
x7: 0.0
x8: 0.0
x9: 0.0
```





```
x0: 30.522852891835818
x1: 0.8192632987778713
x2: 54.87142904802777
x3: 11.01501019716264
x4: 0.3985278032783995
x5: 0.39936548743566913
x6: 31.133763206076143
x7: 100.0
x8: 79.71651781228421
x9: 36.35083758773278
```



Result

Function	n_theta	max_time(min)	Number of Evaluations	Number of Iterations	Objective function Value
Wing-weight Function	1	1	inf	31	123.89321921573641
Wing-weight Function	2	1	inf	24	123.38346842534199
Wing-weight Function	1	0.4	inf	20	125.16786252715022
Wing-weight Function	2	0.4	inf	16	123.38346849453022

The table presents the optimization results for the Wing-weight Function under varying conditions of n_theta (number of parameters) and max_time (maximum allowed time for optimization). It is observed that with n_theta=2, the optimization consistently yields a lower objective function value (123.383468) compared to n_theta=1 under the same max_time conditions. Additionally, reducing the max_time from 1 minute to 0.4 minutes decreases the number of iterations required for convergence, especially for n_theta=2. This suggests that increasing the number of parameters while slightly limiting the optimization time can still achieve efficient convergence with lower objective function values.

Exercises Chapter 10: Using sklearn Surrogates in spotPython

- The following exercises refer to chapter 10 of the book, see https://sequential-parameter-optimization.github.io/Hyperparameter-Tuning-Cookbook/010_num_spot_sklearn_surrogate.html.

In [11]:

```
import numpy as np
from math import inf
from spotPython.fun.objectivefunctions import analytical
from spotPython.utils.init import fun_control_init, design_control_init
from spotPython.spot import spot

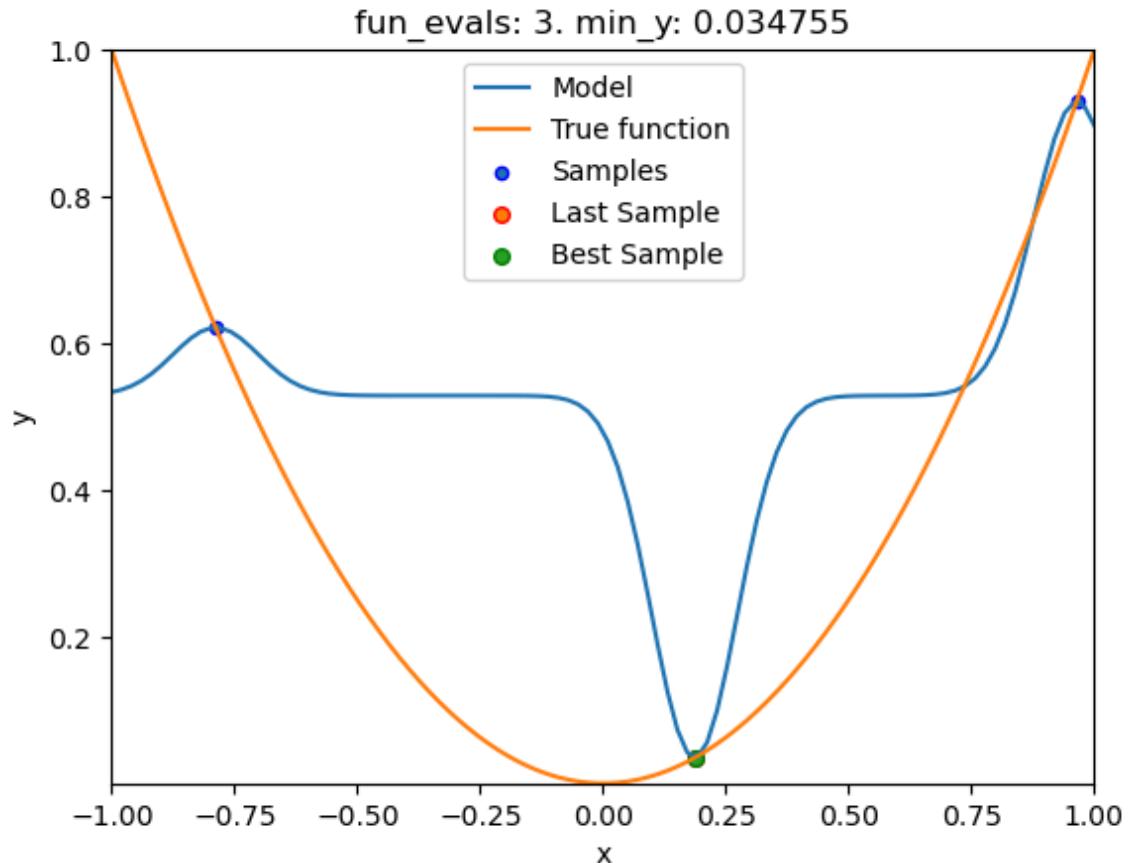
# Needed for the sklearn surrogates:
from sklearn.ensemble import HistGradientBoostingRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
import pandas as pd
S_XGB = HistGradientBoostingRegressor()
S_DTR = DecisionTreeRegressor()
```

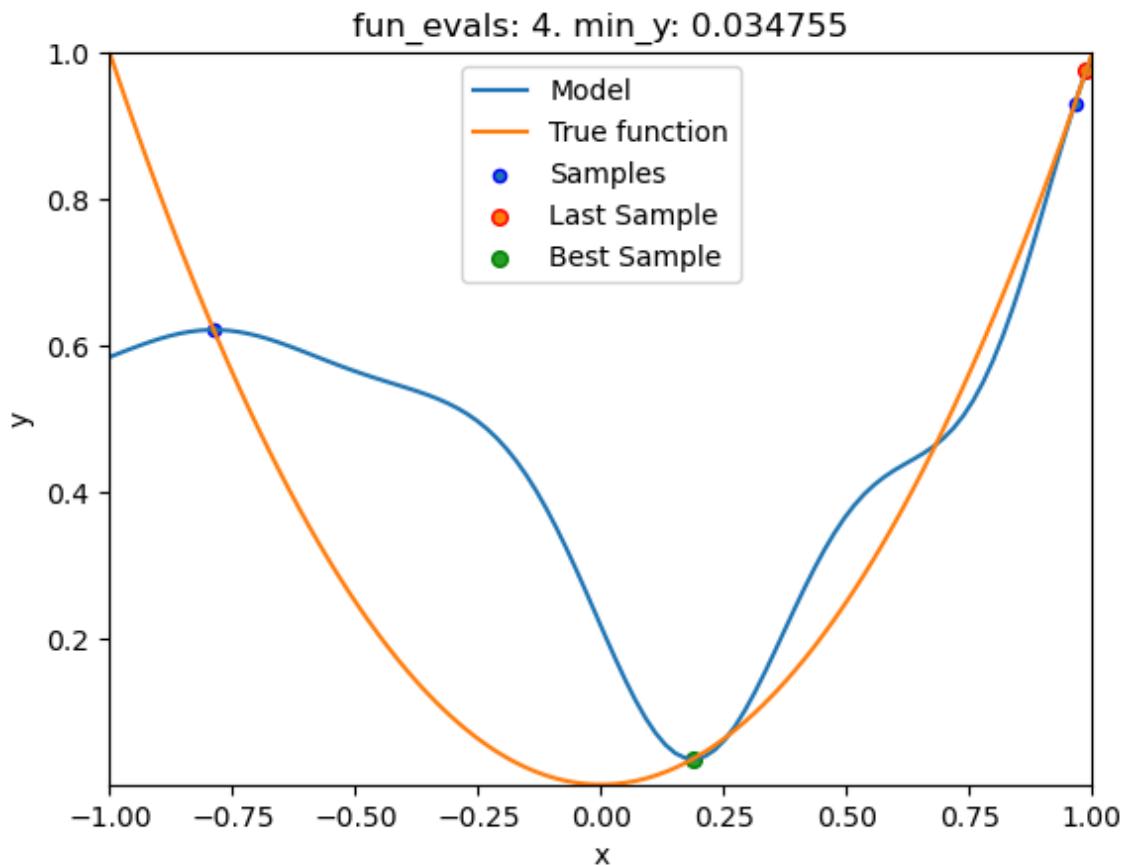
```
S_RFR = RandomForestRegressor()
S_LR = LinearRegression()
S_R = Ridge()
```

1-dim sphere function fun_sphere

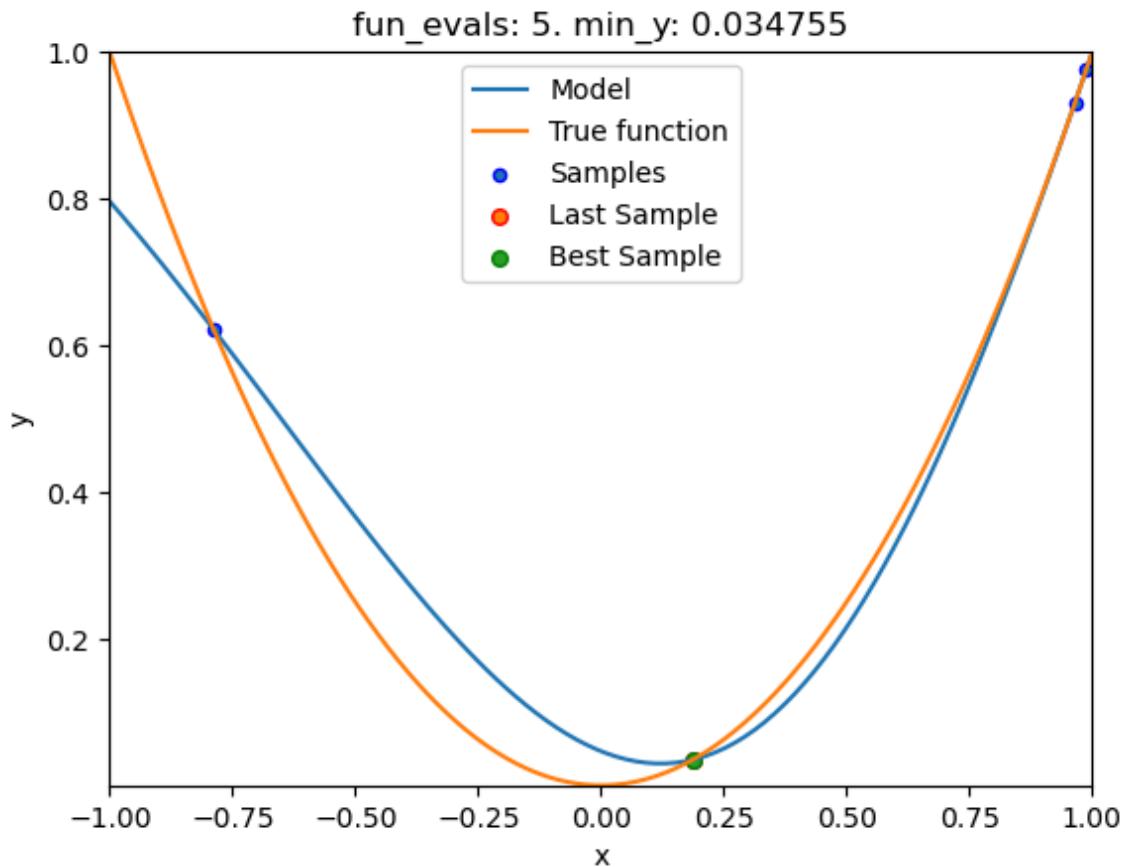
```
In [12]: #1-dim sphere function fun_sphere using spotPython's internal Kriging
from spotPython.fun.objectivefunctions import analytical
fun_control = fun_control_init(
    lower = np.array([-1]),
    upper = np.array([1]),
    fun_evals=10,
    max_time=inf,
    show_models=True,
    tolerance_x = np.sqrt(np.spacing(1)))
fun = analytical(seed=123).fun_sphere
design_control = design_control_init(
    init_size=3)
spot_1 = spot.Spot(fun=fun,
                    fun_control=fun_control,
                    design_control=design_control)
spot_1.run()
spot_1.print_results()
spot_1.plot_progress(log_y=True)
spot_1.plot_model()
```

[rank: 0] Seed set to 123

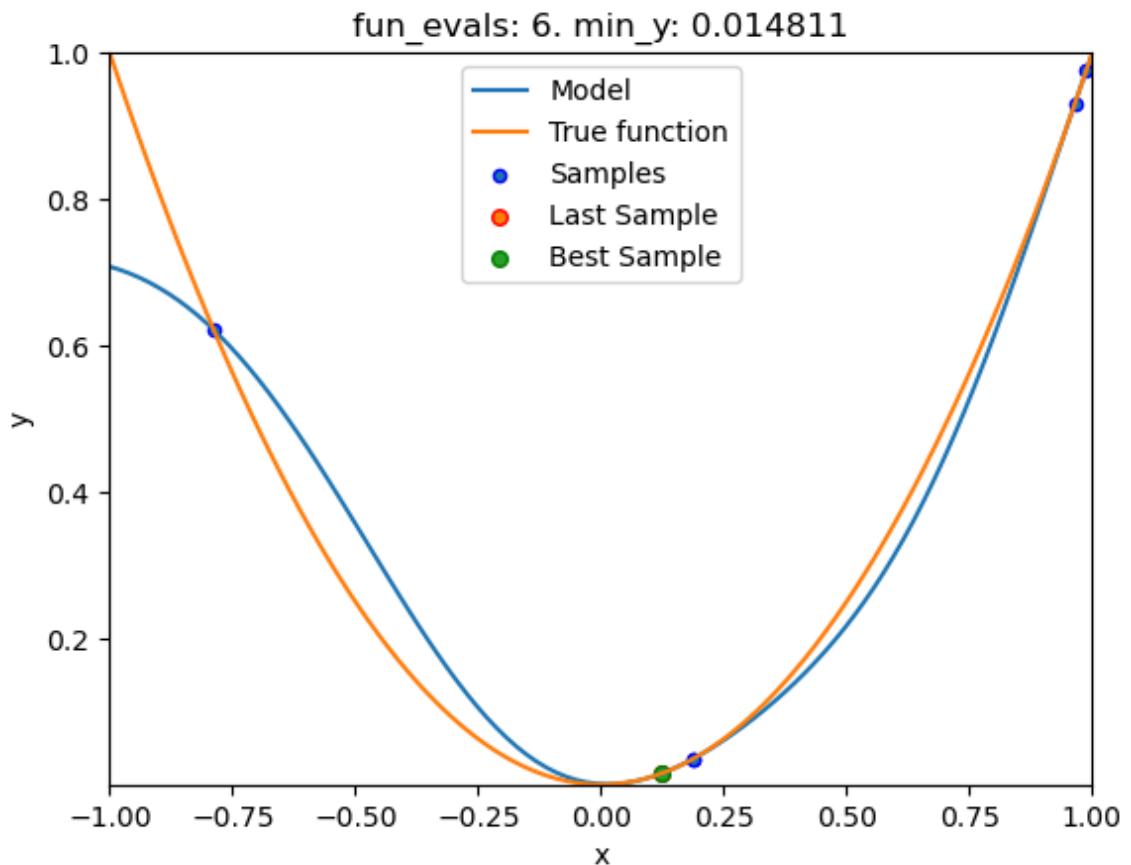




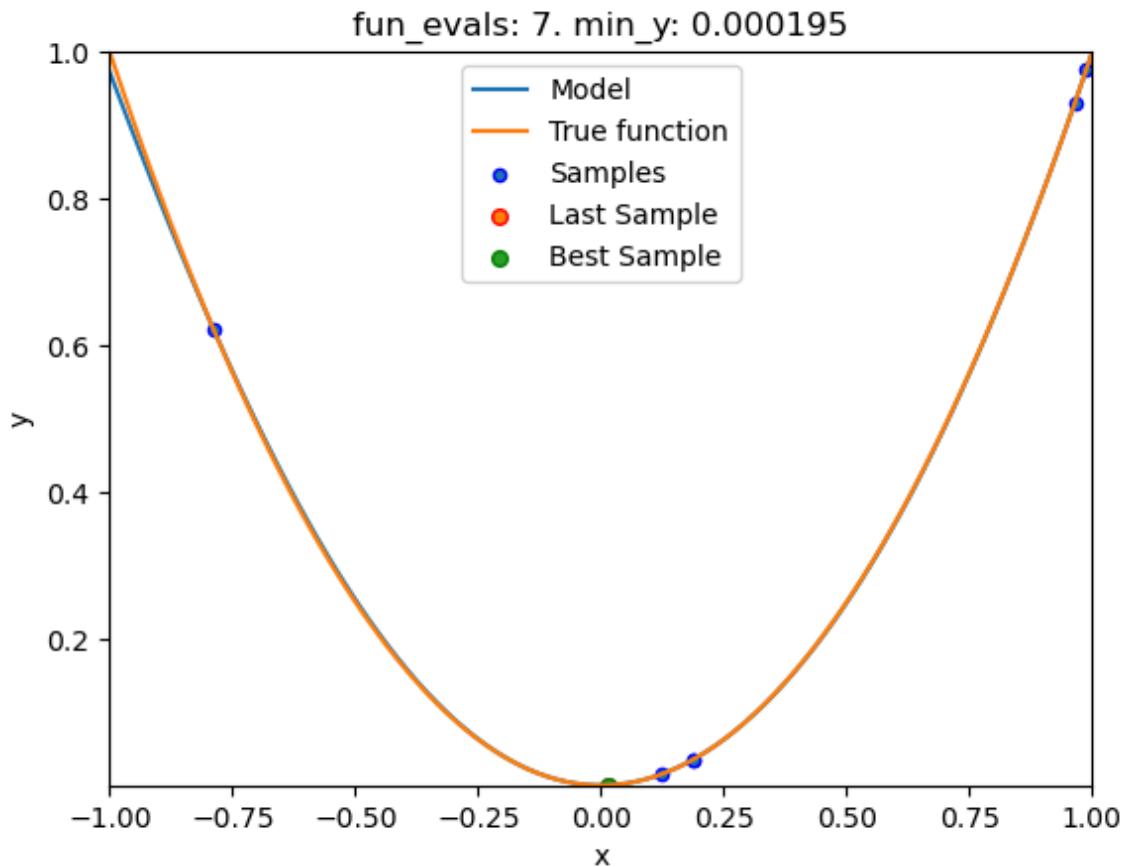
spotPython tuning: 0.03475493366922229 [#####-----] 40.00%



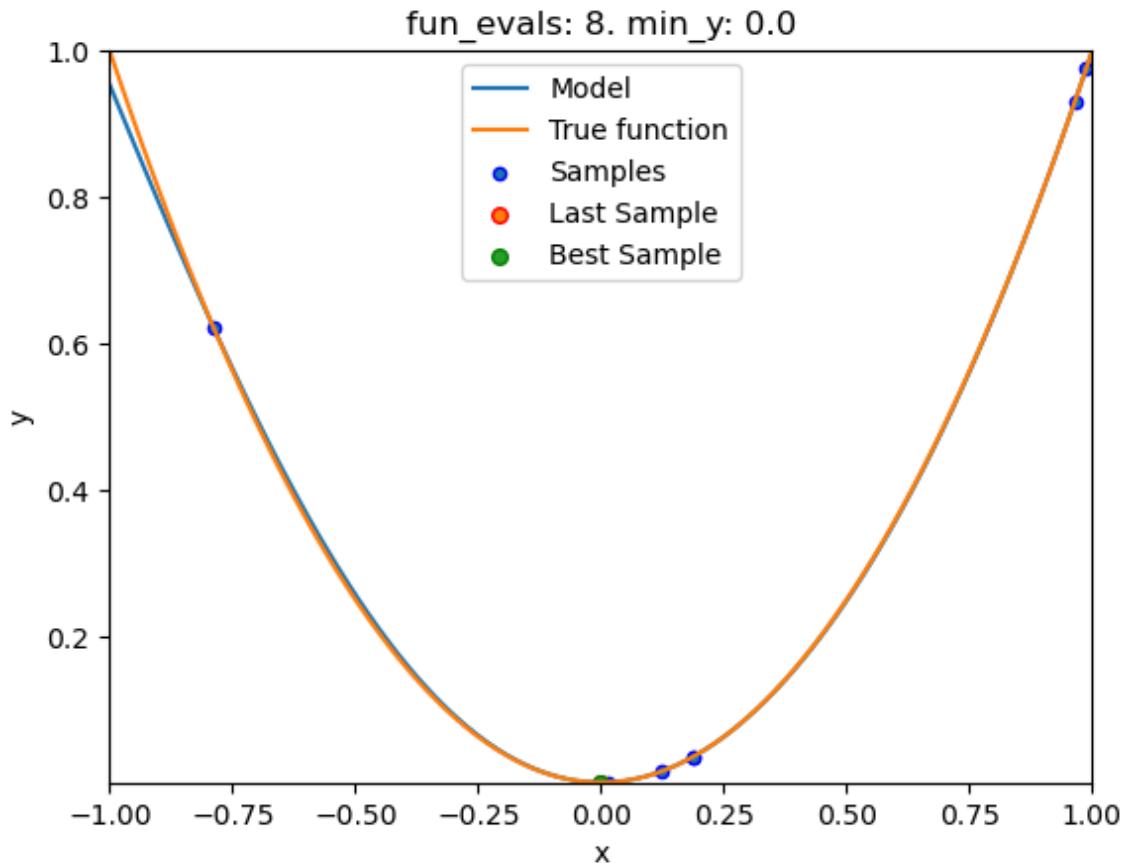
spotPython tuning: 0.03475493366922229 [#####-----] 50.00%



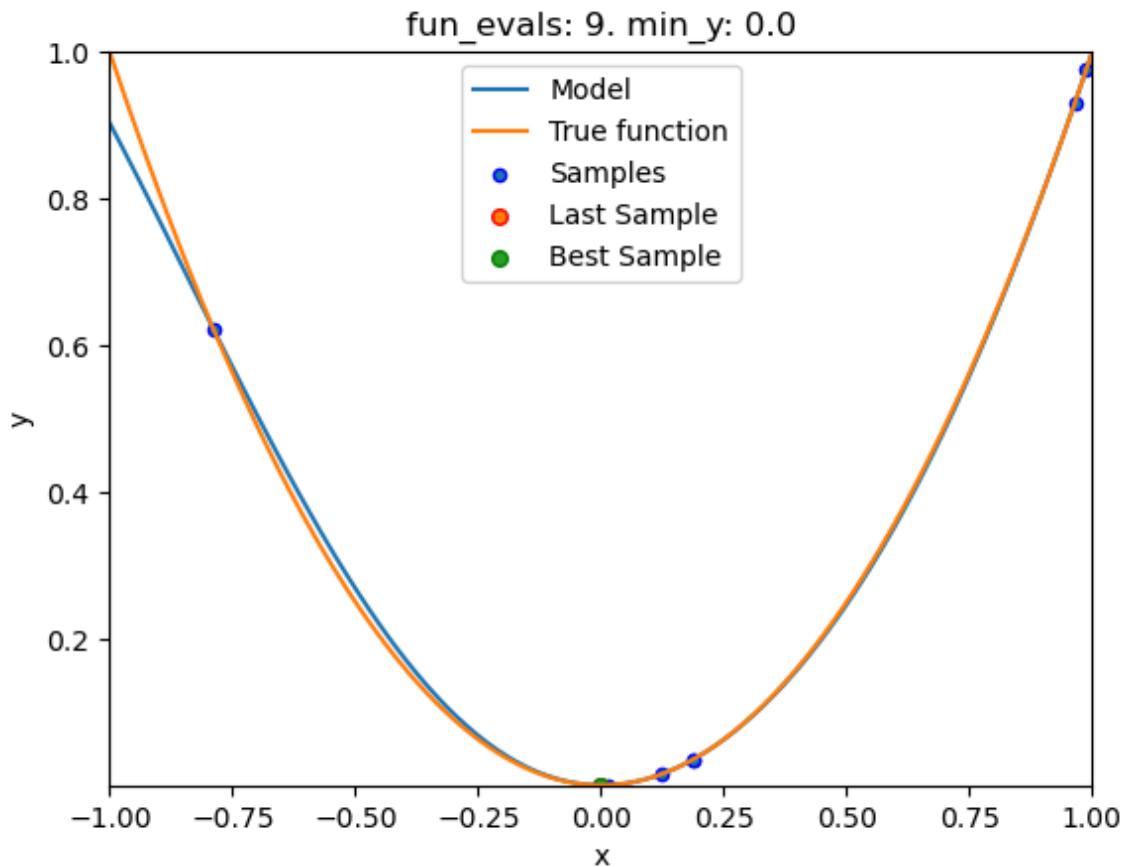
spotPython tuning: 0.014811361922947153 [#####----] 60.00%



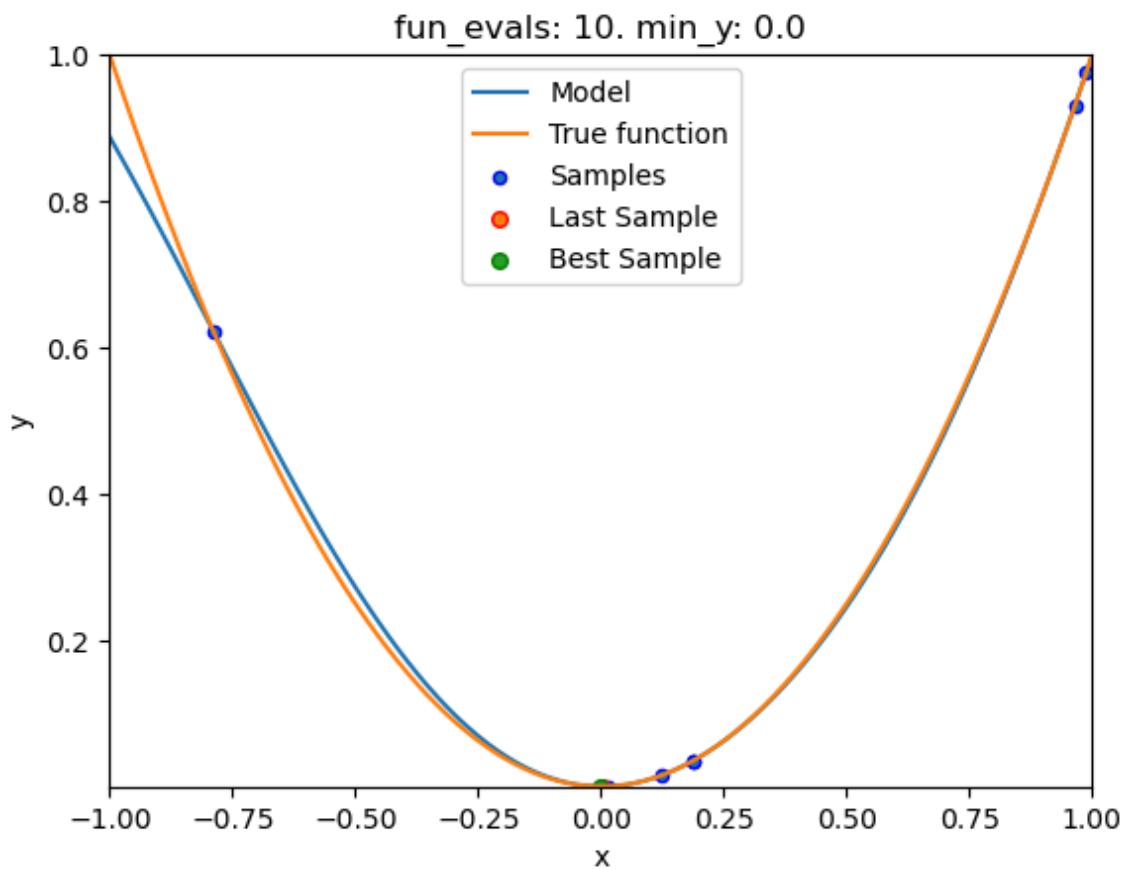
spotPython tuning: 0.00019529794005791534 [#####----] 70.00%



spotPython tuning: 5.4613089651542514e-08 [#####--] 80.00%



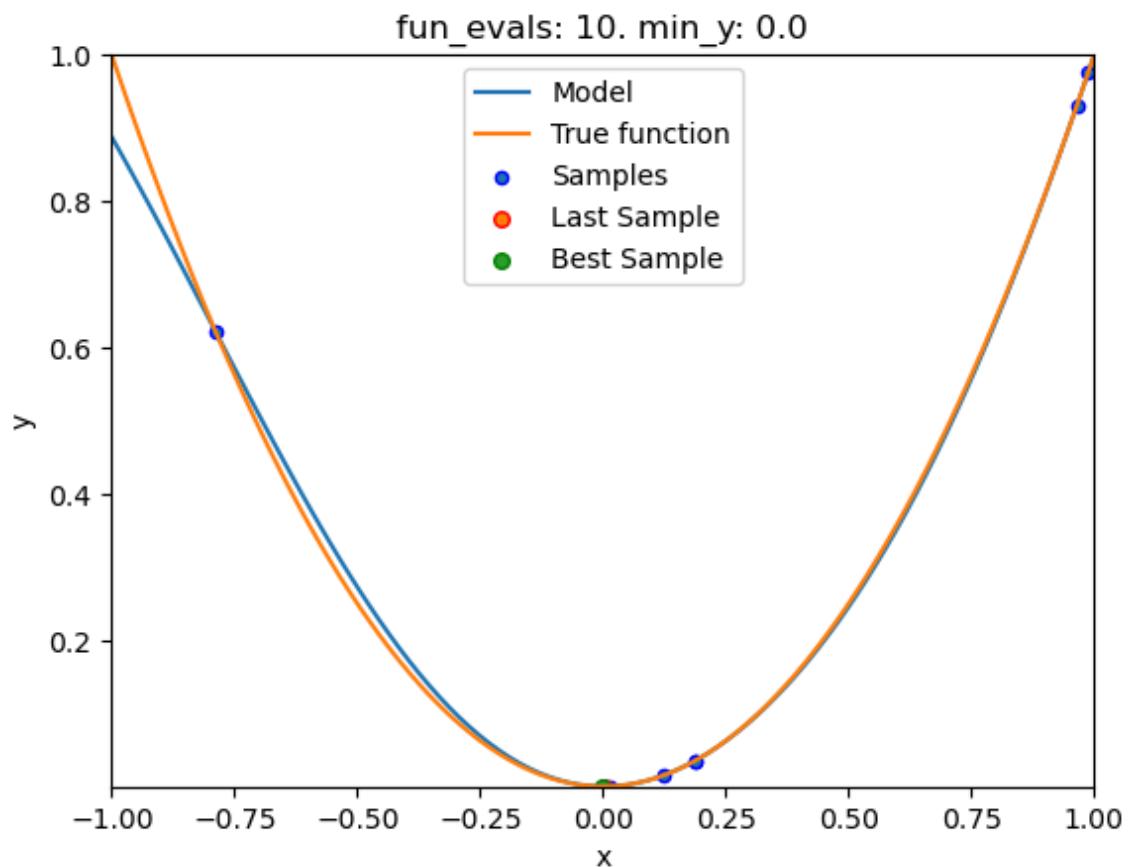
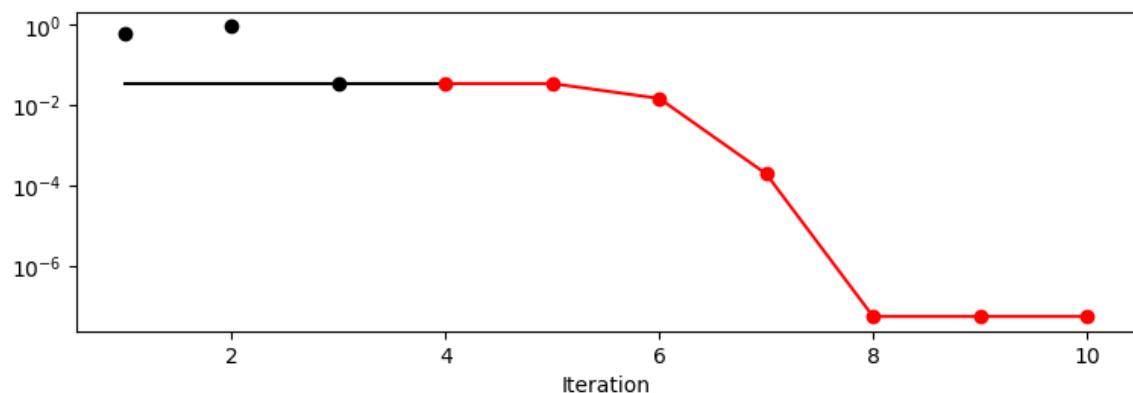
spotPython tuning: 5.4613089651542514e-08 [#####--] 90.00%



spotPython tuning: 5.4613089651542514e-08 [#####] 100.00% Done...

