Introduction to Sequential Parameter Optimization

Expected Improvement

1 Example: Spot and the 1-dim Sphere Function

```
import numpy as np
from math import inf
from spotPython.fun.objectivefunctions import analytical
from spotPython.spot import spot
from scipy.optimize import shgo
from scipy.optimize import direct
from scipy.optimize import differential_evolution
import matplotlib.pyplot as plt
```

The Objective Function: 1-dim Sphere

- The `spotPython` package provides several classes of objective functions.
- We will use an analytical objective function, i.e., a function that can be described by a (closed) formula:

$$f(x) = x^2$$

```
fun = analytical().fun_sphere
```

```
fun = analytical().fun_sphere
fun_control = {"sigma": 0,
               "seed": 123}
```

- The size of the `lower` bound vector determines the problem dimension.
- Here we will use `np.array([-1])`, i.e., a one-dim function.

```
spot_1 = spot.Spot(fun=fun,
                   lower = np.array([-1]),
                   upper = np.array([1]))

spot_1.run()
```

```
<spotPython.spot.spot.Spot at 0x14a95fe80>
```

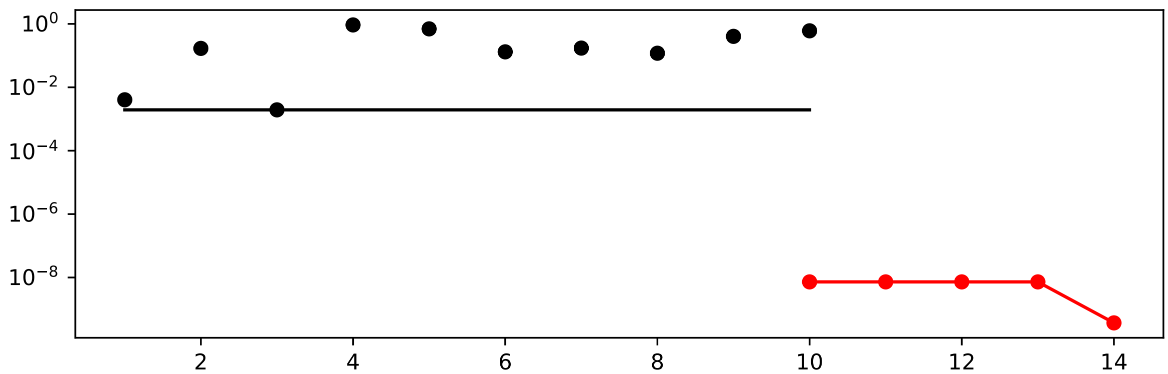
Results

```
spot_1.print_results()
```

```
min y: 3.696886711914087e-10
x0: 1.922728975158508e-05
```

```
[['x0', 1.922728975158508e-05]]
```

```
spot_1.plot_progress(log_y=True)
```