```
{'CHECKPOINT_PATH': 'runs/saved_models/',
'DATASET_PATH': 'data/',
'PREFIX': None,
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,
'core_model_name': None,
'counter': 10,
'data': None,
'data_dir': './data',
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': 10,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([-1]),
'max_surrogate_points': 30,
'max_time': inf,
'metric_params': {},
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': True,
'show_progress': True,
```

```
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': None,
'spot_writer': None,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 1.4901161193847656e-08,
'train': None,
'upper': array([1]),
'vername': None,
'vertype': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 5.4613089651542514e-08
x0: -0.00023369443650104834
```



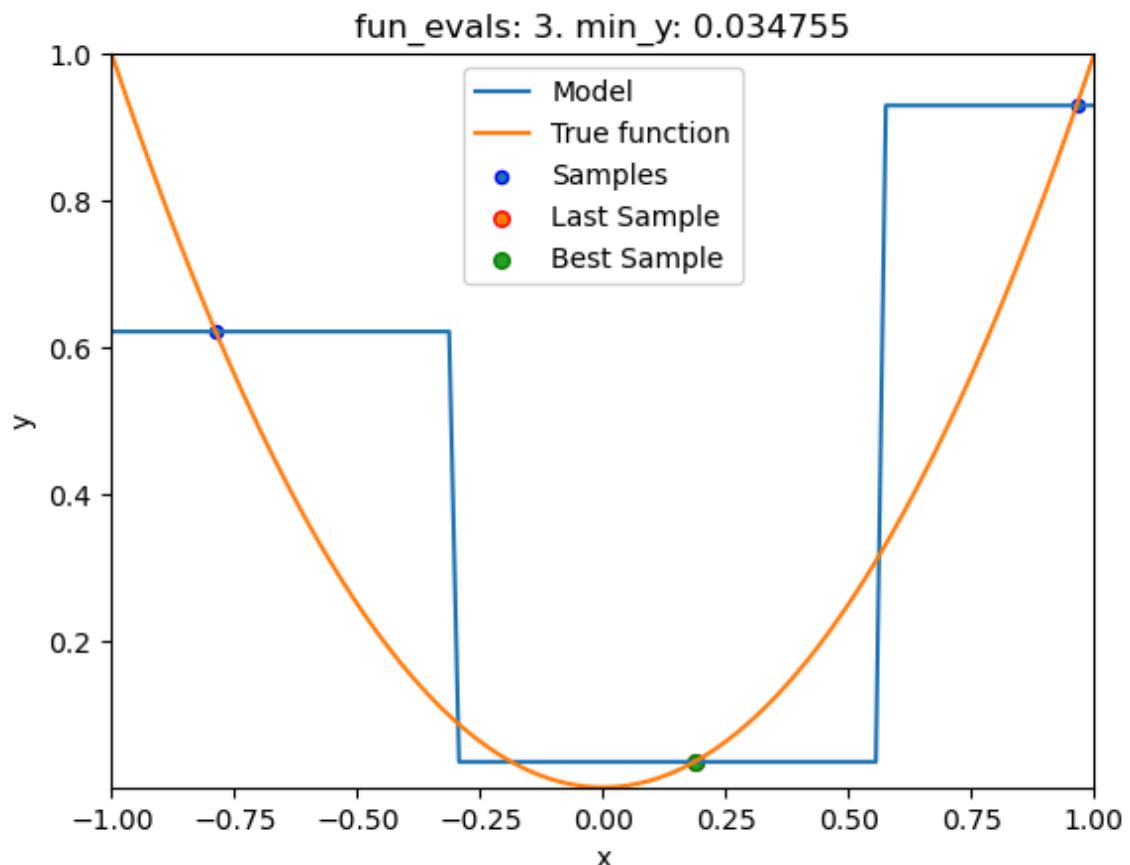
1. A decision tree regressor: DecisionTreeRegressor

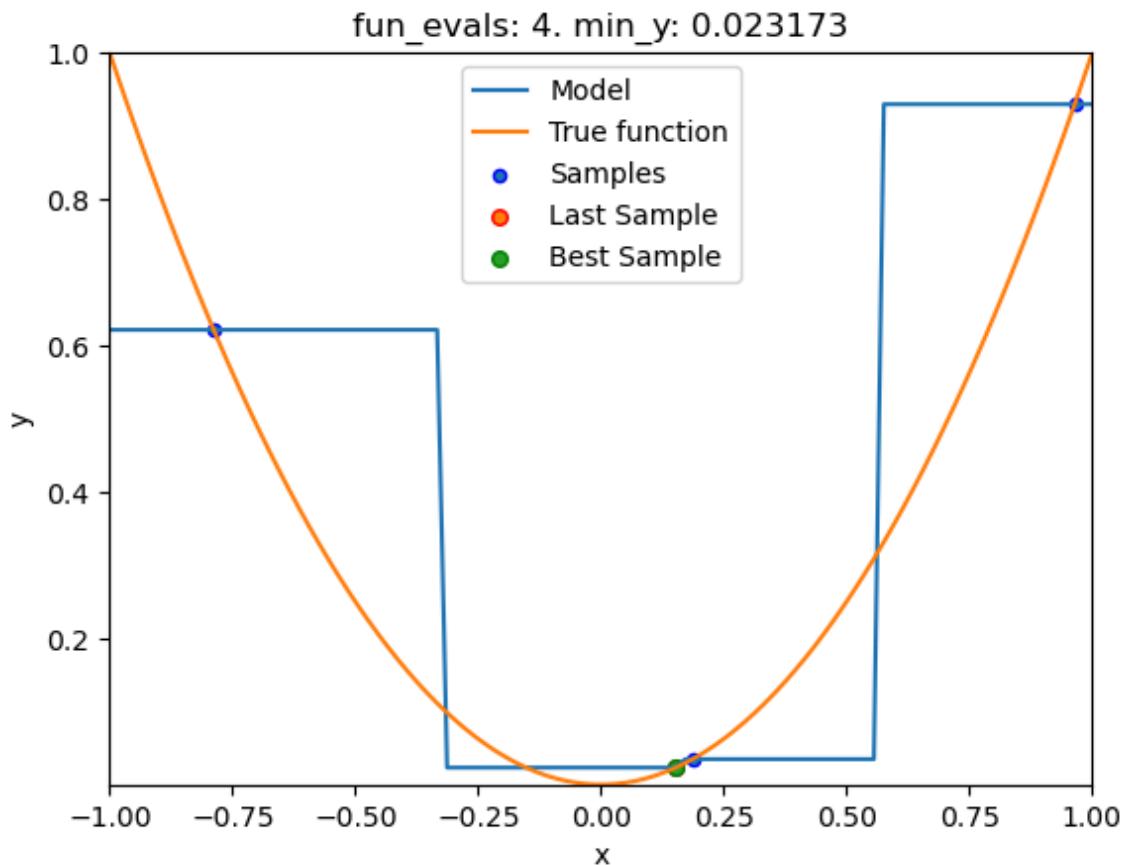
- Describe the surrogate model. Use the information from the [scikit-learn documentation](#).
- Use the surrogate as the model for optimization.

A decision tree regressor: DecisionTreeRegressor

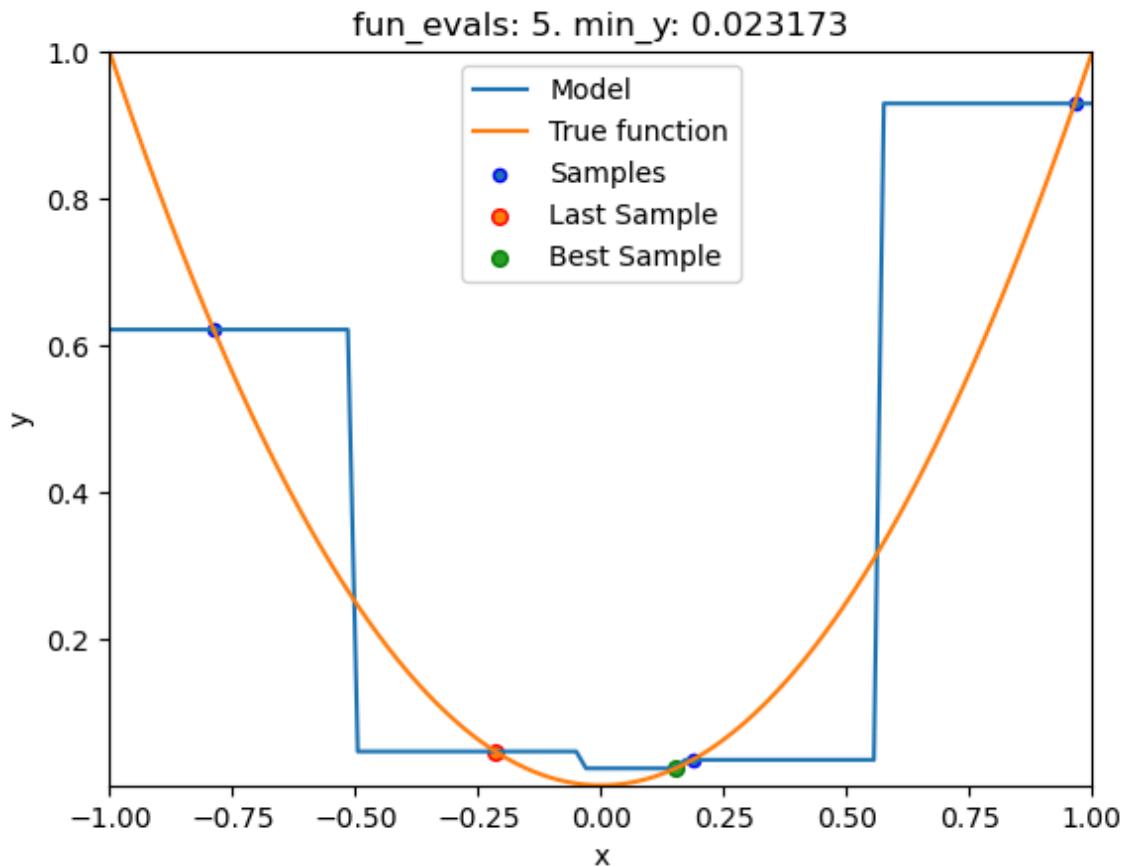
- The Decision Tree Regressor (DecisionTreeRegressor) is a flexible, non-linear regression model that constructs a tree to make predictions based on decision rules derived from the data.
- It is particularly useful for capturing complex relationships in the data without assuming a specific form for the underlying function.
- In the context of optimization, it serves as a surrogate model to approximate the objective function (e.g., the Sphere function) and guide the optimization process efficiently.

```
In [13]: #1-dim sphere function fun_sphere using Sklearn Model DecisionTreeRegressor
spot_1_DTR = spot.Spot(fun=fun,
                       fun_control=fun_control,
                       design_control=design_control,
                       surrogate = S_DTR)
spot_1_DTR.run()
spot_1_DTR.print_results()
spot_1_DTR.plot_progress(log_y=True)
spot_1_DTR.plot_model()
```

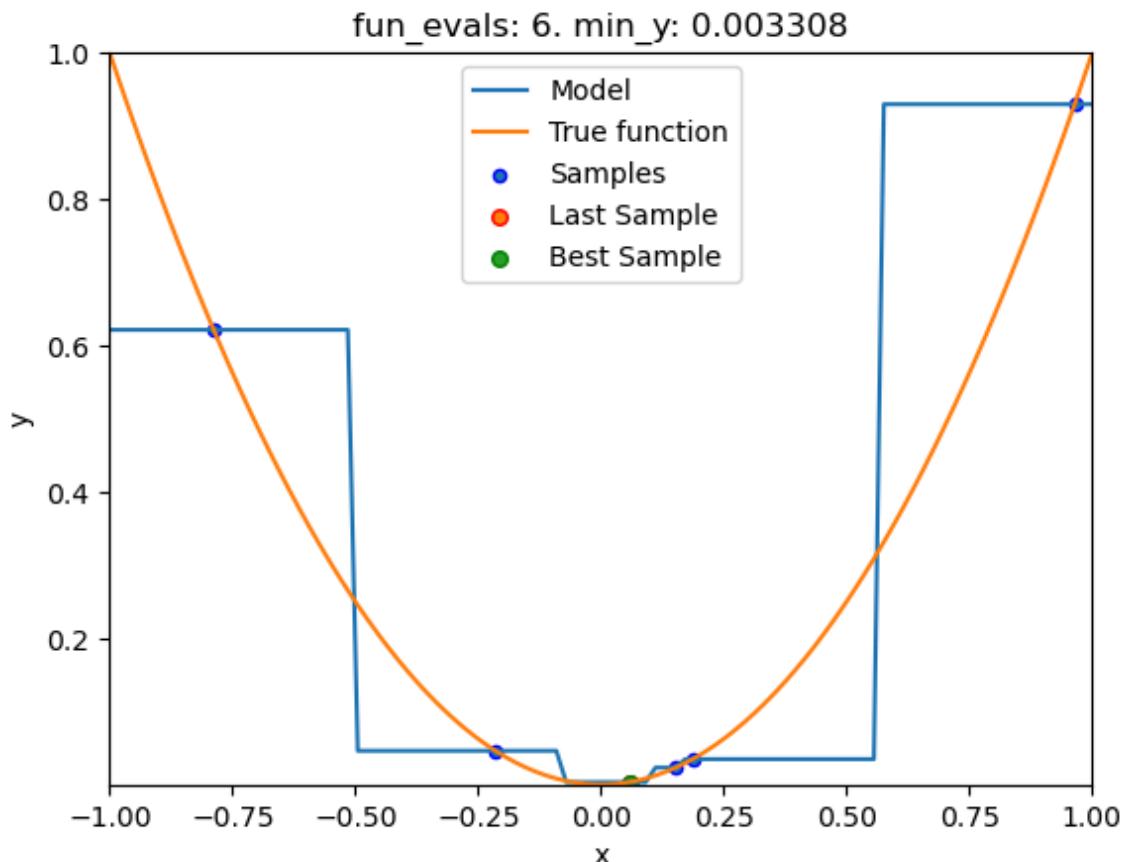




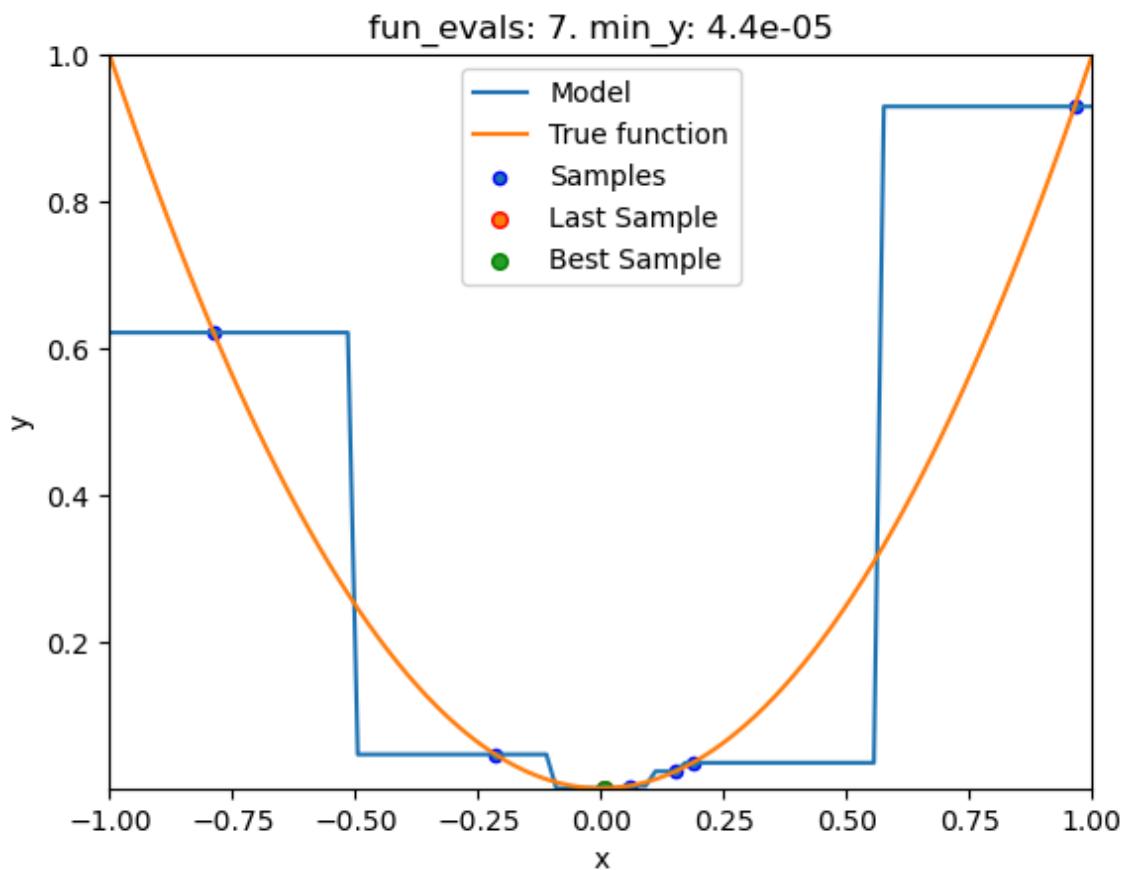
spotPython tuning: 0.02317347273817367 [#####-----] 40.00%



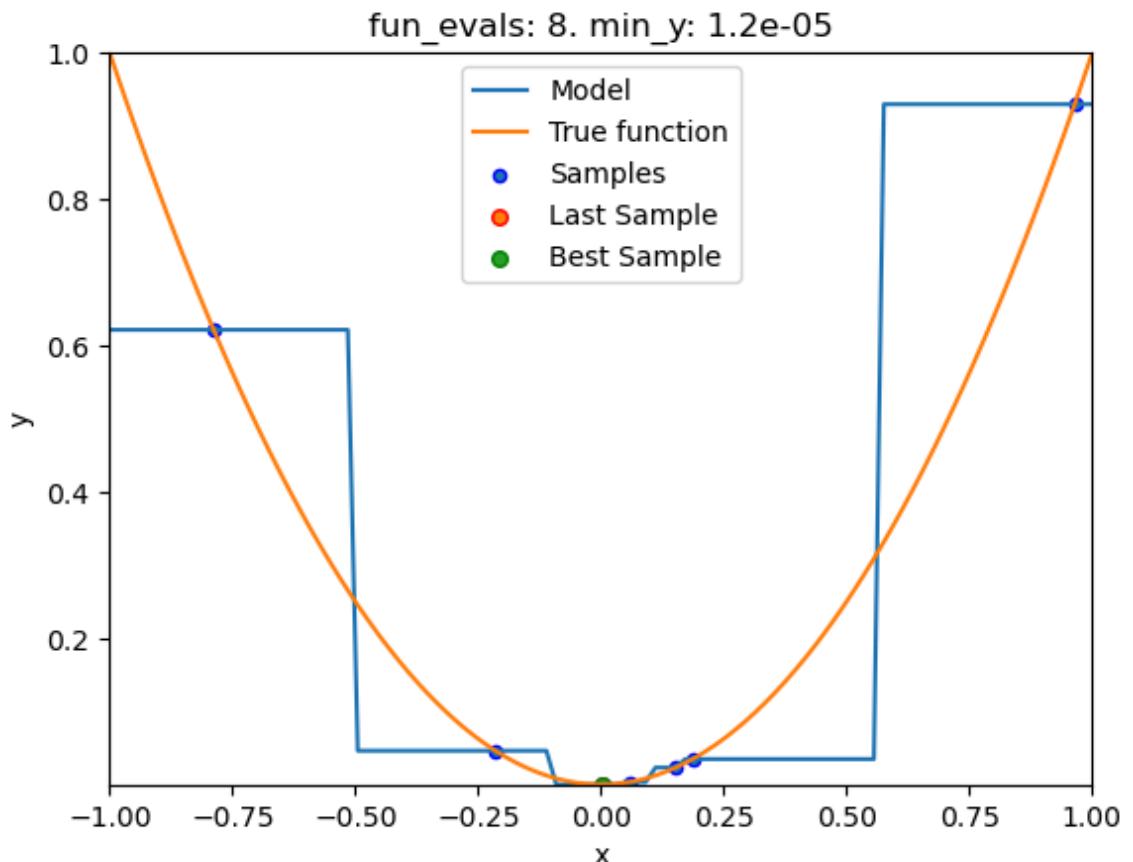
spotPython tuning: 0.02317347273817367 [#####-----] 50.00%



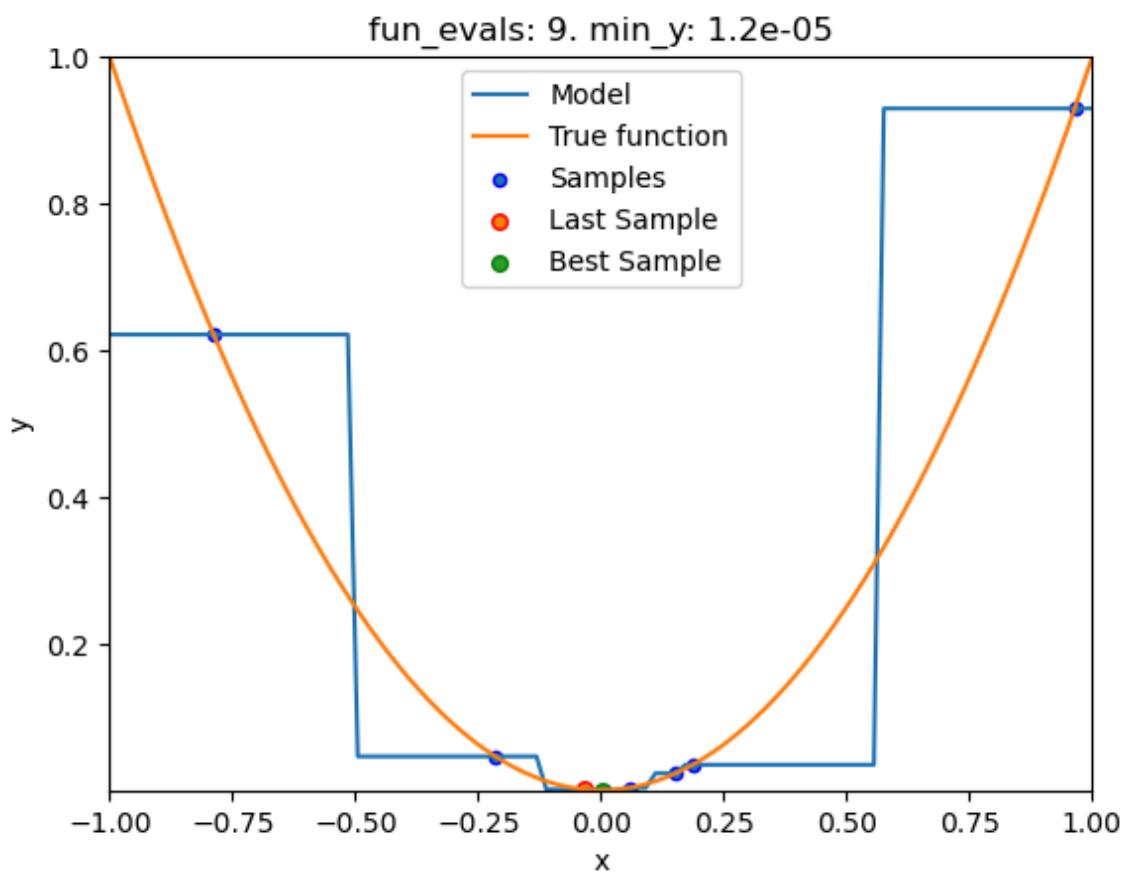
spotPython tuning: 0.003308345217006626 [#####----] 60.00%



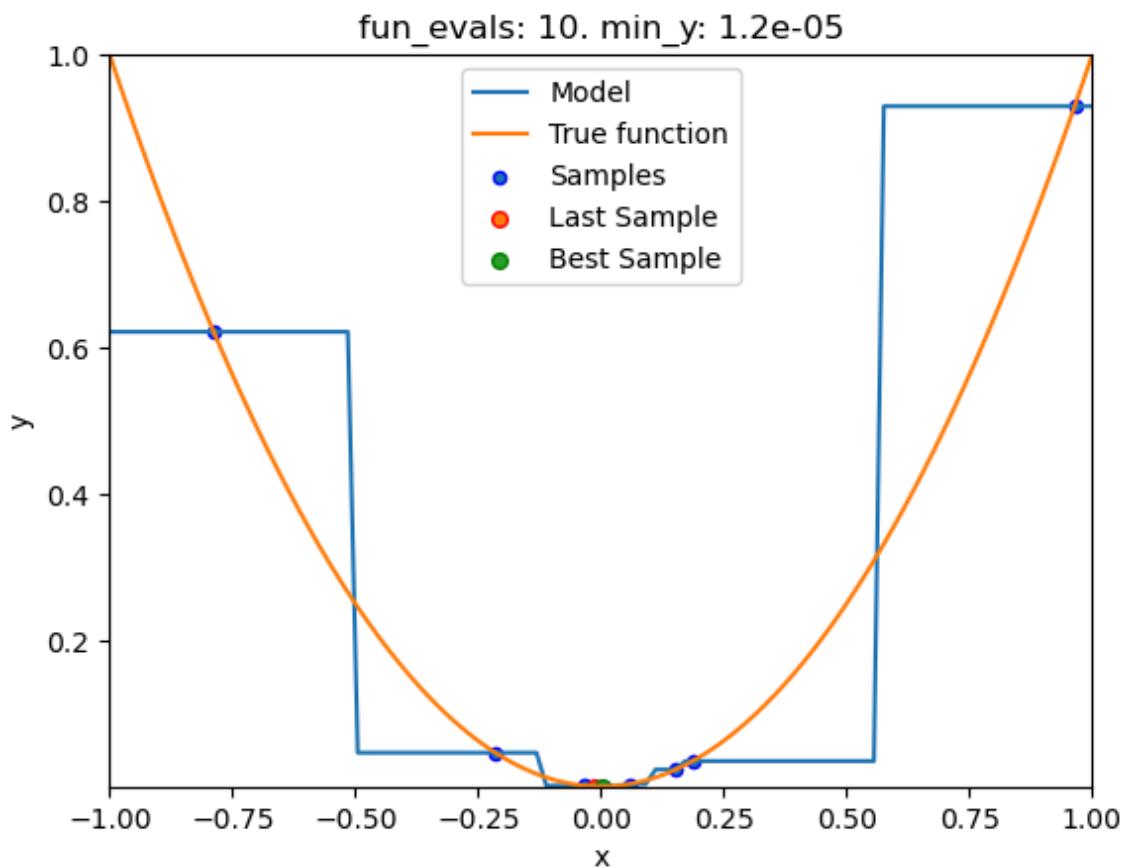
spotPython tuning: 4.364545534686172e-05 [#####----] 70.00%



spotPython tuning: 1.189146537170763e-05 [#####--] 80.00%



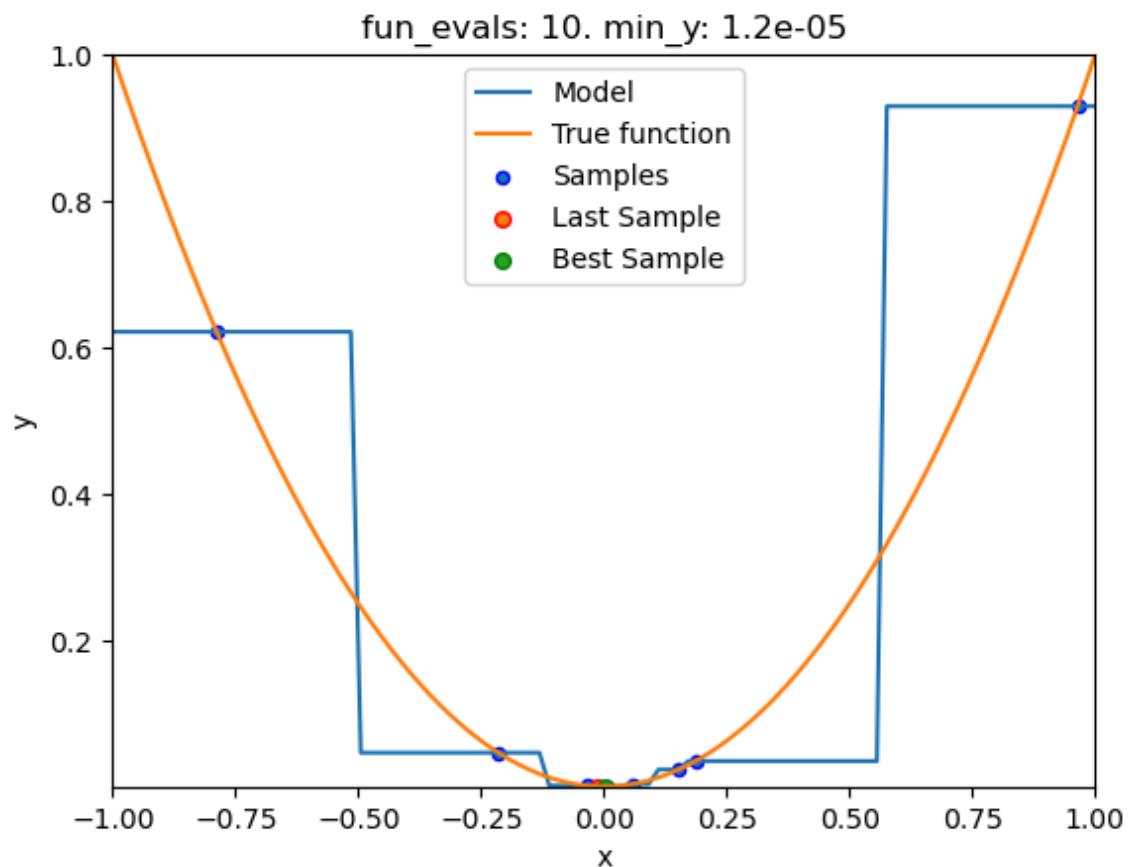
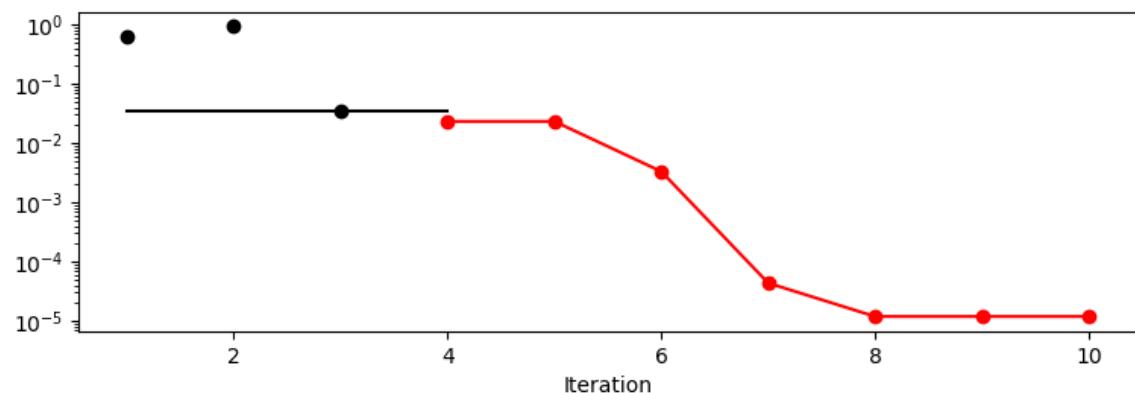
spotPython tuning: 1.189146537170763e-05 [#####--] 90.00%



```
spotPython tuning: 1.189146537170763e-05 [#####] 100.00% Done...
```

```
{'CHECKPOINT_PATH': 'runs/saved_models/',
'DATASET_PATH': 'data/',
'PREFIX': None,
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,
'core_model_name': None,
'counter': 10,
'data': None,
'data_dir': './data',
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': 10,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([-1]),
'max_surrogate_points': 30,
'max_time': inf,
'metric_params': {},
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': True,
'show_progress': True,
```

```
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': None,
'spot_writer': None,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 1.4901161193847656e-08,
'train': None,
'upper': array([1]),
'verbose': None,
'var_name': None,
'var_type': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 1.189146537170763e-05
x0: 0.0034484004076829056
```



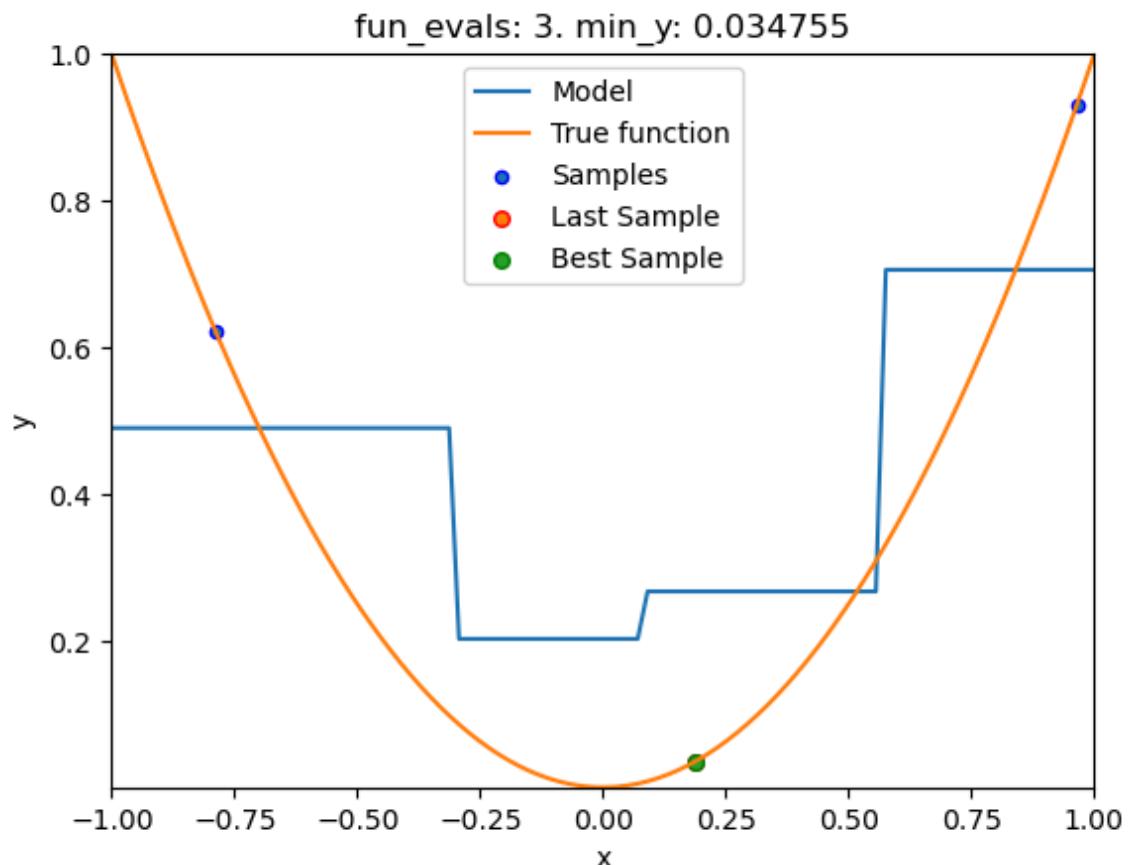
2. A random forest regressor: RandomForestRegressor

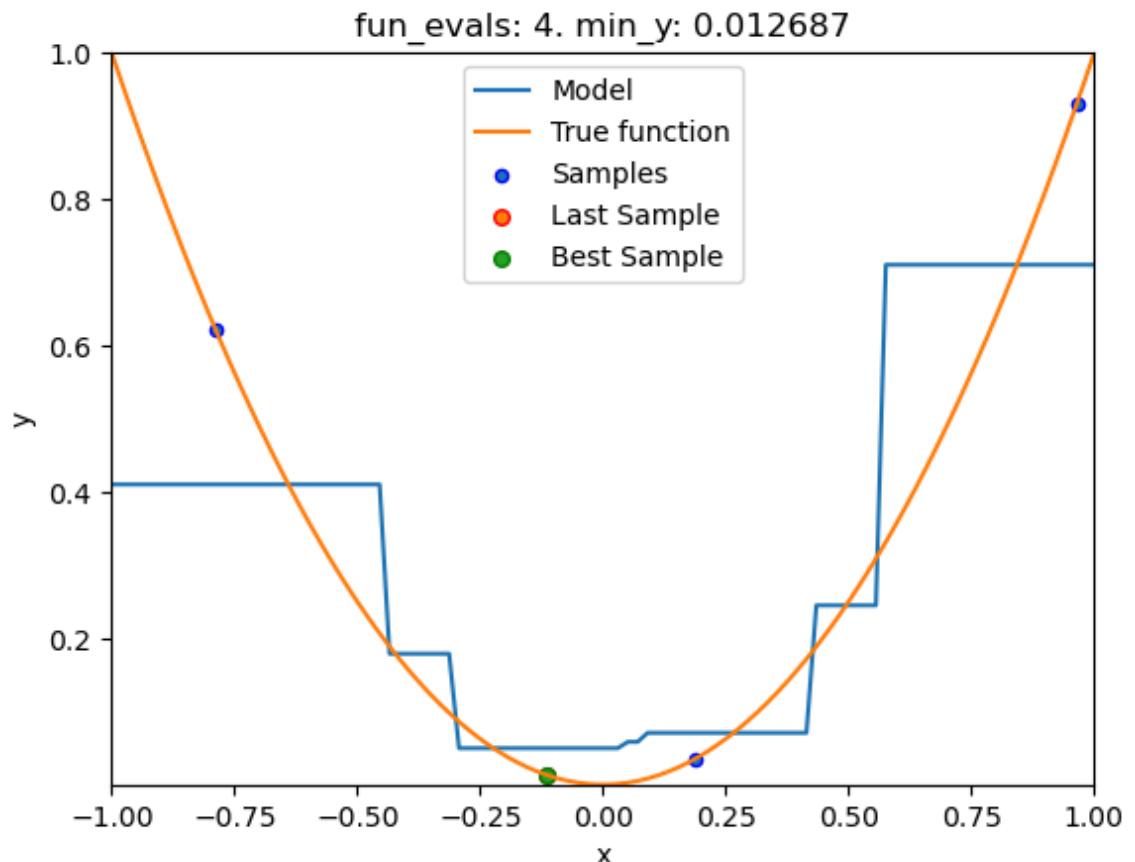
- Describe the surrogate model. Use the information from the [scikit-learn documentation](#).
- Use the surrogate as the model for optimization.

A random forest regressor: RandomForestRegressor

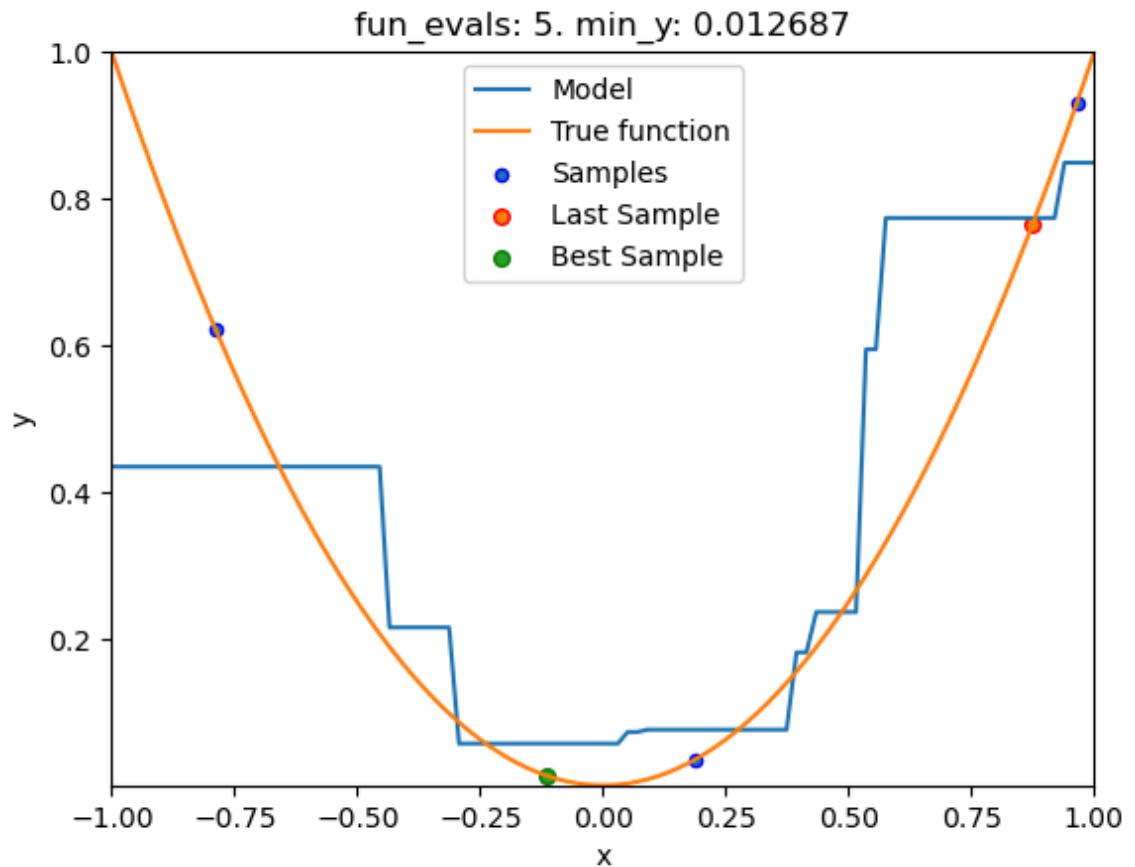
- The Random Forest Regressor (RandomForestRegressor) is an ensemble learning model that leverages the power of multiple decision trees to improve prediction accuracy and reduce overfitting.
- It achieves this through bootstrap sampling and feature randomness during tree construction.
- In the context of optimization, it serves as a surrogate model to approximate the objective function (e.g., the Sphere function) and guide the optimization process efficiently.