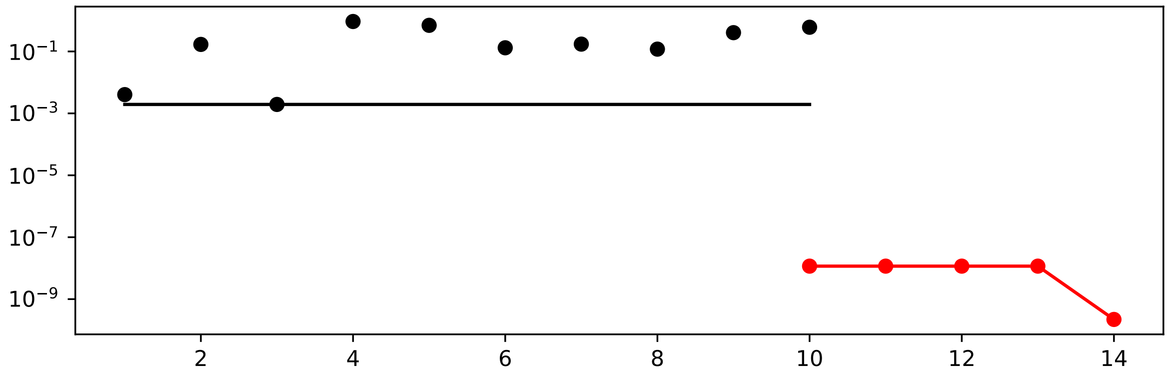


2 Same, but with EI as infill_criterion

```
spot_1_ei = spot.Spot(fun=fun,
                      lower = np.array([-1]),
                      upper = np.array([1]),
                      infill_criterion = "ei")
spot_1_ei.run()
```

```
<spotPython.spot.spot.Spot at 0x16c1596c0>
```

```
spot_1_ei.plot_progress(log_y=True)
```




```
spot_1_ei.print_results()
```

```
min y: 2.207887258868953e-10
x0: 1.4858961130809088e-05
```

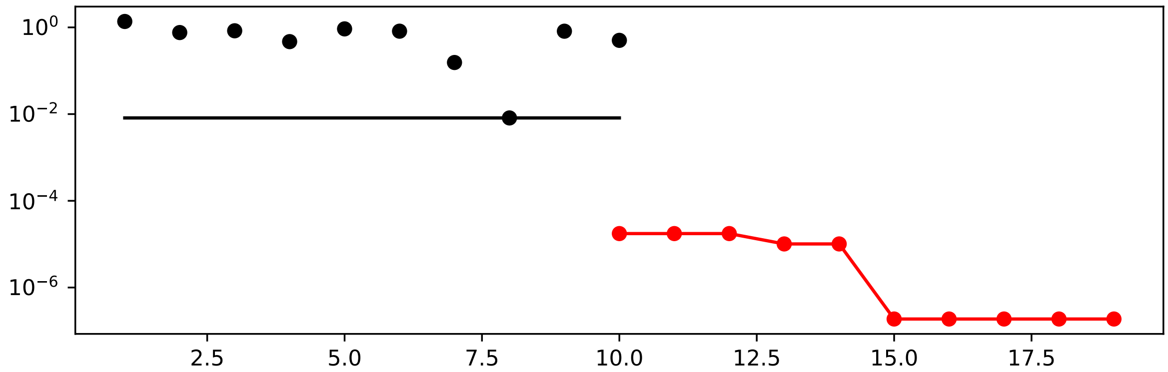
```
[['x0', 1.4858961130809088e-05]]
```

Non-isotropic Kriging

```
spot_2_ei_noniso = spot.Spot(fun=fun,
                   lower = np.array([-1, -1]),
                   upper = np.array([1, 1]),
                   fun_evals = 20,
                   fun_repeats = 1,
                   max_time = inf,
                   noise = False,
                   tolerance_x = np.sqrt(np.spacing(1)),
                   var_type=["num"],
                   infill_criterion = "ei",
                   n_points = 1,
                   seed=123,
                   log_level = 50,
                   show_models=True,
                   fun_control = fun_control,
                   design_control={"init_size": 10,
                                   "repeats": 1},
                   surrogate_control={"noise": False,
                                      "cod_type": "norm",
                                      "min_theta": -4,
                                      "max_theta": 3,
                                      "n_theta": 2,
                                      "model_optimizer": differential_evolution,
                                      "model_fun_evals": 1000,
                                      })
spot_2_ei_noniso.run()
```

```
<spotPython.spot.spot.Spot at 0x17a6c4af0>
```

```
spot_2_ei_noniso.plot_progress(log_y=True)
```