```
In [14]: #1-dim sphere function fun_sphere using Sklearn Model RandomForestRegressor
spot_1_RFR = spot.Spot(fun=fun,
                       fun_control=fun_control,
                       design_control=design_control,
                       surrogate = S_RFR)
spot_1_RFR.run()
spot_1_RFR.print_results()
spot_1_RFR.plot_progress(log_y=True)
spot_1_RFR.plot_model()
```

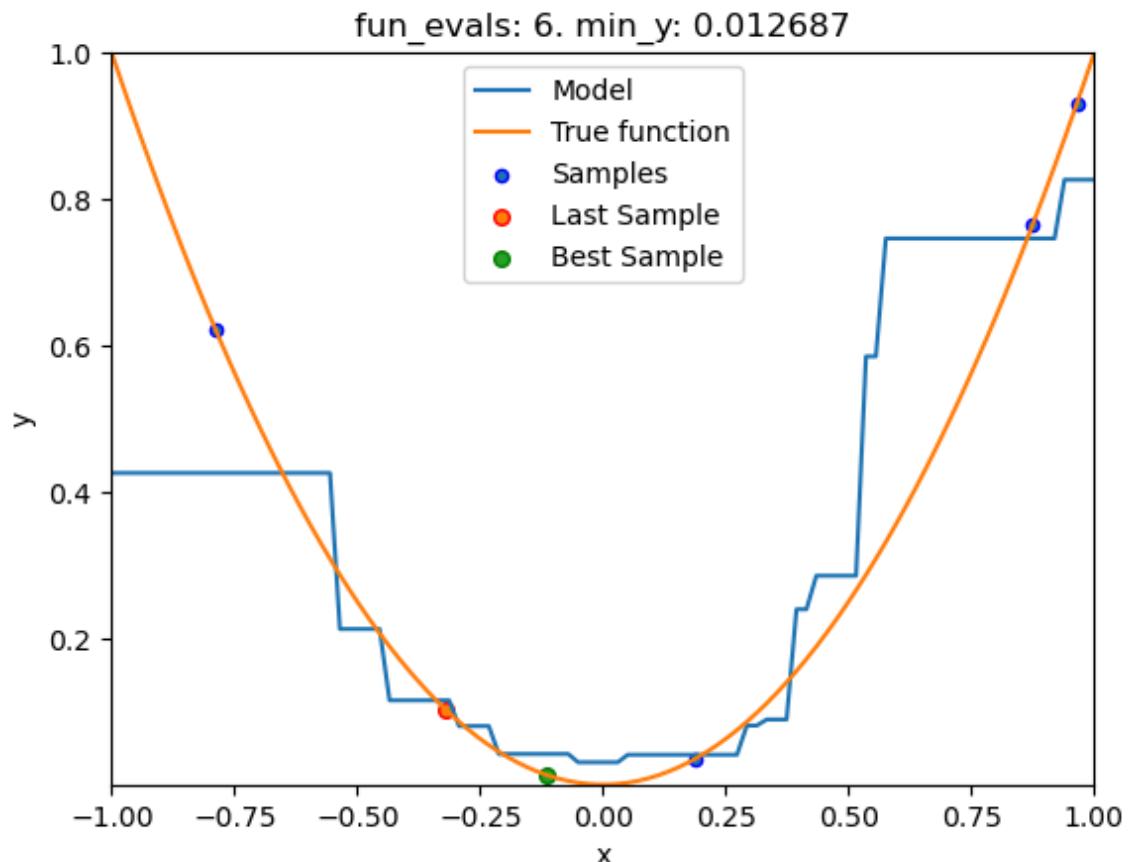




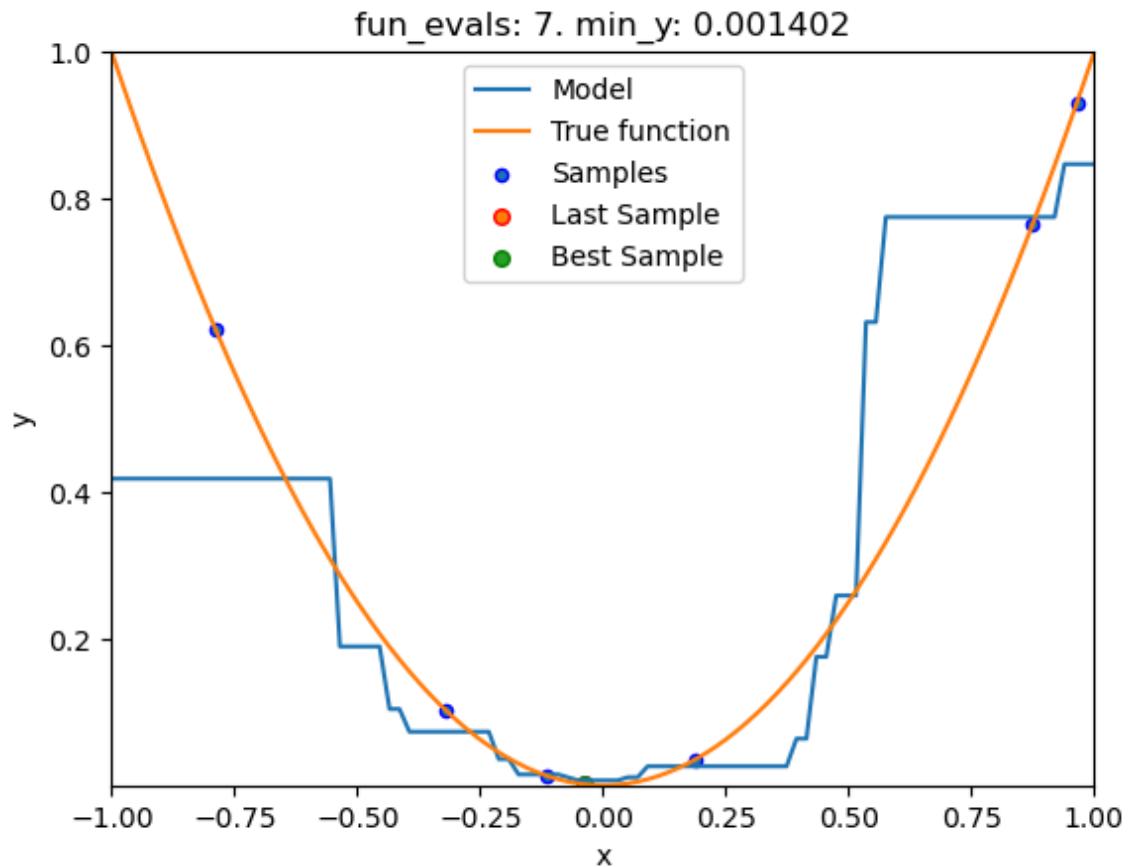
spotPython tuning: 0.012687060069554874 [#####-----] 40.00%



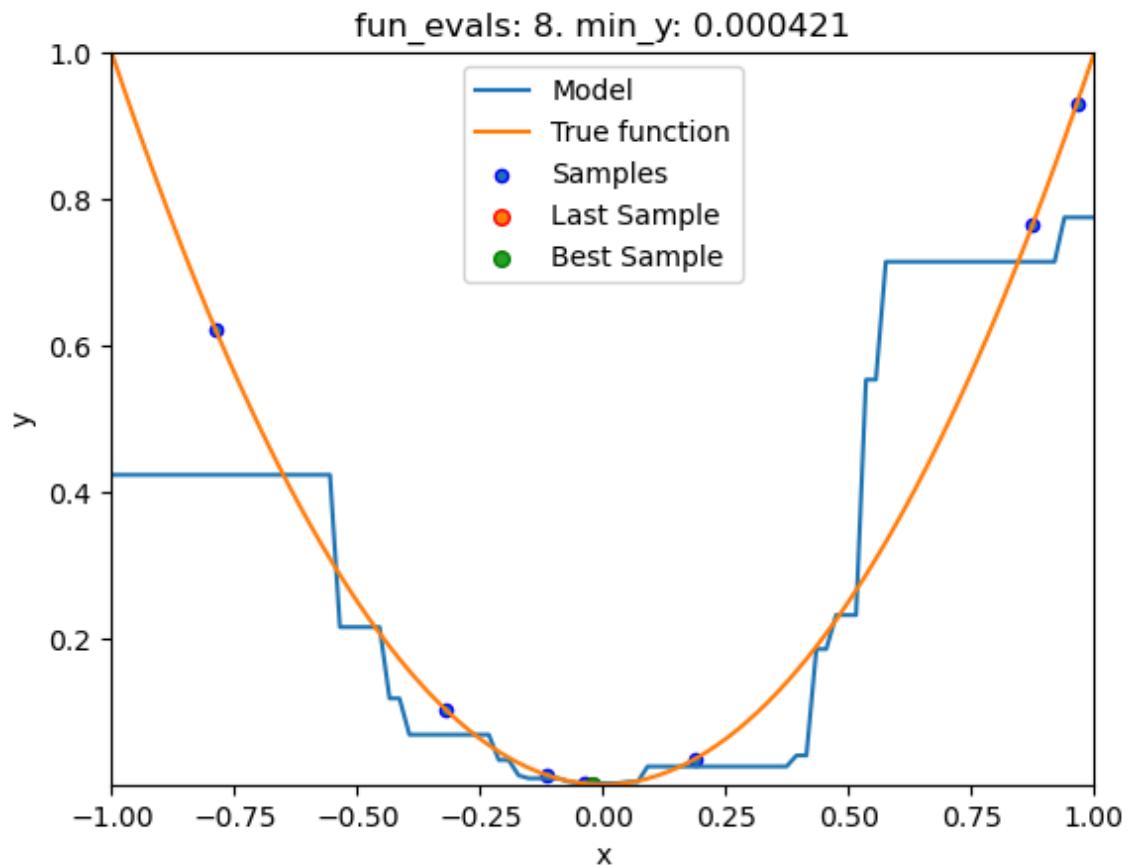
spotPython tuning: 0.012687060069554874 [#####-----] 50.00%



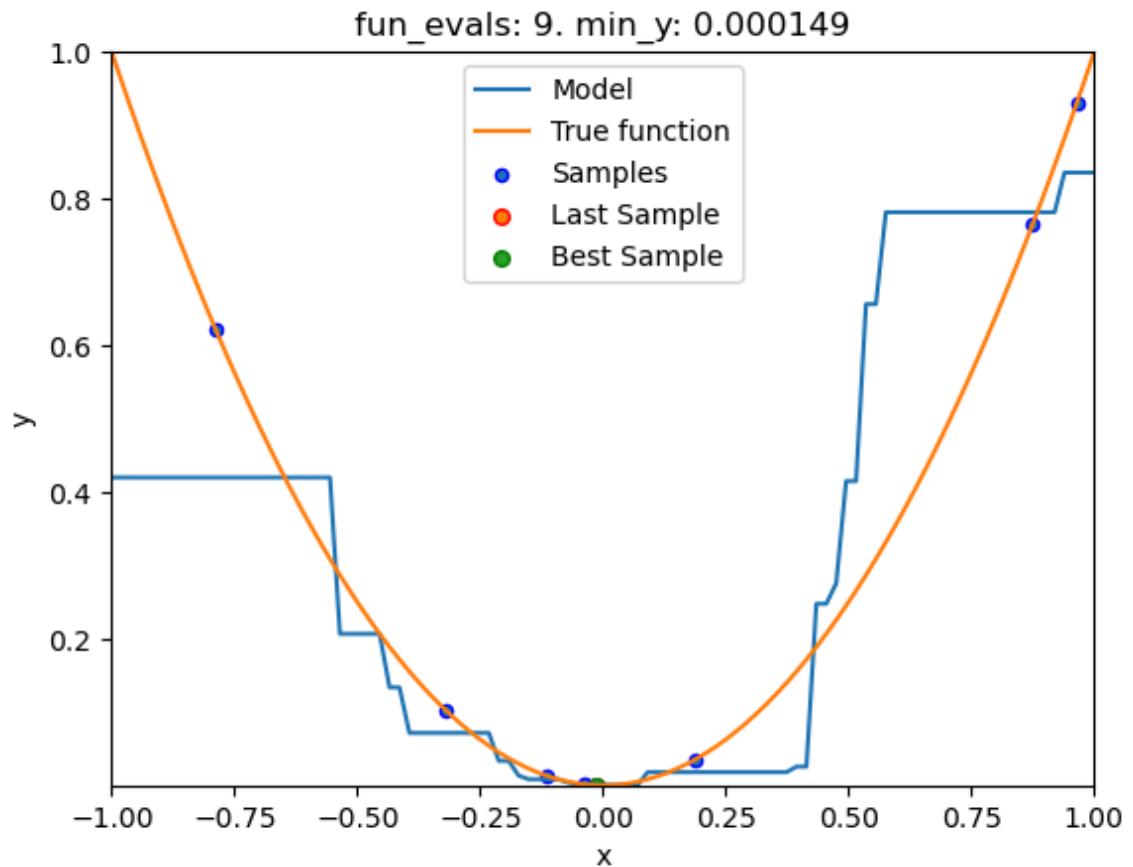
spotPython tuning: 0.012687060069554874 [#####----] 60.00%



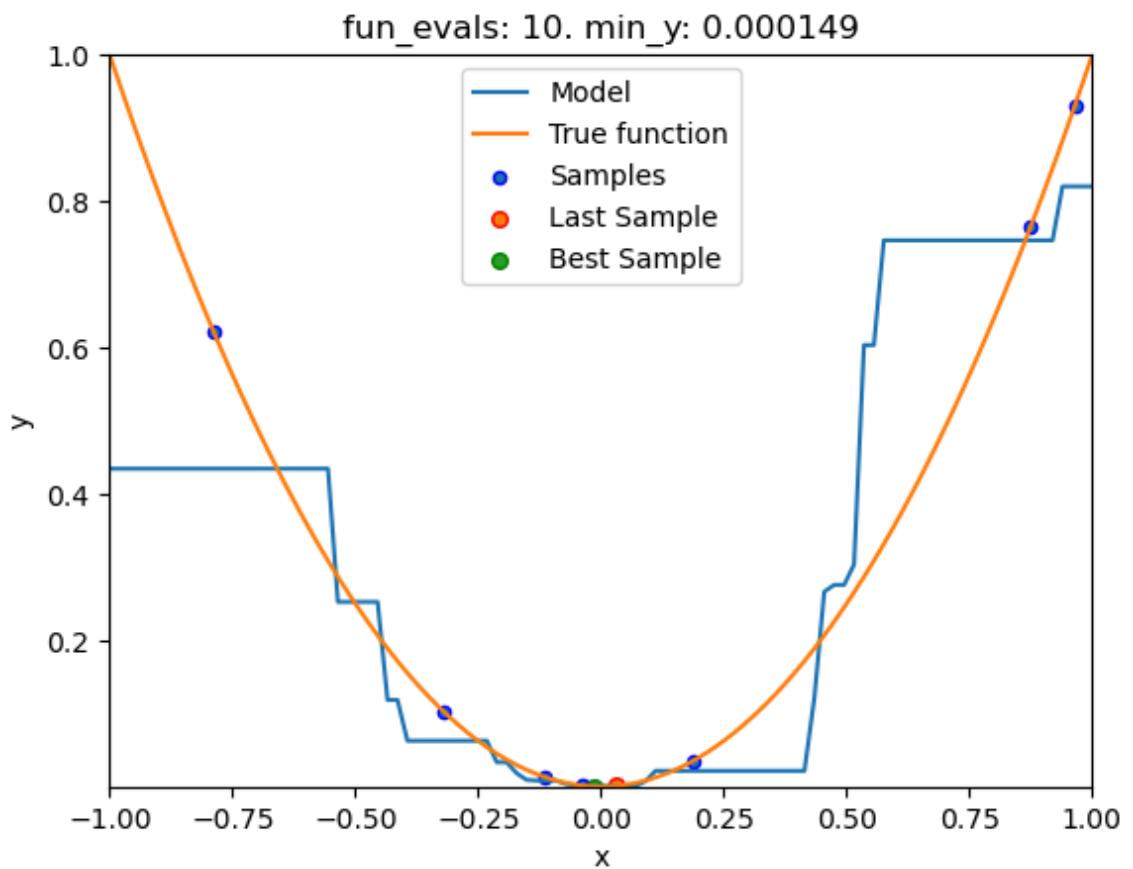
spotPython tuning: 0.0014024463394194796 [#####----] 70.00%



spotPython tuning: 0.0004211231331030619 [#####--] 80.00%



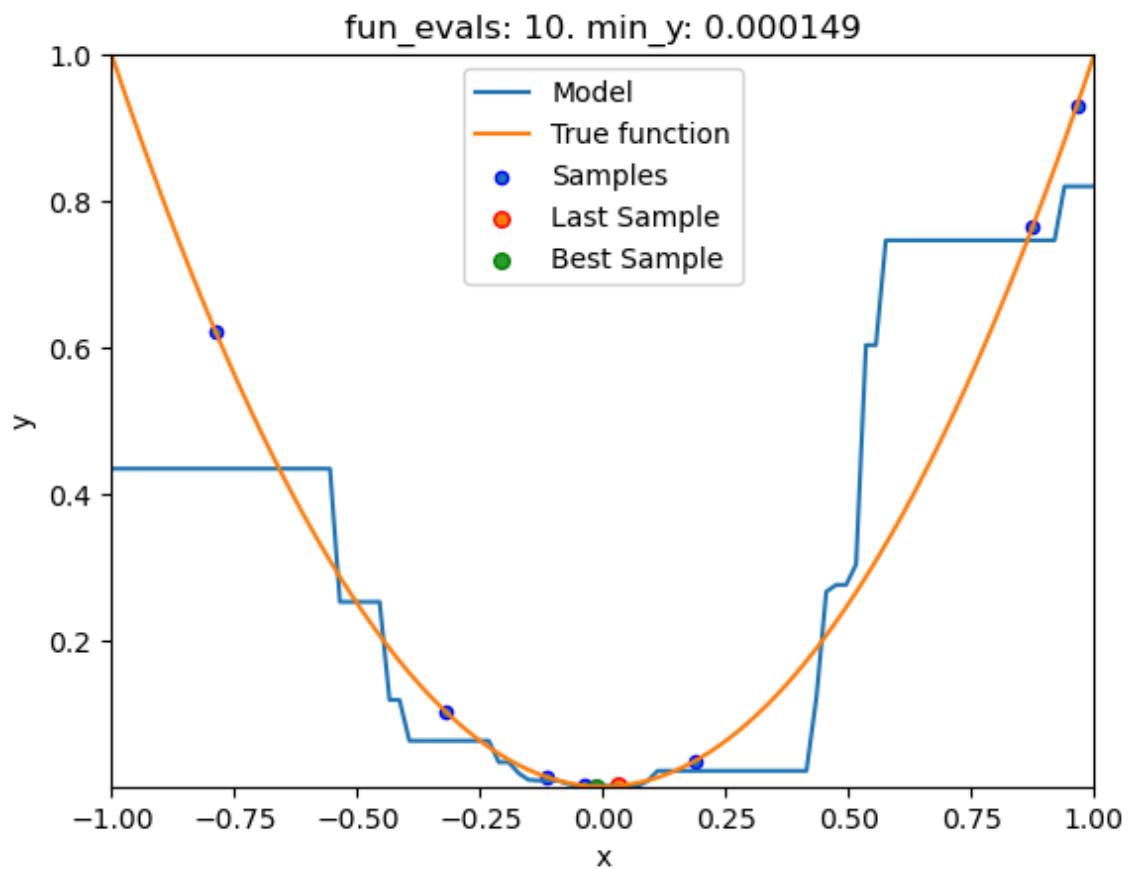
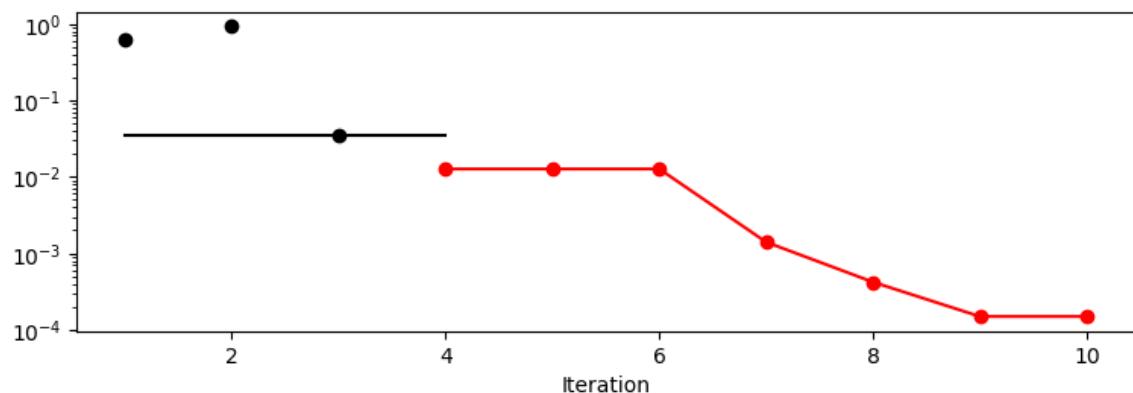
spotPython tuning: 0.00014944386805438418 [#####--] 90.00%



spotPython tuning: 0.00014944386805438418 [#####] 100.00% Done...