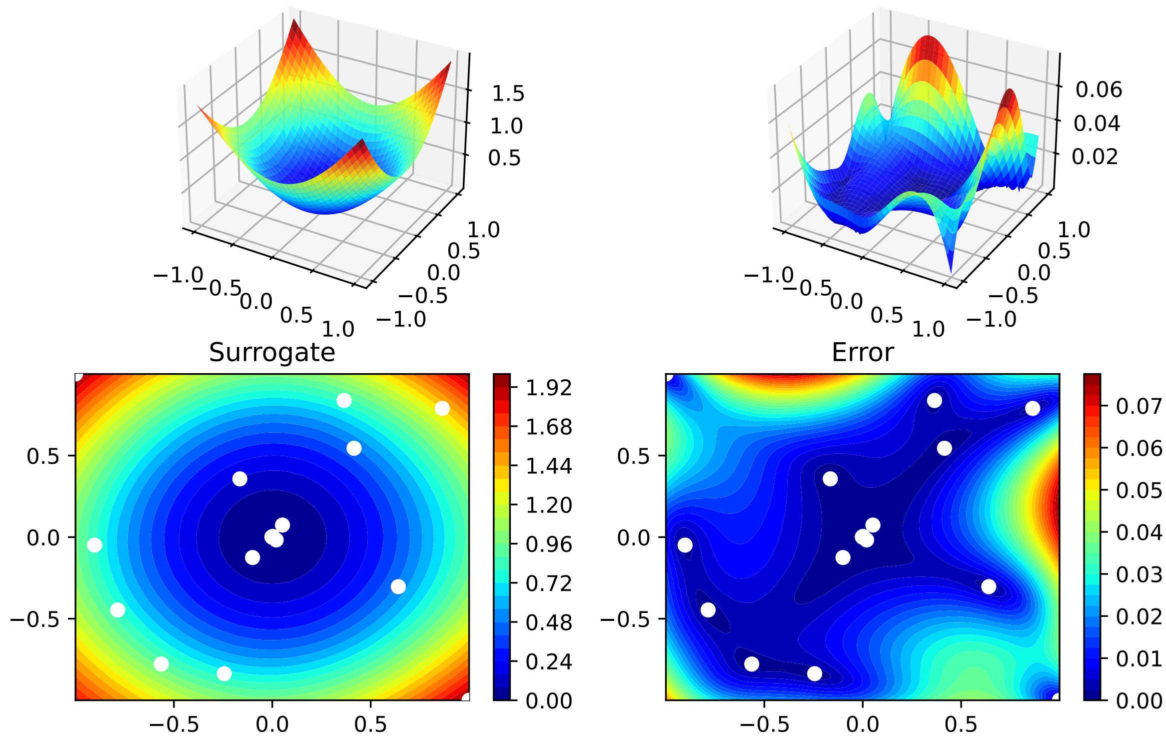



```
spot_2_ei_noniso.print_results()
```

```
min y: 1.8779971830281702e-07
x0: -0.0002783721390529846
x1: 0.0003321274913371111
```

```
[['x0', -0.0002783721390529846], ['x1', 0.0003321274913371111]]
```

```
spot_2_ei_noniso.surrogate.plot()
```



Surrogate
Error

Using `sklearn` Surrogates

The spot Loop

The `spot` loop consists of the following steps:

1. Init: Build initial design X
2. Evaluate initial design on real objective f: $y = f(X)$
3. Build surrogate: $S = S(X, y)$
4. Optimize on surrogate: $X_0 = \text{optimize}(S)$
5. Evaluate on real objective: $y_0 = f(X_0)$
6. Impute (Infill) new points: $X = X \cup X_0$, $y = y \cup y_0$.
7. Got 3.

The `spot` loop is implemented in `R` as follows:

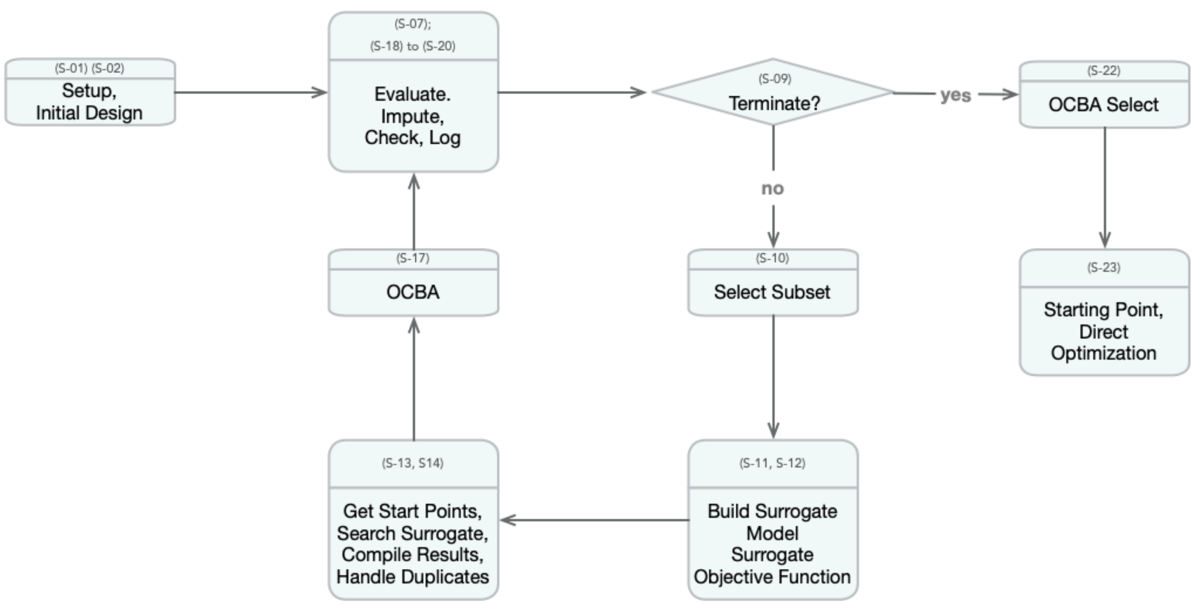


Figure 1: Visual representation of the model based search with SPOT. Taken from: Bartz-Beielstein, T., and Zaefferer, M. Hyperparameter tuning approaches. In Hyperparameter Tuning for Machine and Deep Learning with R - A Practical Guide, E. Bartz, T. Bartz-Beielstein, M. Zaefferer, and O. Mersmann, Eds. Springer, 2022, ch. 4, pp. 67–114.

spot: The Initial Model

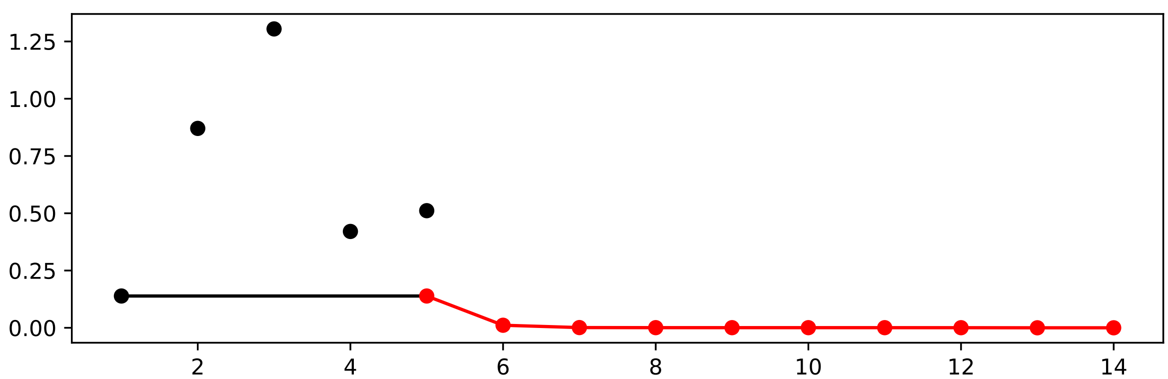
Example: Modifying the initial design size

This is the “Example: Modifying the initial design size” from Chapter 4.5.1 in [bart21i].

```
spot_ei = spot.Spot(fun=fun,
                lower = np.array([-1,-1]),
                upper= np.array([1,1]),
                design_control={"init_size": 5})
spot_ei.run()
```

```
<spotPython.spot.spot.Spot at 0x2bbae2fb0>
```

```
spot_ei.plot_progress()
```




```
np.min(spot_1.y), np.min(spot_ei.y)
```