```
{'CHECKPOINT_PATH': 'runs/saved_models/',
'DATASET_PATH': 'data/',
'PREFIX': None,
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,
'core_model_name': None,
'counter': 10,
'data': None,
'data_dir': './data',
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': 10,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([-1]),
'max_surrogate_points': 30,
'max_time': inf,
'metric_params': {},
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': True,
'show_progress': True,
```

```
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': None,
'spot_writer': None,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 1.4901161193847656e-08,
'train': None,
'upper': array([1]),
'vername': None,
'vertype': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 0.00014944386805438418
x0: -0.012224723639182367
```



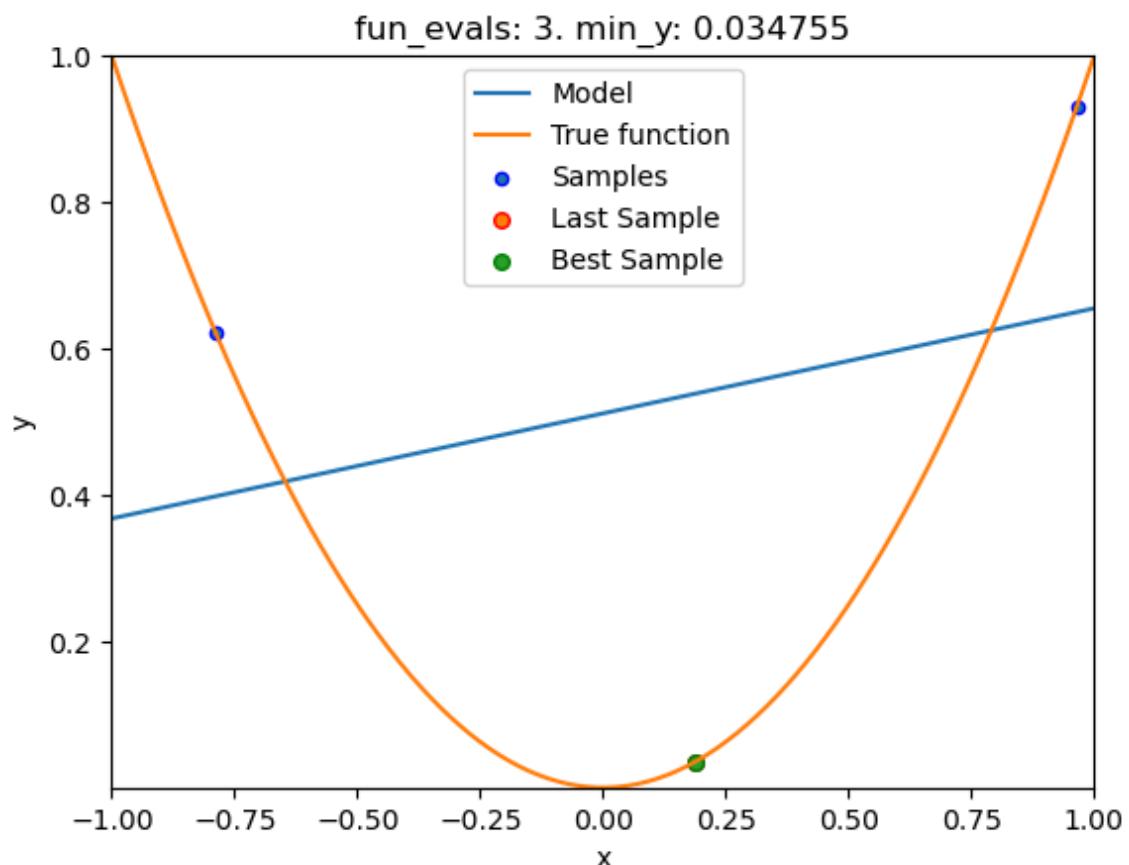
3. Ordinary least squares Linear Regression: LinearRegression

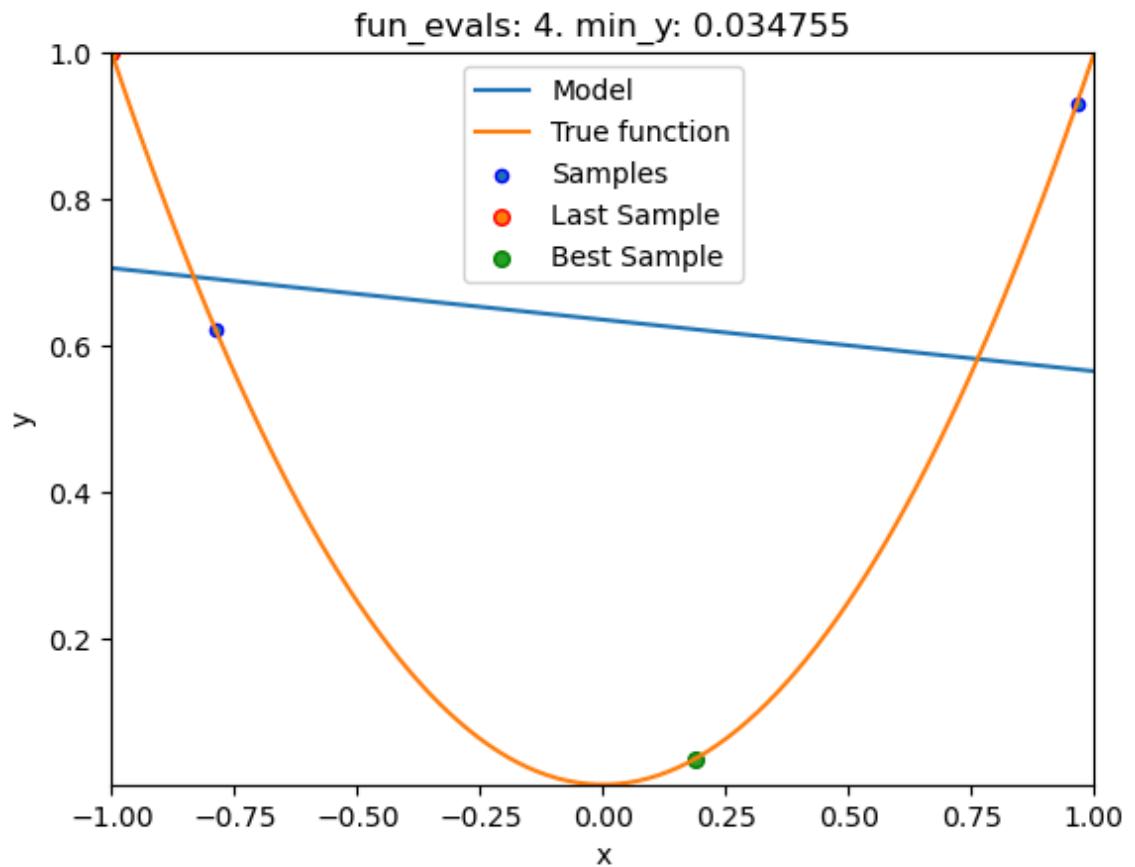
- Describe the surrogate model. Use the information from the [scikit-learn documentation](#).
- Use the surrogate as the model for optimization.

Ordinary least squares Linear Regression: LinearRegression

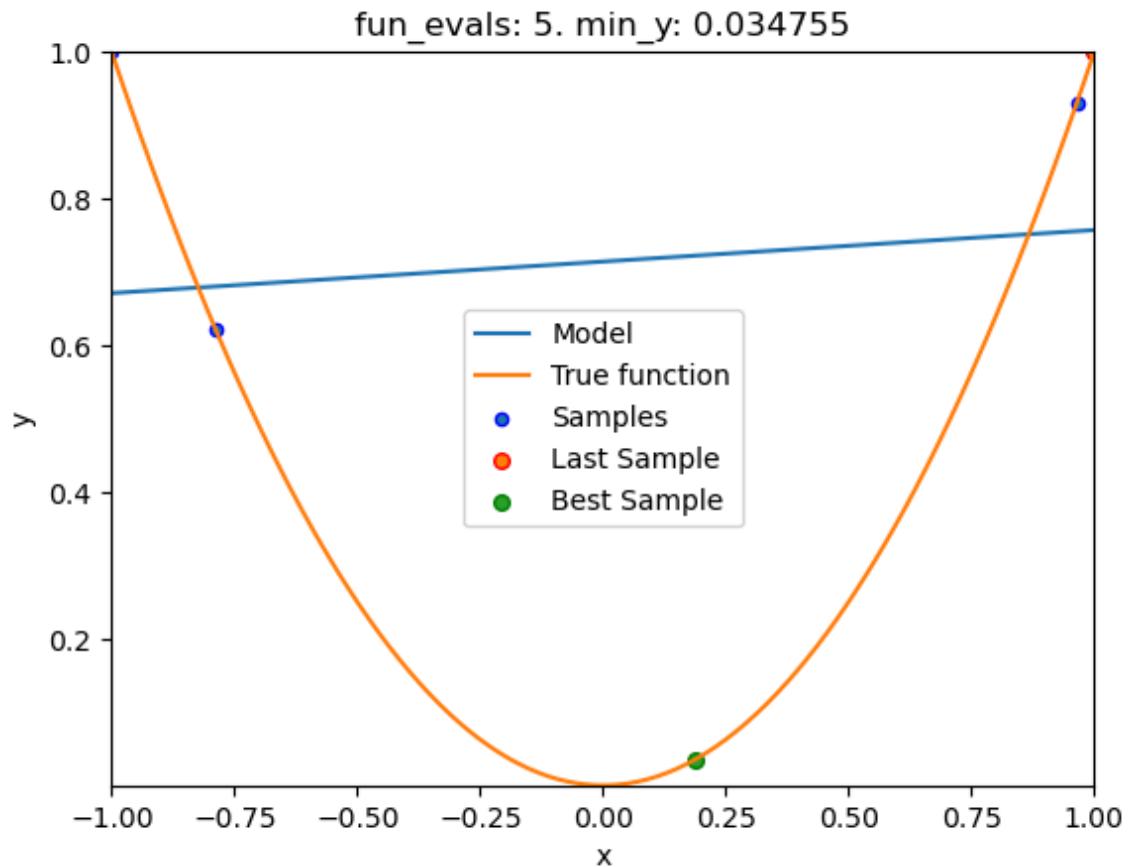
- Ordinary Least Squares (OLS) Linear Regression is a fundamental regression technique that models the relationship between a dependent variable and one or more independent variables by fitting a linear equation to the observed data.
- It aims to find the best-fit line by minimizing the sum of squared differences between observed and predicted values.

```
In [15]: #1-dim sphere function fun_sphere using Sklearn Model LinearRegression
spot_1_LR = spot.Spot(fun=fun,
                      fun_control=fun_control,
                      design_control=design_control,
                      surrogate = S_LR)
spot_1_LR.run()
spot_1_LR.print_results()
spot_1_LR.plot_progress(log_y=True)
spot_1_LR.plot_model()
```

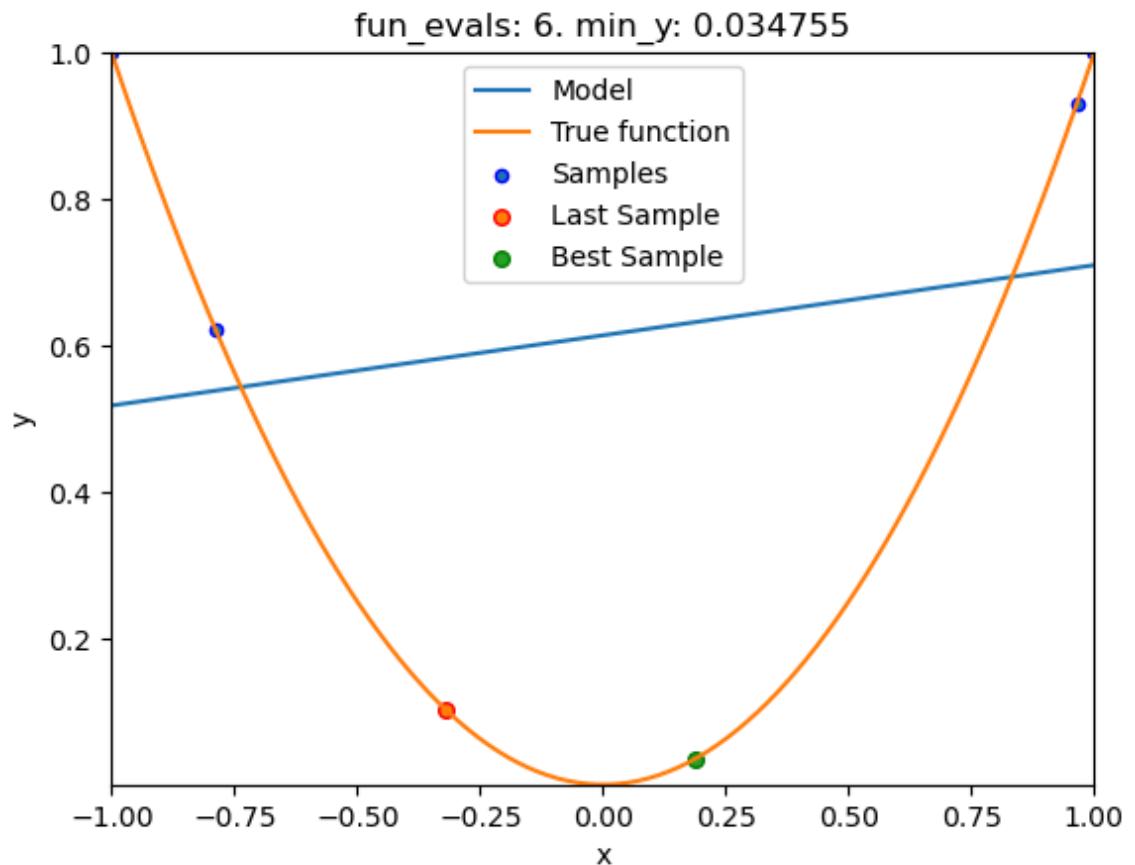




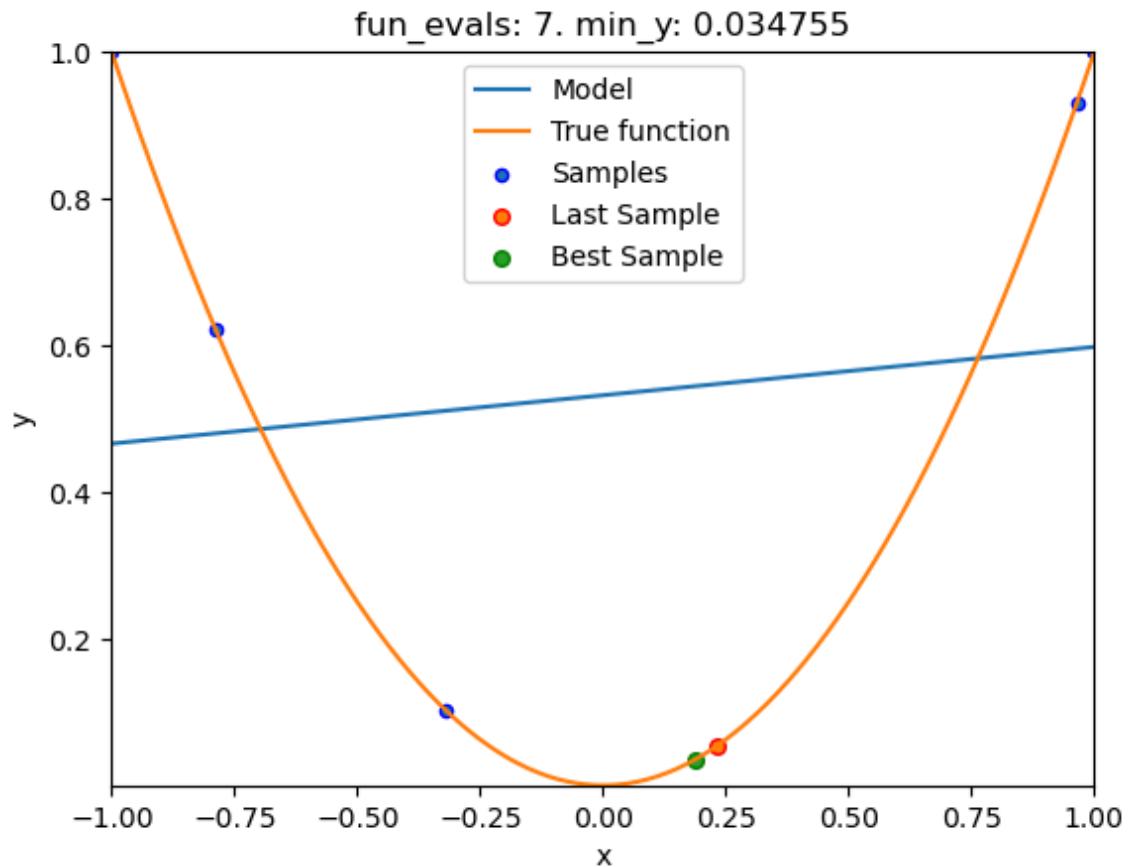
spotPython tuning: 0.03475493366922229 [#####-----] 40.00%



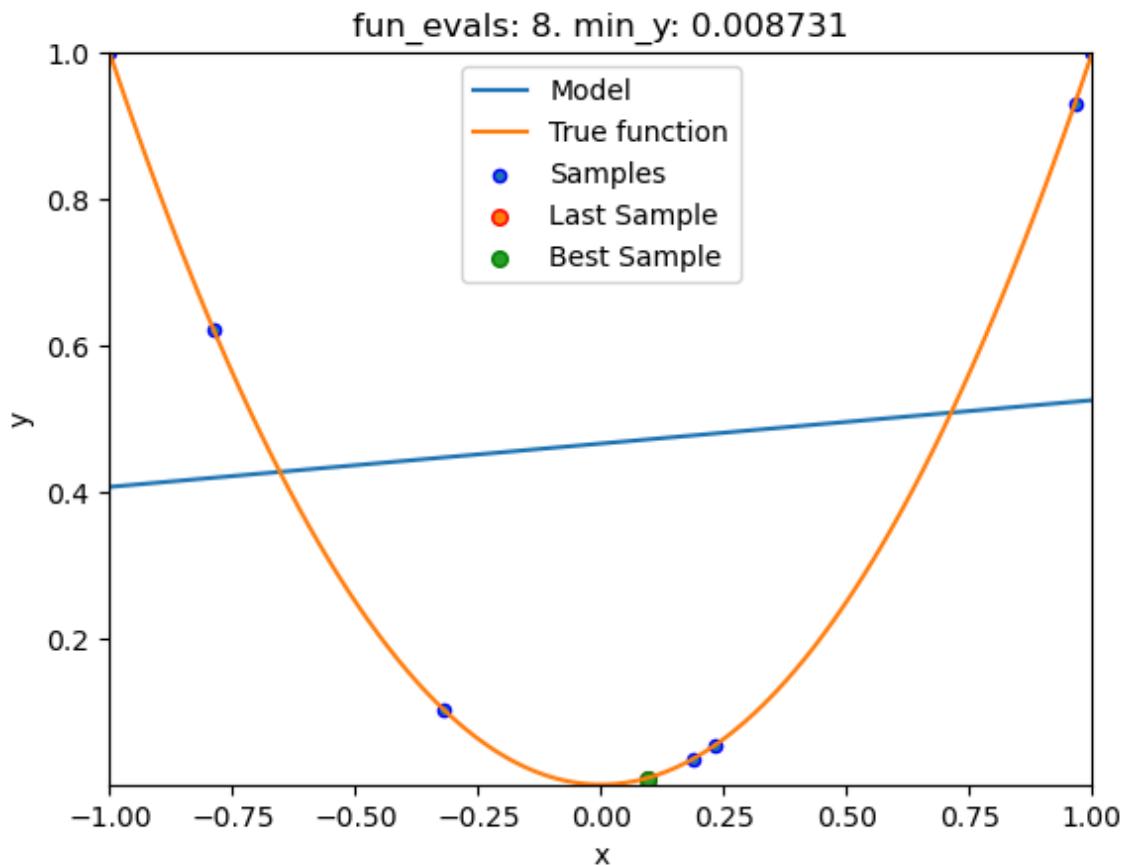
spotPython tuning: 0.03475493366922229 [#####-----] 50.00%



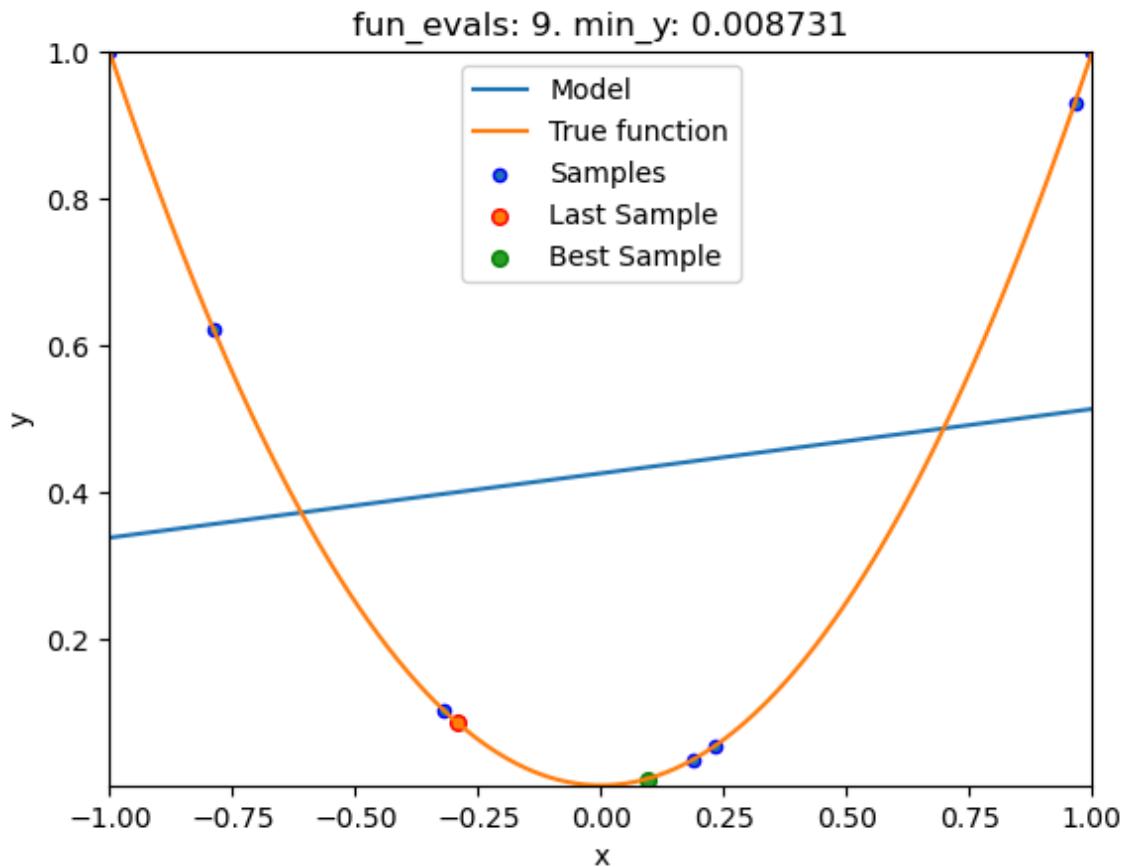
spotPython tuning: 0.03475493366922229 [#####----] 60.00%



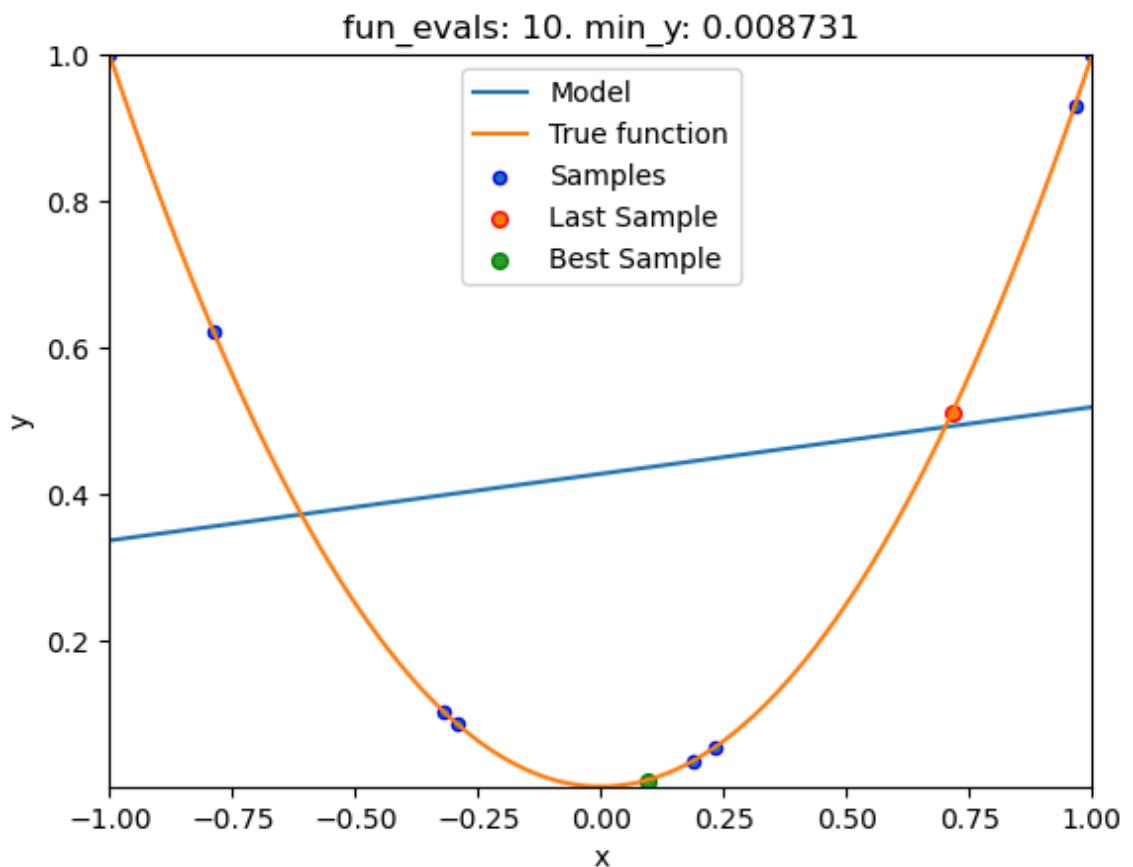
spotPython tuning: 0.03475493366922229 [#####---] 70.00%



spotPython tuning: 0.008730885505764131 [#####--] 80.00%



spotPython tuning: 0.008730885505764131 [#####--] 90.00%



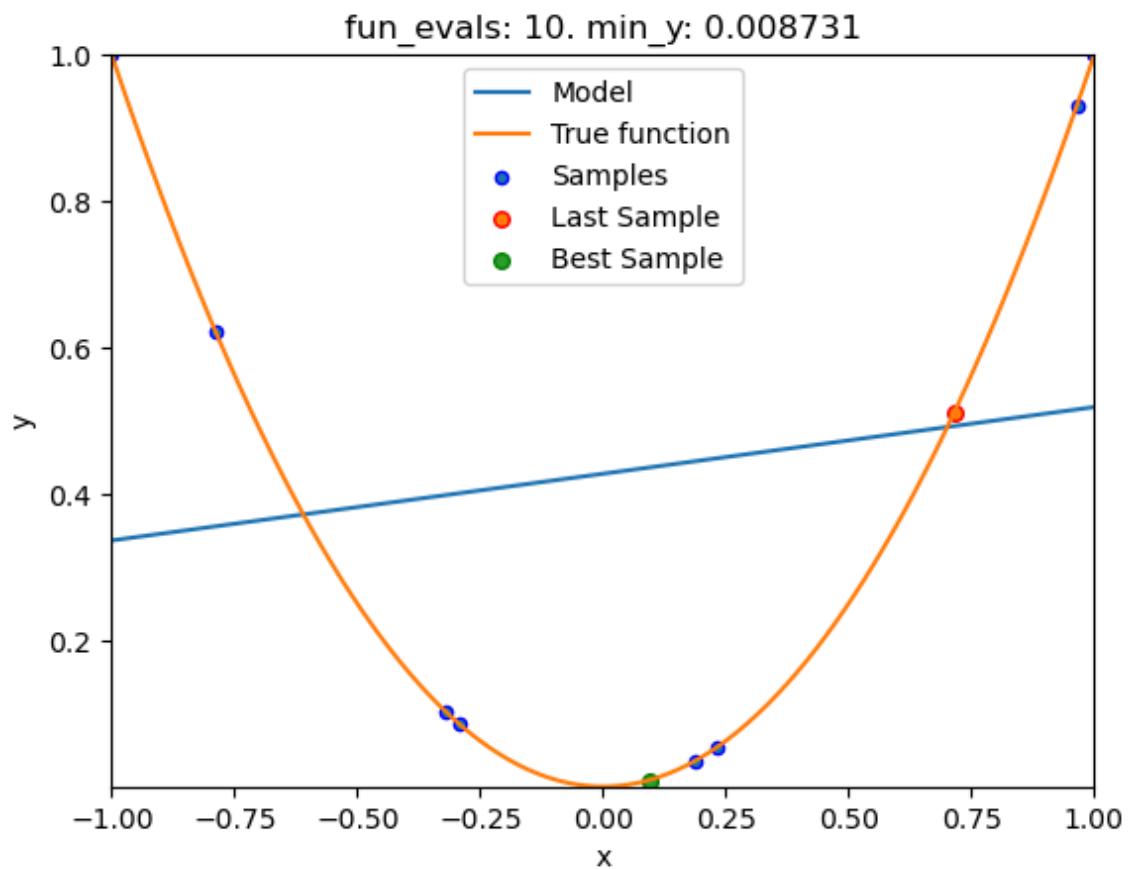
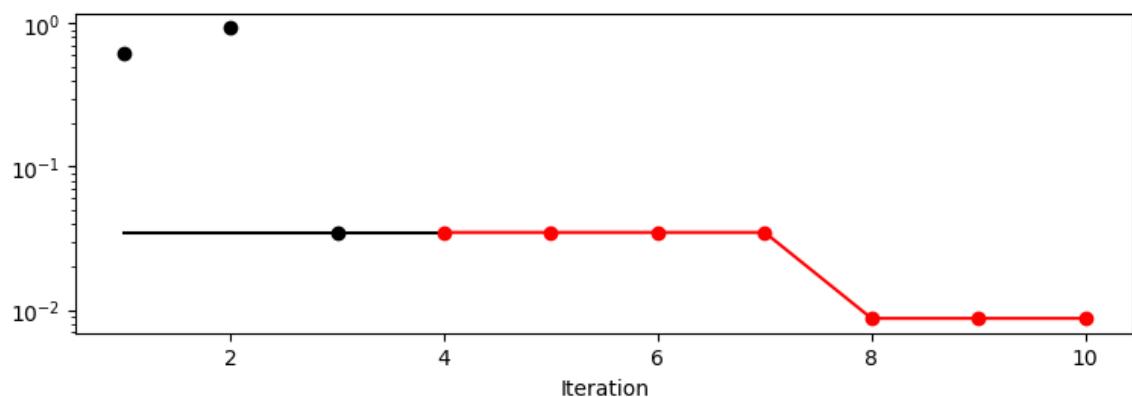
```
spotPython tuning: 0.008730885505764131 [#####] 100.00% Done...
```