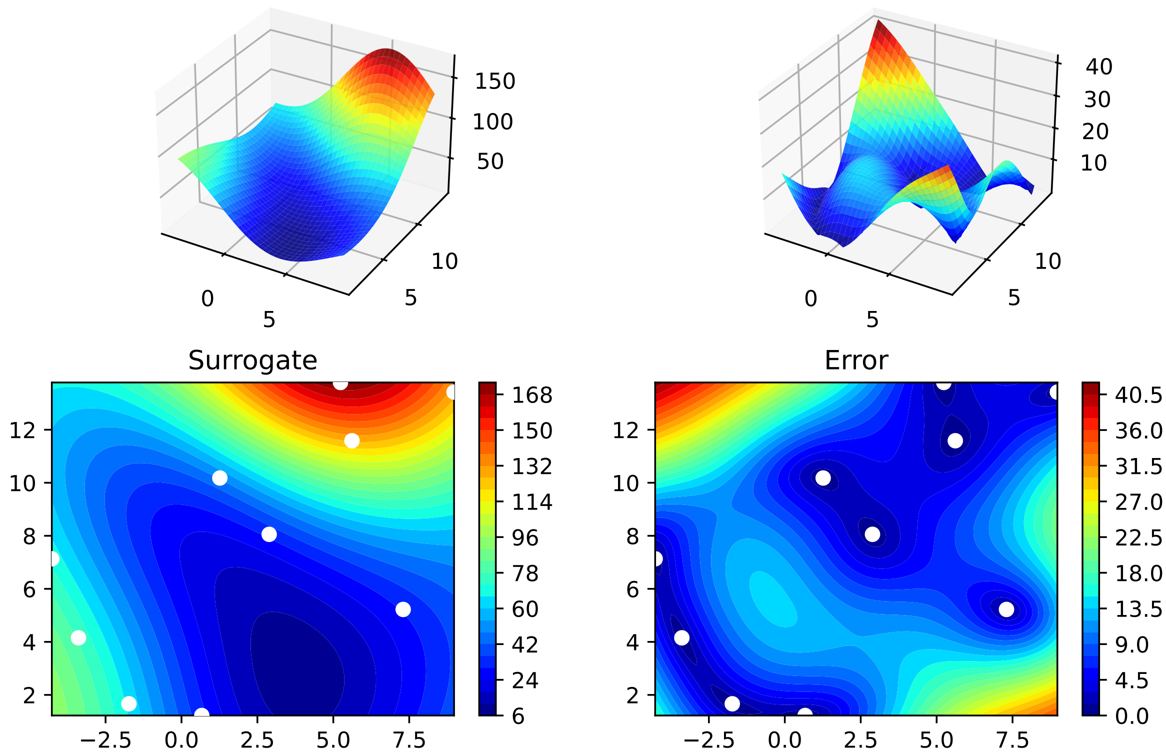
```
(3.696886711914087e-10, 1.7928640814182596e-05)
```

1. Init: Build Initial Design

```
from spotPython.design.spacefilling import spacefilling
from spotPython.build.kriging import Kriging
from spotPython.fun.objectivefunctions import analytical
gen = spacefilling(2)
rng = np.random.RandomState(1)
lower = np.array([-5,-0])
upper = np.array([10,15])
fun = analytical().fun_branin
fun_control = {"sigma": 0,
               "seed": 123}

X = gen.scipy_lhd(10, lower=lower, upper = upper)
print(X)
y = fun(X, fun_control=fun_control)
print(y)
```

```
[[ 8.97647221 13.41926847]
 [ 0.66946019  1.22344228]
 [ 5.23614115 13.78185824]
 [ 5.6149825  11.5851384 ]
 [-1.72963184  1.66516096]
 [-4.26945568  7.1325531 ]
 [ 1.26363761 10.17935555]
 [ 2.88779942  8.05508969]
 [-3.39111089  4.15213772]
 [ 7.30131231  5.22275244]]
[128.95676449  31.73474356 172.89678121 126.71295908  64.34349975
  70.16178611  48.71407916  31.77322887  76.91788181  30.69410529]
```

```
S = Kriging(name='kriging',  seed=123)
S.fit(X, y)
S.plot()
```




```python
gen = spacefilling(2, seed=123)
X0 = gen.scipy_lhd(3)
gen = spacefilling(2, seed=345)
X1 = gen.scipy_lhd(3)
X2 = gen.scipy_lhd(3)
gen = spacefilling(2, seed=123)
X3 = gen.scipy_lhd(3)
X0, X1, X2, X3
```

```
(array([[0.77254938, 0.31539299],
        [0.59321338, 0.93854273],
        [0.27469803, 0.3959685 ]]),
 array([[0.78373509, 0.86811887],
        [0.06692621, 0.6058029 ],
        [0.41374778, 0.00525456]]),
 array([[0.121357  , 0.69043832],
        [0.41906219, 0.32838498],
        [0.86742658, 0.52910374]]),
```

```
array([[0.77254938, 0.31539299],
       [0.59321338, 0.93854273],
       [0.27469803, 0.3959685 ]]))
```