```
{'CHECKPOINT_PATH': 'runs/saved_models/',
'DATASET_PATH': 'data/',
'PREFIX': None,
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,
'core_model_name': None,
'counter': 10,
'data': None,
'data_dir': './data',
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': 10,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([-1]),
'max_surrogate_points': 30,
'max_time': inf,
'metric_params': {},
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': True,
'show_progress': True,
```

```

'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': None,
'spot_writer': None,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 1.4901161193847656e-08,
'train': None,
'upper': array([1]),
'vername': None,
'vertype': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 0.008730885505764131
x0: 0.09343920754032609

```



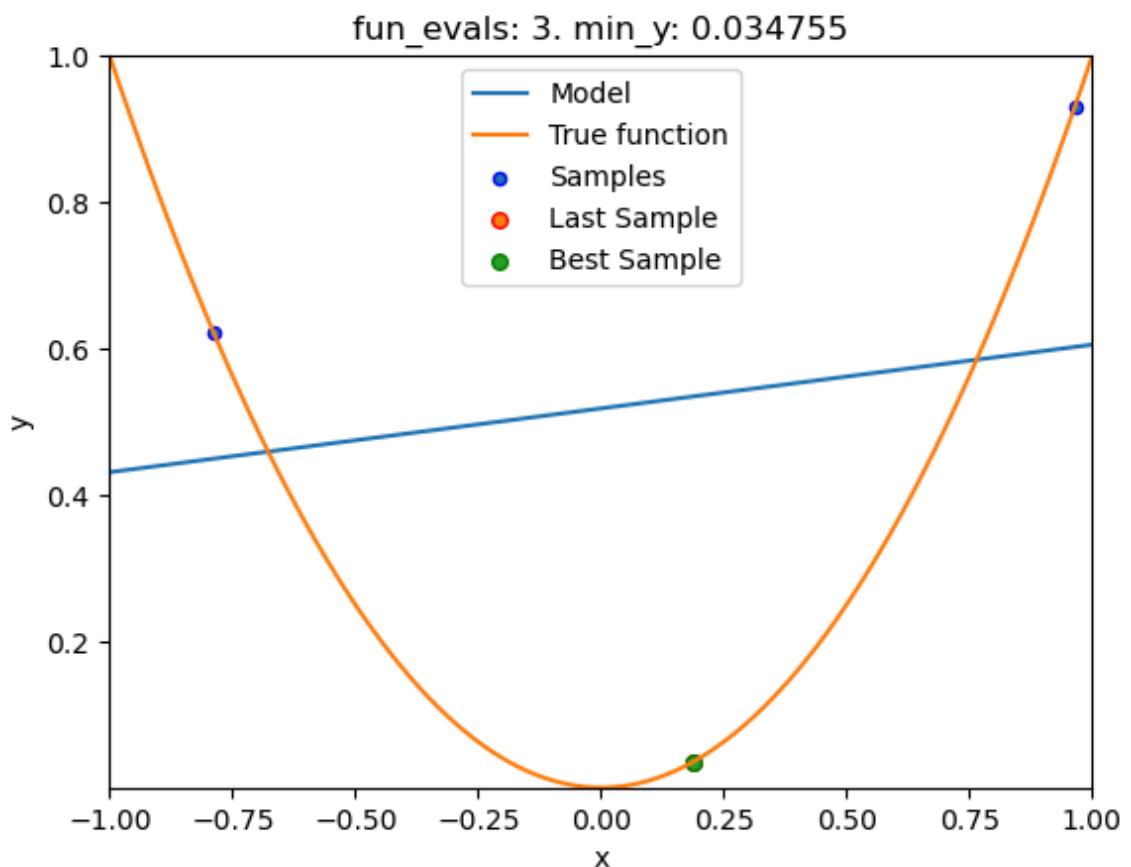
4. Linear least squares with L2 regularization: Ridge

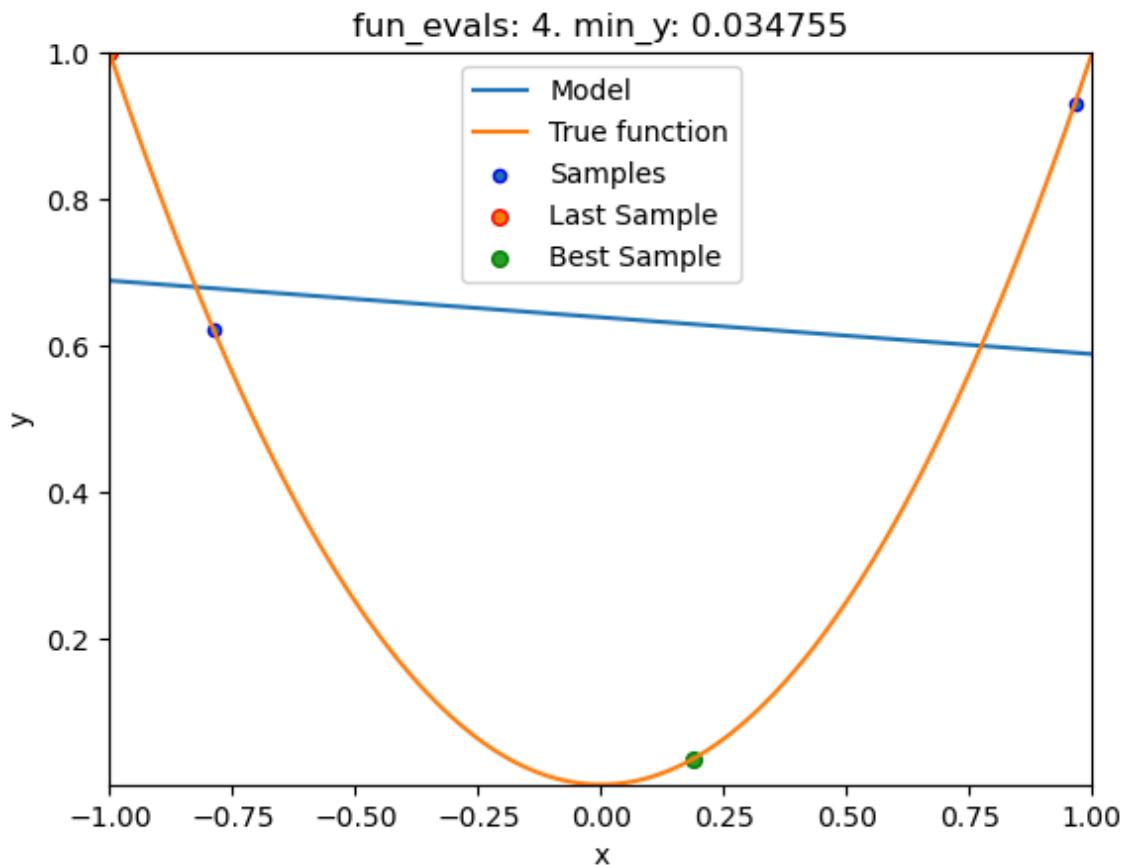
- Describe the surrogate model. Use the information from the [scikit-learn documentation](#).
- Use the surrogate as the model for optimization.

Linear least squares with L2 regularization: Ridge

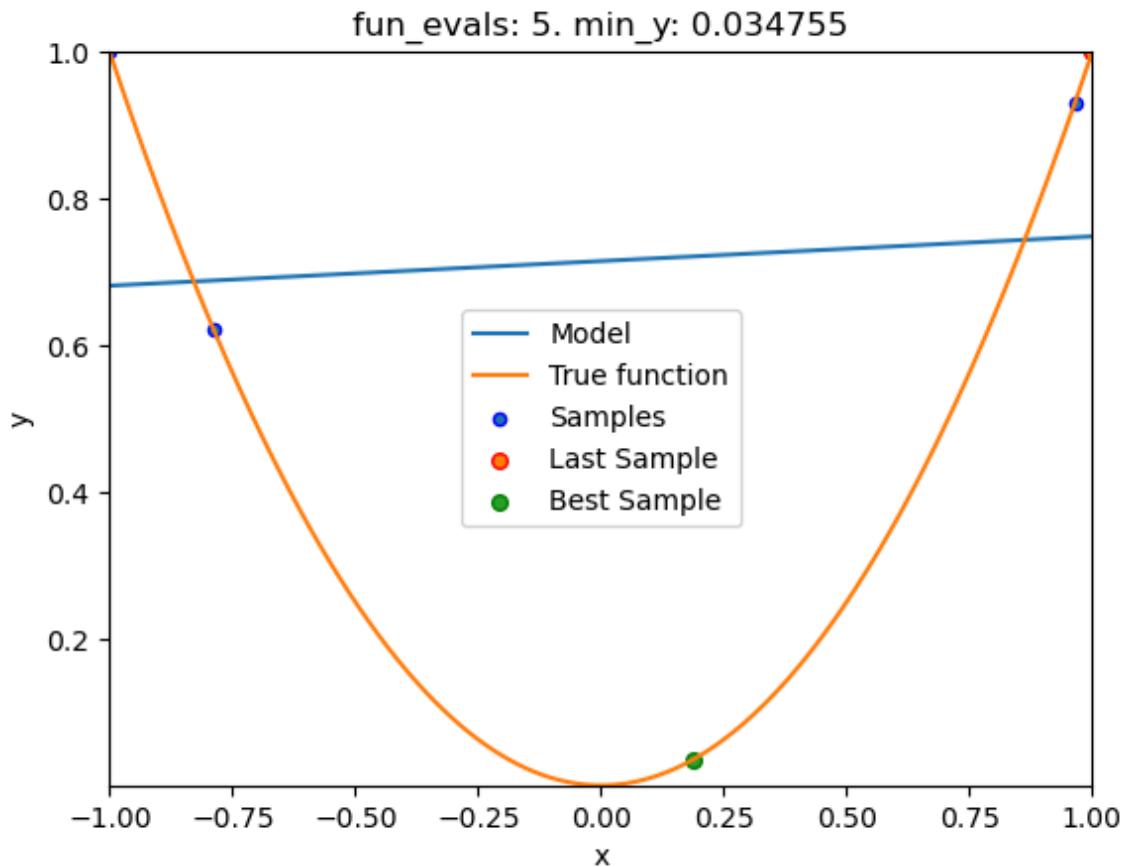
- Ridge Regression (Ridge) in Scikit-learn is a linear model that includes an L2 regularization term to prevent overfitting by penalizing large coefficients.
- This regularization helps the model generalize better to unseen data.
- When used as a surrogate model in optimization, Ridge Regression can approximate the objective function (e.g., the Sphere function) and guide the optimization process effectively.

```
In [16]: #1-dim sphere function fun_sphere using Sklearn Model Ridge
spot_1_R = spot.Spot(fun=fun,
                      fun_control=fun_control,
                      design_control=design_control,
                      surrogate = S_R)
spot_1_R.run()
spot_1_R.print_results()
spot_1_R.plot_progress(log_y=True)
spot_1_R.plot_model()
```

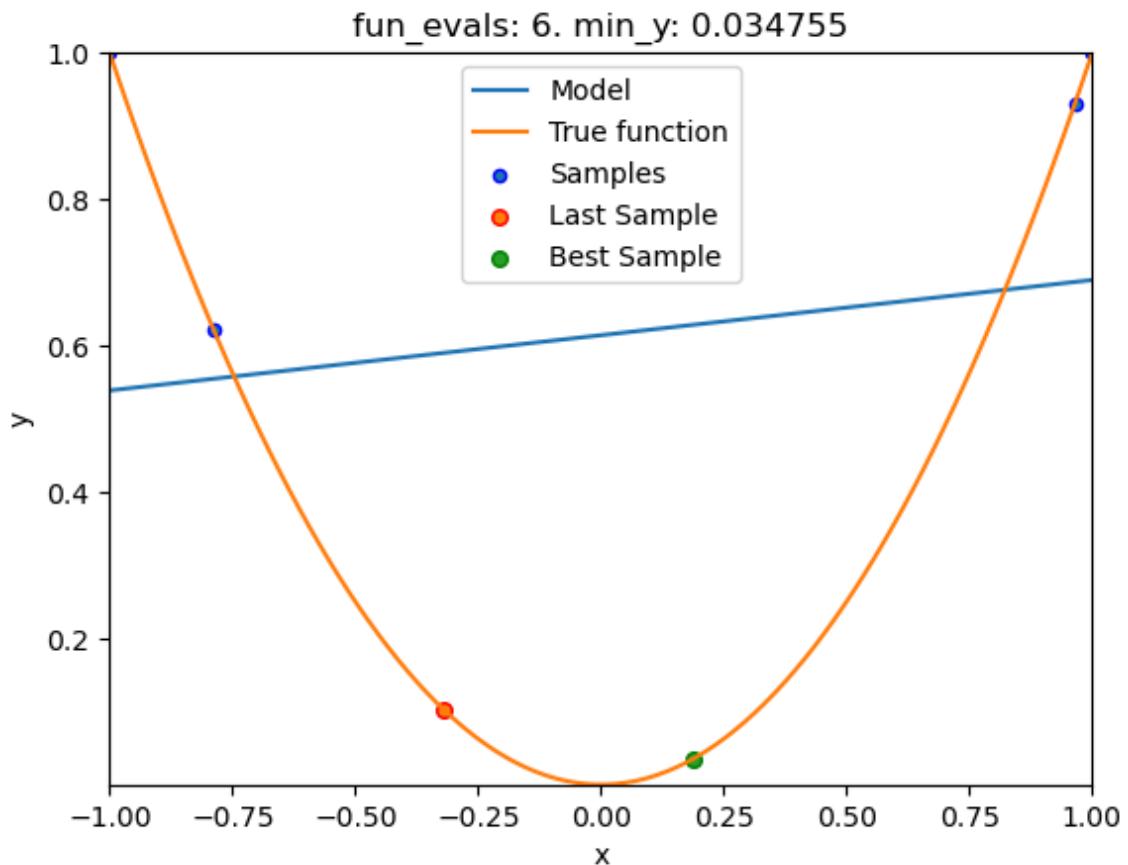




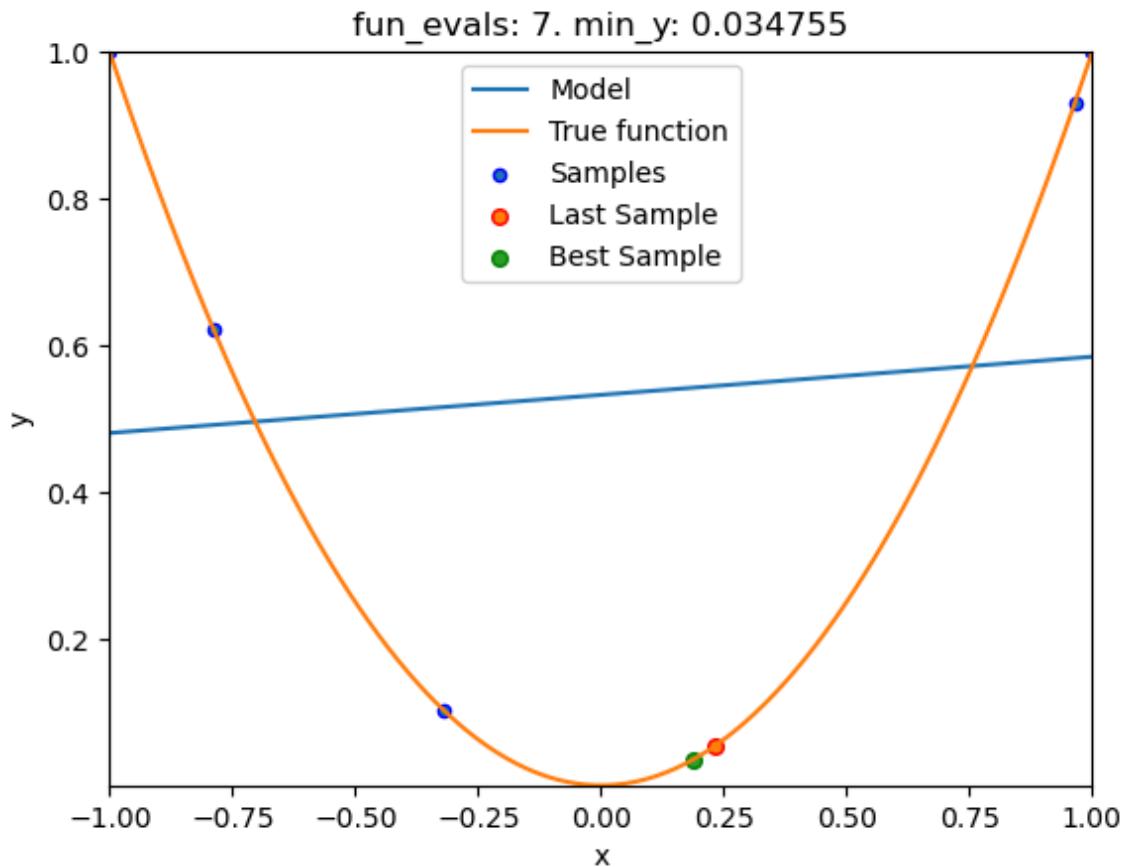
spotPython tuning: 0.03475493366922229 [#####-----] 40.00%



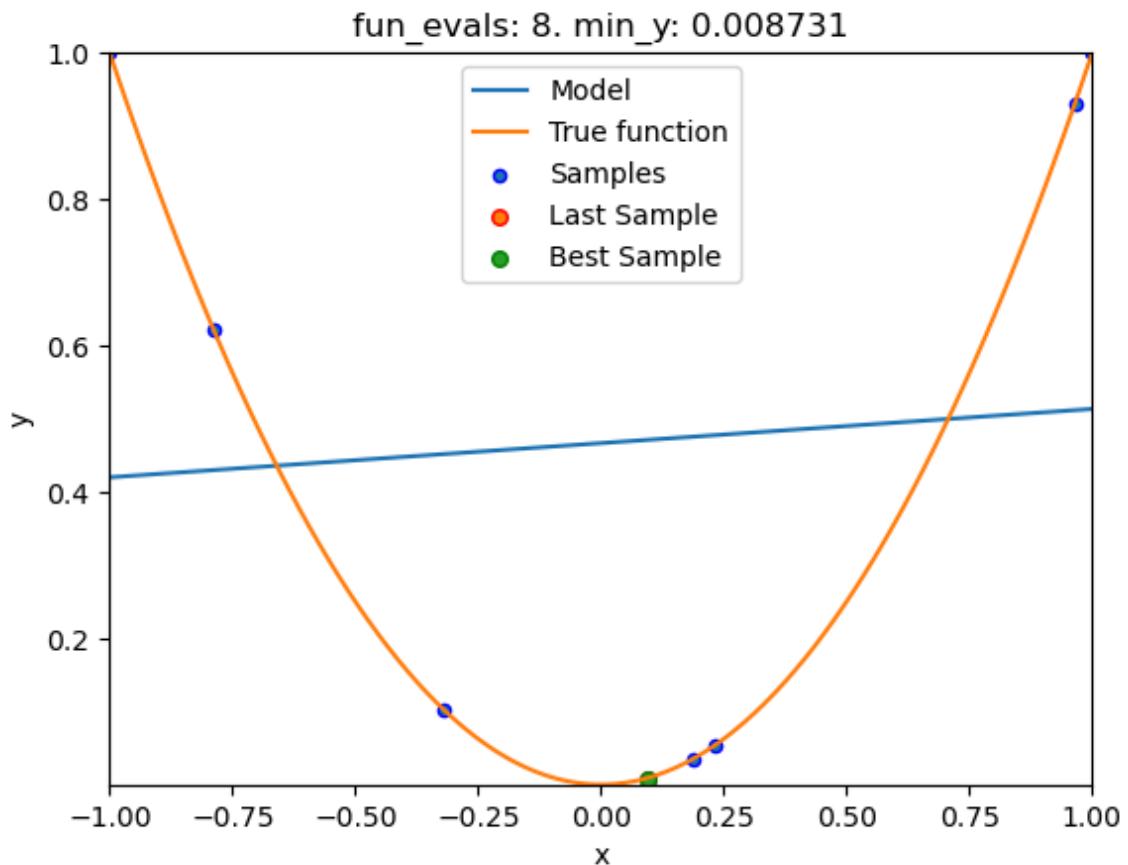
spotPython tuning: 0.03475493366922229 [#####-----] 50.00%



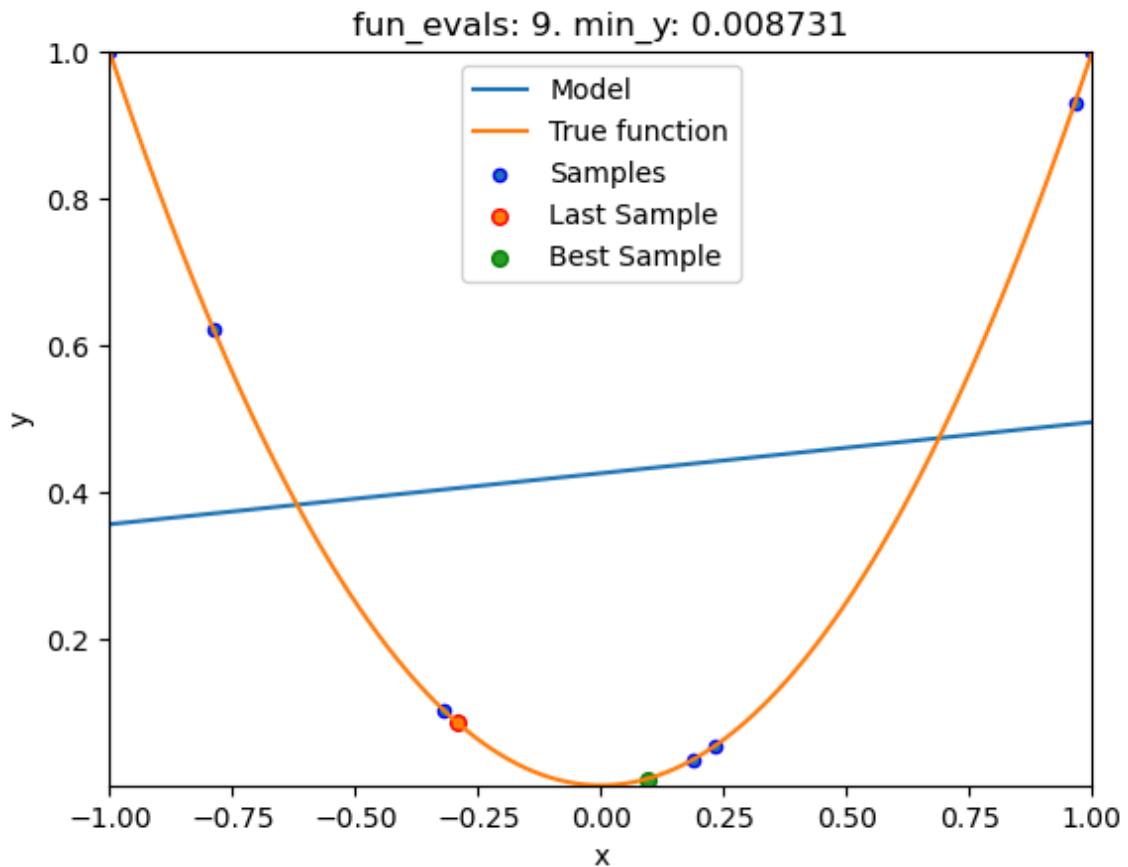
spotPython tuning: 0.03475493366922229 [#####----] 60.00%



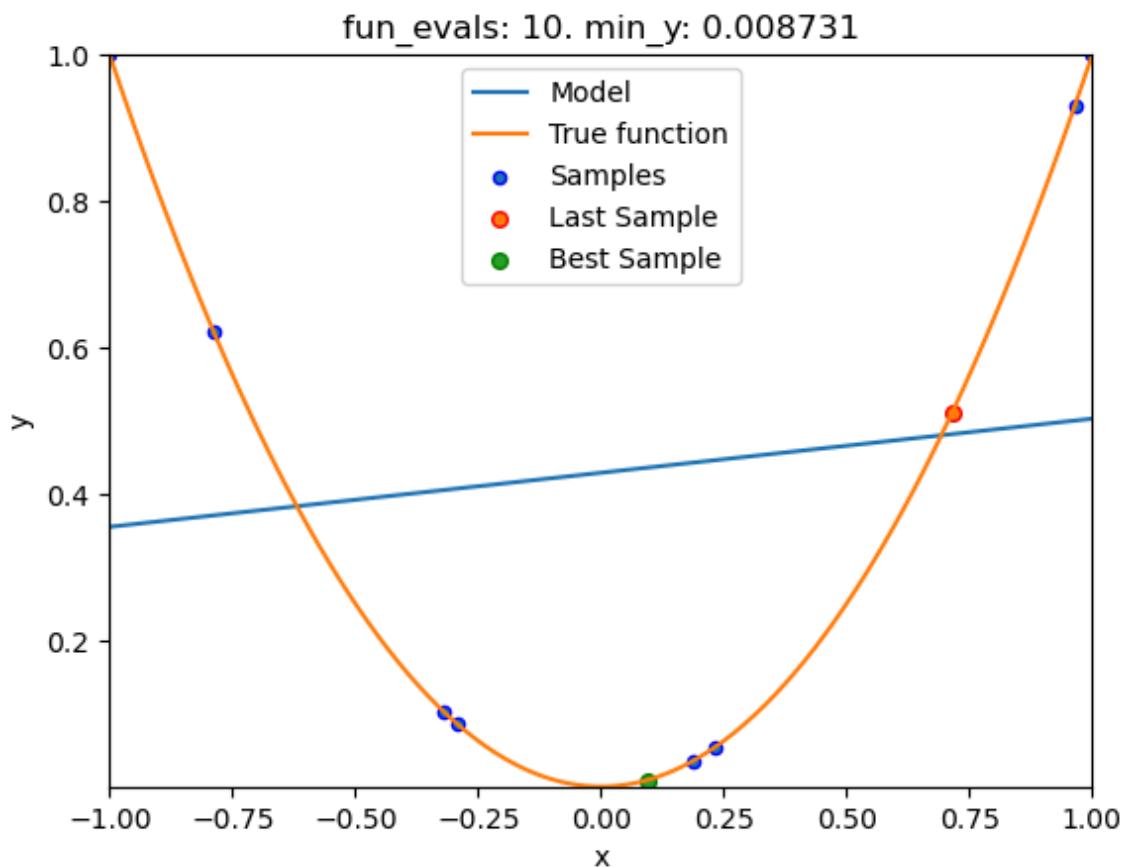
spotPython tuning: 0.03475493366922229 [#####---] 70.00%



spotPython tuning: 0.008730885505764131 [#####--] 80.00%



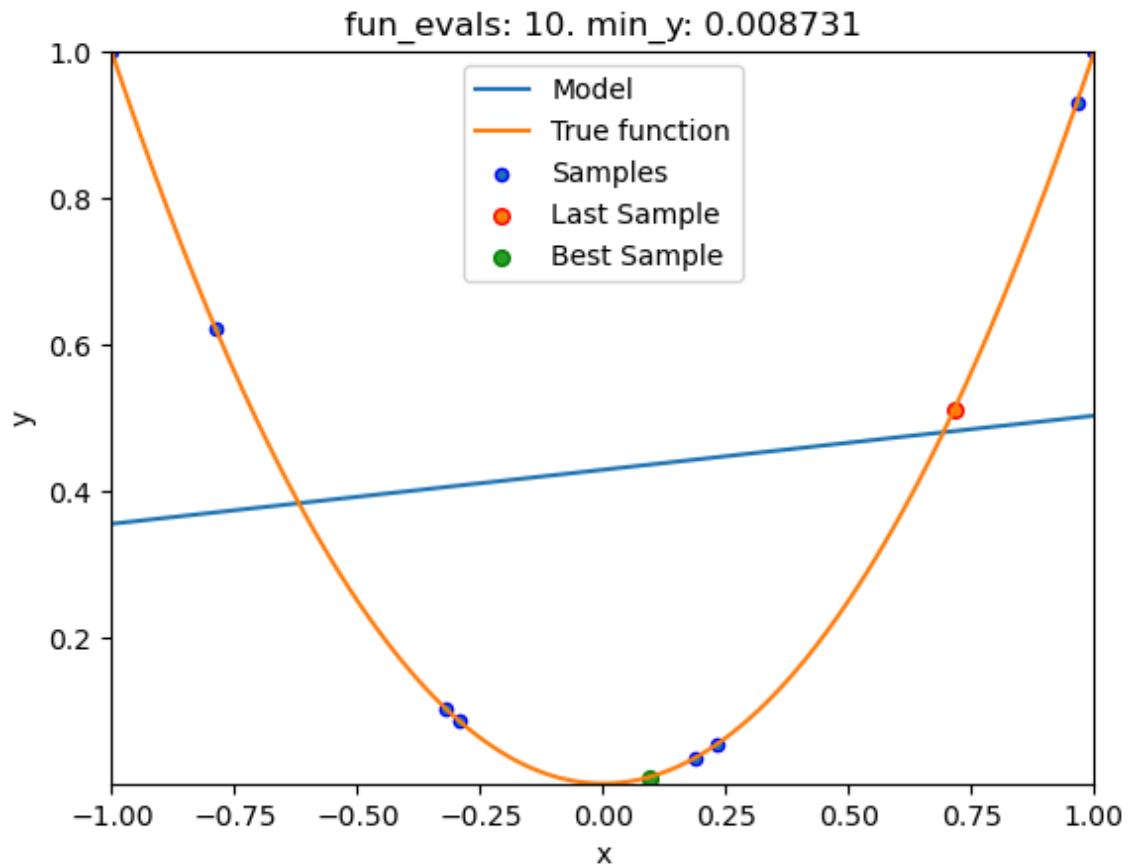
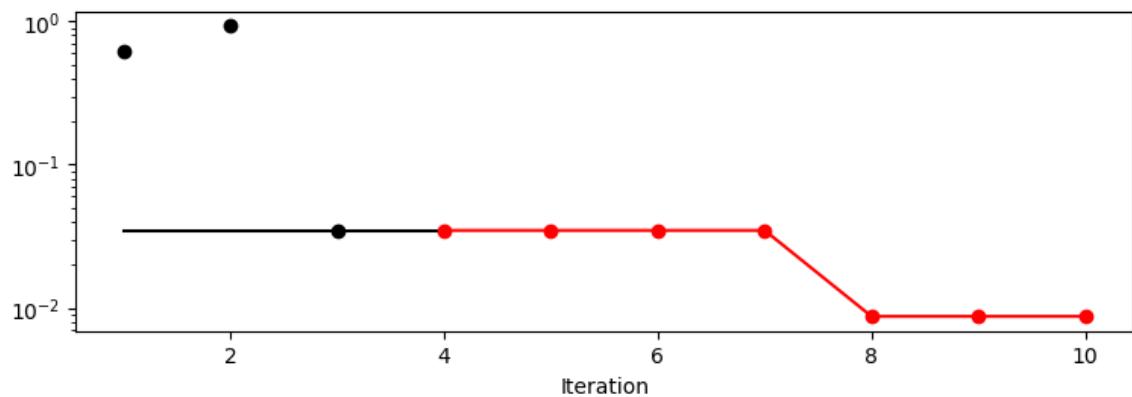
spotPython tuning: 0.008730885505764131 [#####--] 90.00%



```
spotPython tuning: 0.008730885505764131 [#####] 100.00% Done...
```

```
{'CHECKPOINT_PATH': 'runs/saved_models/',
'DATASET_PATH': 'data/',
'PREFIX': None,
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,
'core_model_name': None,
'counter': 10,
'data': None,
'data_dir': './data',
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': 10,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([-1]),
'max_surrogate_points': 30,
'max_time': inf,
'metric_params': {},
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': True,
'show_progress': True,
```

```
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': None,
'spot_writer': None,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 1.4901161193847656e-08,
'train': None,
'upper': array([1]),
'vername': None,
'vertype': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 0.008730885505764131
x0: 0.09343920754032609
```



2-dim Branin function fun_branin

```
In [17]: #2-dim Branin function fun_branin using spotPython's internal Kriging
fun = analytical().fun_branin
PREFIX = "BRANIN"
fun_control = fun_control_init(
    PREFIX=PREFIX,
    lower = np.array([-5,-0]),
    upper = np.array([10,15]),
    fun_evals=20,
    max_time=inf)

design_control = design_control_init(
    init_size=10)

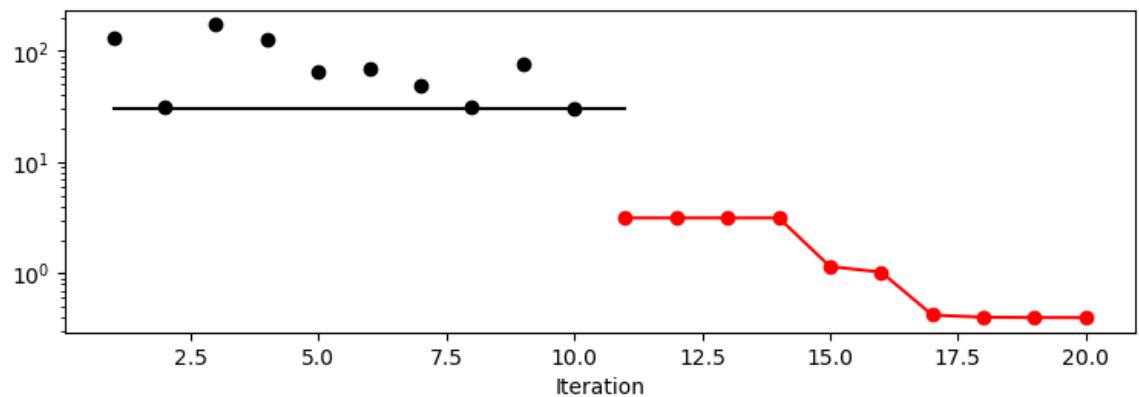
spot_2 = spot.Spot(fun=fun,
                    fun_control=fun_control,
                    design_control=design_control)
spot_2.run()
spot_2.print_results()
spot_2.plot_progress(log_y=True)
spot_2.surrogate.plot()
```

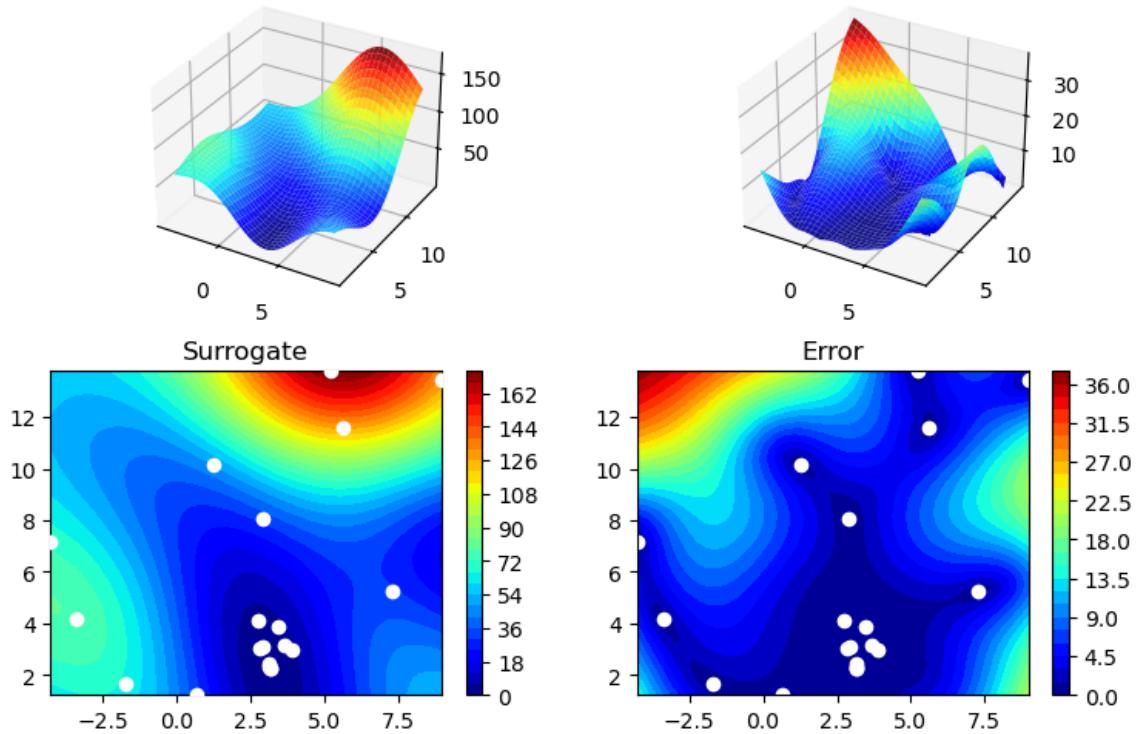
```
[rank: 0] Seed set to 123
```

```
Created spot_tensorboard_path: runs/spot_logs/BRANIN_c02-b03_2024-05-20_19-37-24 for SummaryWriter()
spotPython tuning: 3.146896546946783 [#####----] 55.00%
spotPython tuning: 3.146896546946783 [#####----] 60.00%
spotPython tuning: 3.146896546946783 [#####----] 65.00%
spotPython tuning: 3.146896546946783 [#####---] 70.00%
spotPython tuning: 1.1487866236306008 [#####----] 75.00%
spotPython tuning: 1.0240824742350494 [#####----] 80.00%
spotPython tuning: 0.4211249770550065 [#####----] 85.00%
spotPython tuning: 0.40198655236716263 [#####----] 90.00%
spotPython tuning: 0.3991783652663141 [#####----] 95.00%
spotPython tuning: 0.3991783652663141 [#####----] 100.00% Done...

{'CHECKPOINT_PATH': 'runs/saved_models/',
 'DATASET_PATH': 'data/',
 'PREFIX': 'BRANIN',
 'RESULTS_PATH': 'results/',
 'TENSORBOARD_PATH': 'runs/',
 '_L_in': None,
 '_L_out': None,
 '_torchmetric': None,
 'accelerator': 'auto',
 'converters': None,
 'core_model': None,
 'core_model_name': None,
 'counter': 20,
 'data': None,
 'data_dir': './data',
 'data_module': None,
 'data_set': None,
 'data_set_name': None,
 'db_dict_name': None,
 'design': None,
 'device': None,
 'devices': 1,
 'enable_progress_bar': False,
 'eval': None,
 'fun_evals': 20,
 'fun_repeats': 1,
 'horizon': None,
 'infill_criterion': 'y',
 'k_folds': 3,
 'log_graph': False,
 'log_level': 50,
 'loss_function': None,
 'lower': array([-5, 0]),
 'max_surrogate_points': 30,
 'max_time': inf,
 'metric_params': {},
 'metric_river': None,
 'metric_sklearn': None,
 'metric_sklearn_name': None,
 'metric_torch': None,
 'model_dict': {},
 'n_points': 1,
 'n_samples': None,
 'n_total': None,
 'noise': False,
 'num_workers': 0,
 'ocba_delta': 0,
```

```
'oml_grace_period': None,
'optimizer': None,
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/BRANIN_c02-b03_2024-05-20_19-3
7-24',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x1513fb031390>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([10, 15]),
'vername': None,
'vertype': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 0.3991783652663141
x0: 3.155010806689893
x1: 2.2852134198920027
```





1. A decision tree regressor: DecisionTreeRegressor

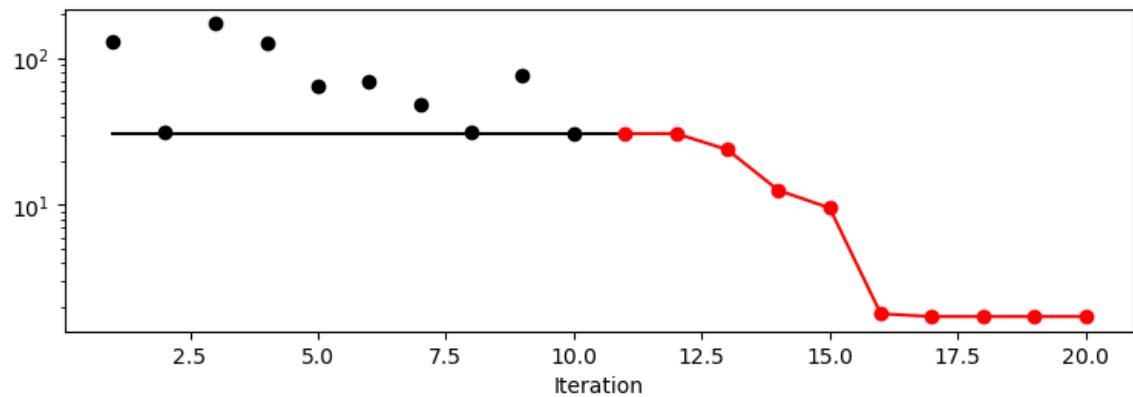
- Describe the surrogate model. Use the information from the [scikit-learn documentation](#).
- Use the surrogate as the model for optimization.

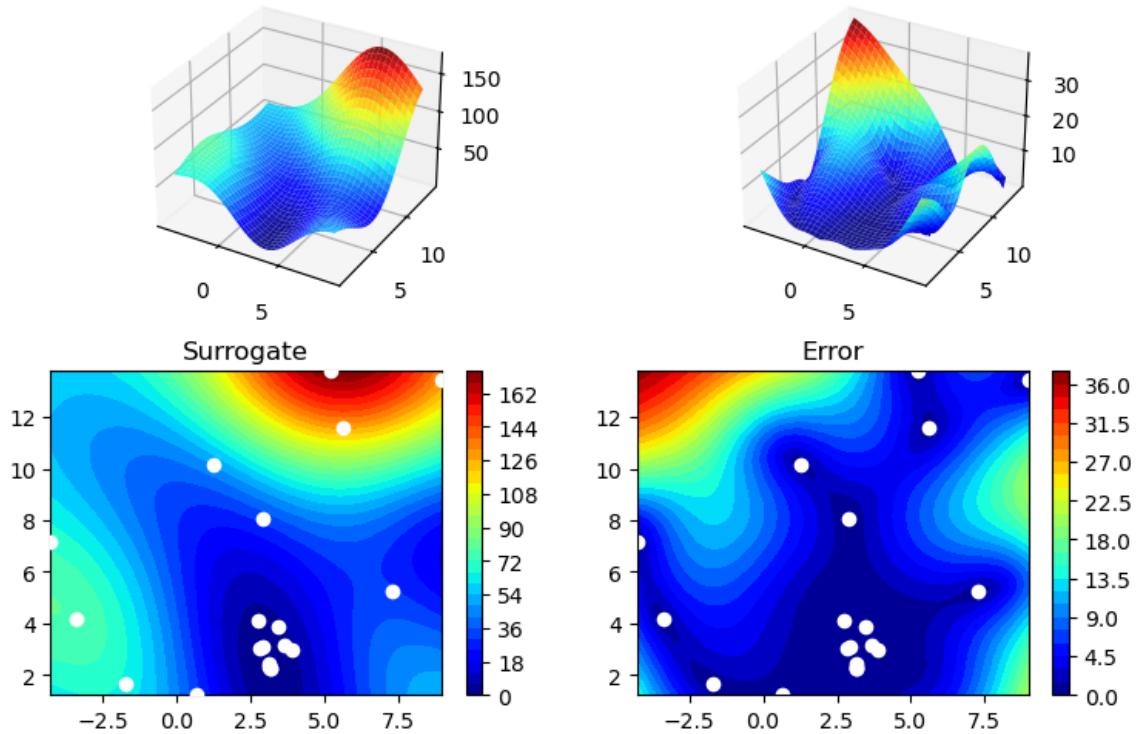
```
In [22]: #2-dim Branin function using Sklearn Model DecisionTreeRegressor
spot_2_DTR = spot.Spot(fun=fun,
                       fun_control=fun_control,
                       design_control=design_control,
                       surrogate = S_DTR)
spot_2_DTR.run()
spot_2_DTR.print_results()
spot_2_DTR.plot_progress(log_y=True)
spot_2_DTR.plot_model()
spot_2.surrogate.plot()
```

```
spotPython tuning: 30.69410528614059 [#####----] 55.00%
spotPython tuning: 30.69410528614059 [#####----] 60.00%
spotPython tuning: 23.864494553469576 [#####----] 65.00%
spotPython tuning: 12.573463667287577 [#####----] 70.00%
spotPython tuning: 9.541981201358062 [#####---] 75.00%
spotPython tuning: 1.8090034135005286 [#####---] 80.00%
spotPython tuning: 1.7303849957606374 [#####---] 85.00%
spotPython tuning: 1.7303849957606374 [#####---] 90.00%
spotPython tuning: 1.7303849957606374 [#####---] 95.00%
spotPython tuning: 1.7303849957606374 [#####---] 100.00% Done...
```

```
{'CHECKPOINT_PATH': 'runs/saved_models/',
'DATASET_PATH': 'data/',
'PREFIX': 'BRANIN',
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,
'core_model_name': None,
'counter': 20,
'data': None,
'data_dir': './data',
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': 20,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([-5, 0]),
'max_surrogate_points': 30,
'max_time': inf,
'metric_params': {},
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
```

```
'path': None,
'prep_model': None,
'prep_model_name': None,
'progress_file': None,
'save_model': False,
'seed': 123,
'show_batch_interval': 1000000,
'show_models': False,
'show_progress': True,
'shuffle': None,
'sigma': 0.0,
'spot_tensorboard_path': 'runs/spot_logs/BRANIN_c02-b03_2024-05-20_19-3
7-24',
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at
0x1513fb031390>,
'target_column': None,
'target_type': None,
'task': None,
'test': None,
'test_seed': 1234,
'test_size': 0.4,
'tolerance_x': 0,
'train': None,
'upper': array([10, 15]),
'var_name': None,
'var_type': ['num'],
'verbosity': 0,
'weight_coeff': 0.0,
'weights': 1.0,
'weights_entry': None}
min y: 1.7303849957606374
x0: 8.969889798848781
x1: 1.5213154614144528
```





2. A random forest regressor: RandomForestRegressor

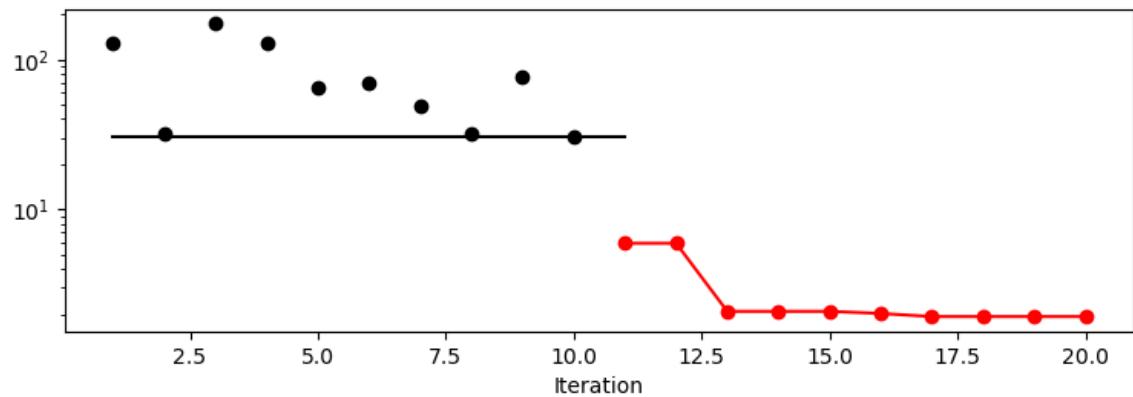
- Describe the surrogate model. Use the information from the [scikit-learn documentation](#).
- Use the surrogate as the model for optimization.

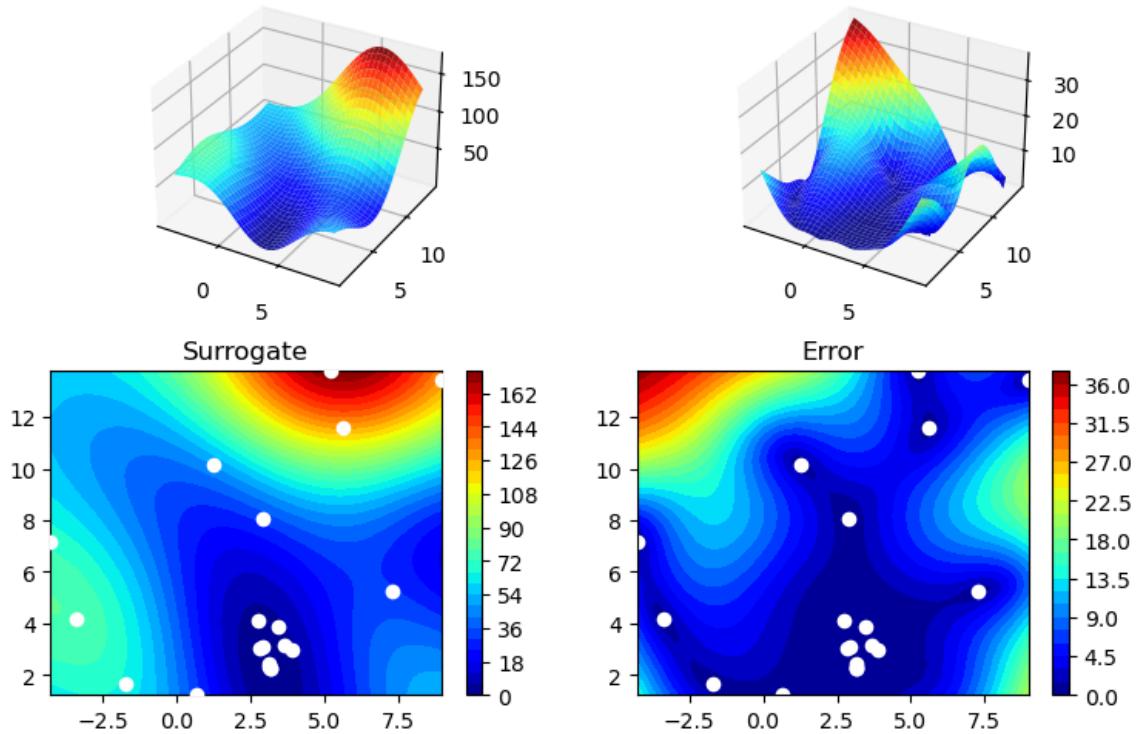
```
In [23]: #2-dim Branin function using Sklearn Model RandomForestRegressor
spot_2_RFR = spot.Spot(fun=fun,
                       fun_control=fun_control,
                       design_control=design_control,
                       surrogate = S_RFR)
spot_2_RFR.run()
spot_2_RFR.print_results()
spot_2_RFR.plot_progress(log_y=True)
spot_2_RFR.plot_model()
spot_2_RFR.surrogate.plot()
```

```
spotPython tuning: 5.946800592019464 [#####----] 55.00%
spotPython tuning: 5.946800592019464 [#####----] 60.00%
spotPython tuning: 2.093665266757097 [#####----] 65.00%
spotPython tuning: 2.093665266757097 [#####----] 70.00%
spotPython tuning: 2.093665266757097 [#####----] 75.00%
spotPython tuning: 2.0264784623082113 [#####----] 80.00%
spotPython tuning: 1.9316864959536755 [#####----] 85.00%
spotPython tuning: 1.9316864959536755 [#####----] 90.00%
spotPython tuning: 1.9316864959536755 [#####----] 95.00%
spotPython tuning: 1.9316864959536755 [#####----] 100.00% Done...
```

```
{'CHECKPOINT_PATH': 'runs/saved_models/',
'DATASET_PATH': 'data/',
'PREFIX': 'BRANIN',
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,
'core_model_name': None,
'counter': 20,
'data': None,
'data_dir': './data',
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': 20,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([-5, 0]),
'max_surrogate_points': 30,
'max_time': inf,
'metric_params': {},
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
```

```
'path': None,  
'prep_model': None,  
'prep_model_name': None,  
'progress_file': None,  
'save_model': False,  
'seed': 123,  
'show_batch_interval': 1000000,  
'show_models': False,  
'show_progress': True,  
'shuffle': None,  
'sigma': 0.0,  
'spot_tensorboard_path': 'runs/spot_logs/BRANIN_c02-b03_2024-05-20_19-3  
7-24',  
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at  
0x1513fb031390>,  
'target_column': None,  
'target_type': None,  
'task': None,  
'test': None,  
'test_seed': 1234,  
'test_size': 0.4,  
'tolerance_x': 0,  
'train': None,  
'upper': array([10, 15]),  
'var_name': None,  
'var_type': ['num'],  
'verbosity': 0,  
'weight_coeff': 0.0,  
'weights': 1.0,  
'weights_entry': None}  
min y: 1.9316864959536755  
x0: 2.5779270057913575  
x1: 2.5356828187404297
```





3. Ordinary least squares Linear Regression: LinearRegression

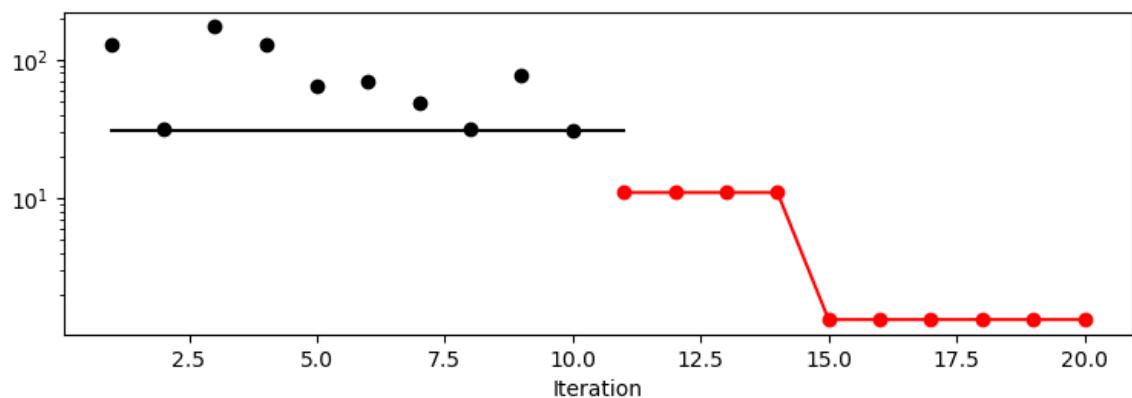
- Describe the surrogate model. Use the information from the [scikit-learn documentation](#).
- Use the surrogate as the model for optimization.

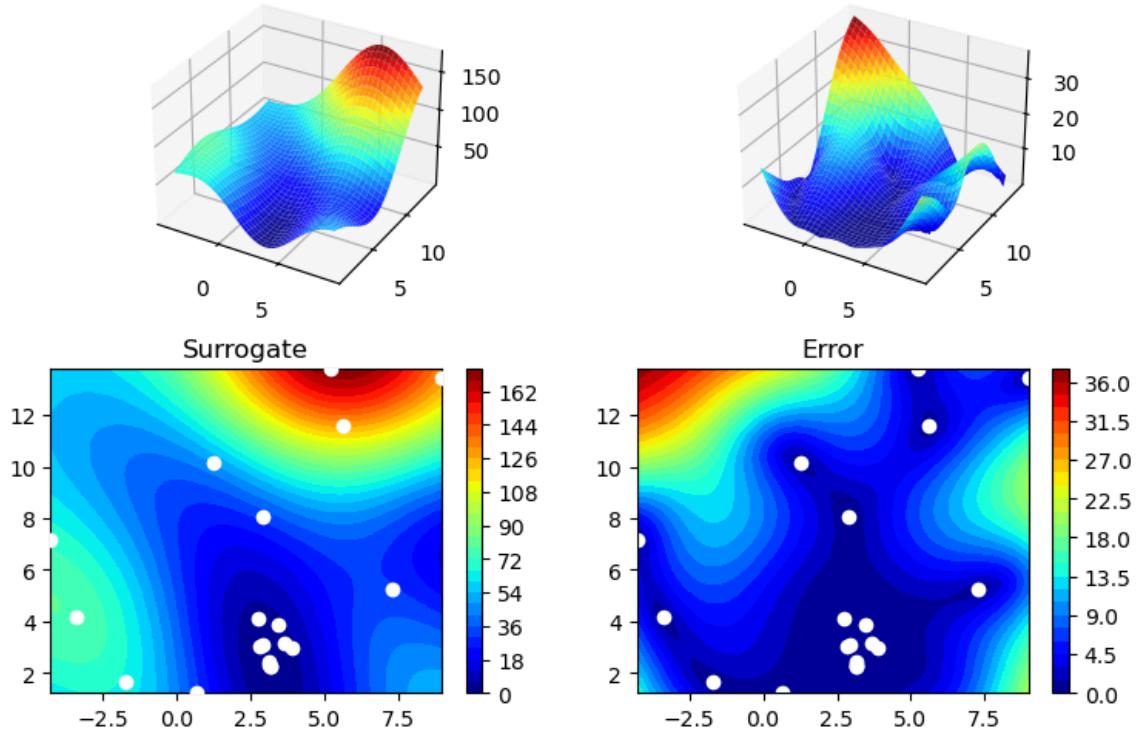
In [24]: #2-dim Branin function using Sklearn Model LinearRegression

```
spot_2_LR = spot.Spot(fun=fun,
                      fun_control=fun_control,
                      design_control=design_control,
                      surrogate = S_LR)
spot_2_LR.run()
spot_2_LR.print_results()
spot_2_LR.plot_progress(log_y=True)
spot_2_LR.plot_model()
spot_2.surrogate.plot()
```

```
spotPython tuning: 10.960827645483478 [#####----] 55.00%
spotPython tuning: 10.960827645483478 [#####----] 60.00%
spotPython tuning: 10.960827645483478 [#####----] 65.00%
spotPython tuning: 10.960827645483478 [#####----] 70.00%
spotPython tuning: 1.3263745845108854 [#####----] 75.00%
spotPython tuning: 1.3263745845108854 [#####----] 80.00%
spotPython tuning: 1.3263745845108854 [#####----] 85.00%
spotPython tuning: 1.3263745845108854 [#####----] 90.00%
spotPython tuning: 1.3263745845108854 [#####----] 95.00%
spotPython tuning: 1.3263745845108854 [#####----] 100.00% Done...
{'CHECKPOINT_PATH': 'runs/saved_models/',
 'DATASET_PATH': 'data/',
 'PREFIX': 'BRANIN',
 'RESULTS_PATH': 'results/',
 'TENSORBOARD_PATH': 'runs/',
 '_L_in': None,
 '_L_out': None,
 '_torchmetric': None,
 'accelerator': 'auto',
 'converters': None,
 'core_model': None,
 'core_model_name': None,
 'counter': 20,
 'data': None,
 'data_dir': './data',
 'data_module': None,
 'data_set': None,
 'data_set_name': None,
 'db_dict_name': None,
 'design': None,
 'device': None,
 'devices': 1,
 'enable_progress_bar': False,
 'eval': None,
 'fun_evals': 20,
 'fun_repeats': 1,
 'horizon': None,
 'infill_criterion': 'y',
 'k_folds': 3,
 'log_graph': False,
 'log_level': 50,
 'loss_function': None,
 'lower': array([-5, 0]),
 'max_surrogate_points': 30,
 'max_time': inf,
 'metric_params': {},
 'metric_river': None,
 'metric_sklearn': None,
 'metric_sklearn_name': None,
 'metric_torch': None,
 'model_dict': {},
 'n_points': 1,
 'n_samples': None,
 'n_total': None,
 'noise': False,
 'num_workers': 0,
 'ocba_delta': 0,
 'oml_grace_period': None,
 'optimizer': None,
```

```
'path': None,  
'prep_model': None,  
'prep_model_name': None,  
'progress_file': None,  
'save_model': False,  
'seed': 123,  
'show_batch_interval': 1000000,  
'show_models': False,  
'show_progress': True,  
'shuffle': None,  
'sigma': 0.0,  
'spot_tensorboard_path': 'runs/spot_logs/BRANIN_c02-b03_2024-05-20_19-3  
7-24',  
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at  
0x1513fb031390>,  
'target_column': None,  
'target_type': None,  
'task': None,  
'test': None,  
'test_seed': 1234,  
'test_size': 0.4,  
'tolerance_x': 0,  
'train': None,  
'upper': array([10, 15]),  
'var_name': None,  
'var_type': ['num'],  
'verbosity': 0,  
'weight_coeff': 0.0,  
'weights': 1.0,  
'weights_entry': None}  
min y: 1.3263745845108854  
x0: -2.872730773493426  
x1: 10.874313833535739
```





4. Linear least squares with l2 regularization: Ridge

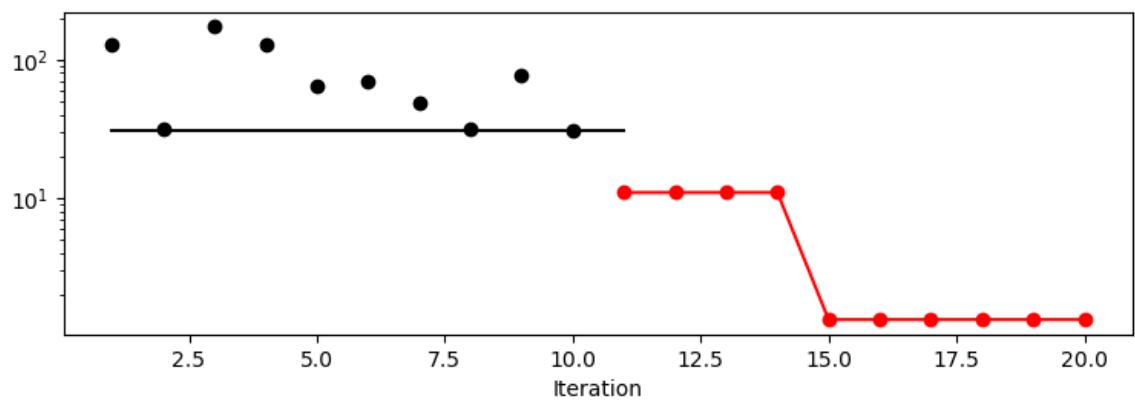
- Describe the surrogate model. Use the information from the [scikit-learn documentation](#).
- Use the surrogate as the model for optimization.

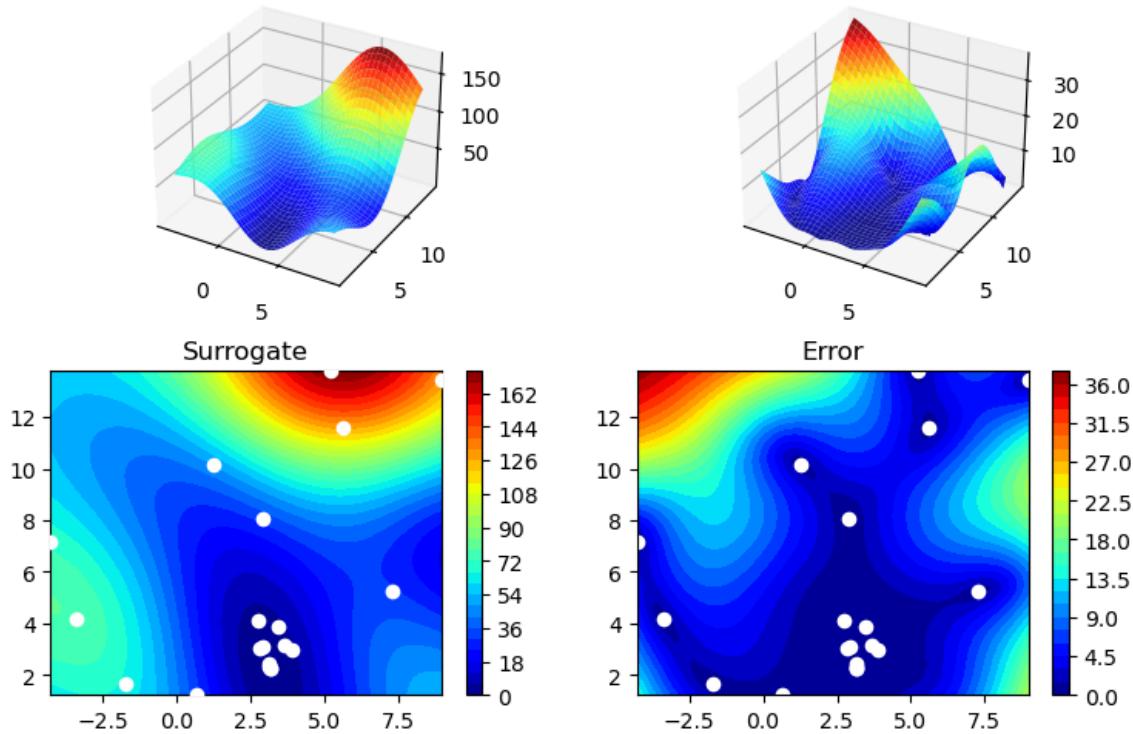
```
In [25]: #2-dim Branin function using Sklearn Model Ridge
spot_2_R = spot.Spot(fun=fun,
                      fun_control=fun_control,
                      design_control=design_control,
                      surrogate = S_R)
spot_2_R.run()
spot_2_R.print_results()
spot_2_R.plot_progress(log_y=True)
spot_2_R.plot_model()
spot_2_R.surrogate.plot()
```

```
spotPython tuning: 10.960827645483478 [#####----] 55.00%
spotPython tuning: 10.960827645483478 [#####----] 60.00%
spotPython tuning: 10.960827645483478 [#####----] 65.00%
spotPython tuning: 10.960827645483478 [#####----] 70.00%
spotPython tuning: 1.3263745845108854 [#####----] 75.00%
spotPython tuning: 1.3263745845108854 [#####----] 80.00%
spotPython tuning: 1.3263745845108854 [#####----] 85.00%
spotPython tuning: 1.3263745845108854 [#####----] 90.00%
spotPython tuning: 1.3263745845108854 [#####----] 95.00%
spotPython tuning: 1.3263745845108854 [#####----] 100.00% Done...
```

```
{'CHECKPOINT_PATH': 'runs/saved_models/',
'DATASET_PATH': 'data/',
'PREFIX': 'BRANIN',
'RESULTS_PATH': 'results/',
'TENSORBOARD_PATH': 'runs/',
'_L_in': None,
'_L_out': None,
'_torchmetric': None,
'accelerator': 'auto',
'converters': None,
'core_model': None,
'core_model_name': None,
'counter': 20,
'data': None,
'data_dir': './data',
'data_module': None,
'data_set': None,
'data_set_name': None,
'db_dict_name': None,
'design': None,
'device': None,
'devices': 1,
'enable_progress_bar': False,
'eval': None,
'fun_evals': 20,
'fun_repeats': 1,
'horizon': None,
'infill_criterion': 'y',
'k_folds': 3,
'log_graph': False,
'log_level': 50,
'loss_function': None,
'lower': array([-5, 0]),
'max_surrogate_points': 30,
'max_time': inf,
'metric_params': {},
'metric_river': None,
'metric_sklearn': None,
'metric_sklearn_name': None,
'metric_torch': None,
'model_dict': {},
'n_points': 1,
'n_samples': None,
'n_total': None,
'noise': False,
'num_workers': 0,
'ocba_delta': 0,
'oml_grace_period': None,
'optimizer': None,
```

```
'path': None,  
'prep_model': None,  
'prep_model_name': None,  
'progress_file': None,  
'save_model': False,  
'seed': 123,  
'show_batch_interval': 1000000,  
'show_models': False,  
'show_progress': True,  
'shuffle': None,  
'sigma': 0.0,  
'spot_tensorboard_path': 'runs/spot_logs/BRANIN_c02-b03_2024-05-20_19-3  
7-24',  
'spot_writer': <torch.utils.tensorboard.writer.SummaryWriter object at  
0x1513fb031390>,  
'target_column': None,  
'target_type': None,  
'task': None,  
'test': None,  
'test_seed': 1234,  
'test_size': 0.4,  
'tolerance_x': 0,  
'train': None,  
'upper': array([10, 15]),  
'var_name': None,  
'var_type': ['num'],  
'verbosity': 0,  
'weight_coeff': 0.0,  
'weights': 1.0,  
'weights_entry': None}  
min y: 1.3263745845108854  
x0: -2.872730773493426  
x1: 10.874313833535739
```





5. Comparison of Surrogates

- Use the following two objective functions
 1. the 1-dim sphere function `fun_sphere` and
 2. the two-dim Branin function `fun_branin`:
 for a comparison of the performance of the five different surrogates:
 - `spotPython`'s internal Kriging
 - `DecisionTreeRegressor`
 - `RandomForestRegressor`
 - `linear_model.LinearRegression`
 - `linear_model.Ridge`.
- Generate a table with the results (number of function evaluations, best function value, and best parameter vector) for each surrogate and each function as shown in @tbl-results.

surrogate	fun	fun_evals	max_time	x_0
Kriging	fun_sphere	10	inf	0.003770156
Kriging	fun_branin	10	inf	3.1549178 2.2859

surrogate	fun	fun_evals	max_time	x_0
DecisionTreeRegressor	fun_sphere	10	inf	-0.026685
DecisionTreeRegressor	fun_branin	10	inf	3.479820933 0.995
RandomForestRegressor	fun_sphere	10	inf	-0.000737464
RandomForestRegressor	fun_branin	10	inf	3.018658 5.8
LinearRegression	fun_sphere	10	inf	0.093439208
LinearRegression	fun_branin	10	inf	-2.87273077 10.8
Ridge	fun_sphere	10	inf	0.94339208
Ridge	fun_branin	10	inf	-2.8727307 10.87

Conclusion

The table compares the performance of different surrogate models, specifically Kriging, Decision Tree Regressor, Random Forest Regressor, Linear Regression, and Ridge Regression. The models are evaluated on two objective functions, fun_sphere and fun_branin. The evaluation considers the number of function evaluations, the best function value, and the best parameter vector.

Here are some general observations from the table:

Kriging appears to be the most performant model overall. It achieves the best function value for both fun_sphere and fun_branin with a moderate number of function evaluations. Decision Tree Regressor and Random Forest Regressor achieve similar performance. They find close-to-optimal solutions for both functions but require more function evaluations than Kriging. Linear Regression and Ridge Regression show similar performance. They perform moderately well but do not reach the optimal value for either function. Overall, the choice of surrogate model depends on the specific needs of the application. If finding the absolute best solution is critical, Kriging might be the best choice despite its higher computational cost. If computational efficiency is a priority, then Decision Tree Regressor or Random Forest Regressor could be reasonable alternatives, even if they might not reach the absolute optimal value. Linear Regression and Ridge Regression might be suitable for applications where a moderately good solution is sufficient.