2. Evaluate

3. Build Surrogate

A Simple Predictor

The code below shows how to use a simple model for prediction.

- Assume that only two (very costly) measurements are available:
  1. f(0) = 0.5
  2. f(2) = 2.5
- We are interested in the value at $x_0 = 1$, i.e., $f(x_0 = 1)$, but cannot run an additional, third experiment.

```
from sklearn import linear_model
X = np.array([[0], [2]])
y = np.array([0.5, 2.5])
S_lm = linear_model.LinearRegression()
S_lm = S_lm.fit(X, y)
X0 = np.array([[1]])
y0 = S_lm.predict(X0)
print(y0)
```

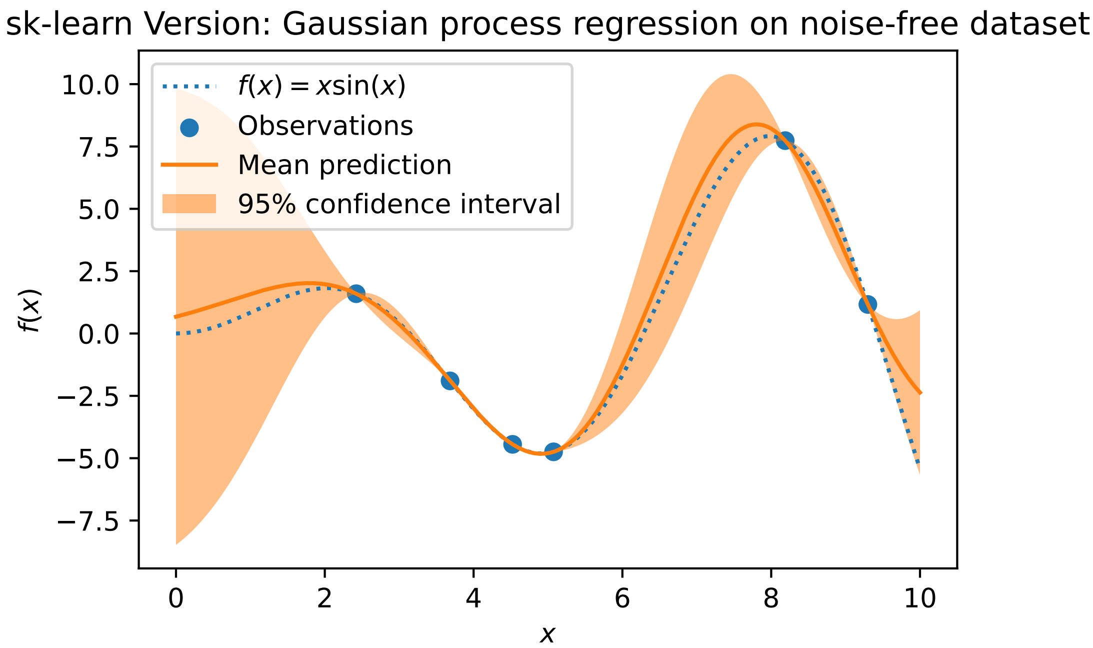
```
[1.5]
```

- Central Idea:
  - Evaluation of the surrogate model `S_lm` is much cheaper (or / and much faster) than running the real-world experiment f.

Gaussian Processes regression: basic introductory example

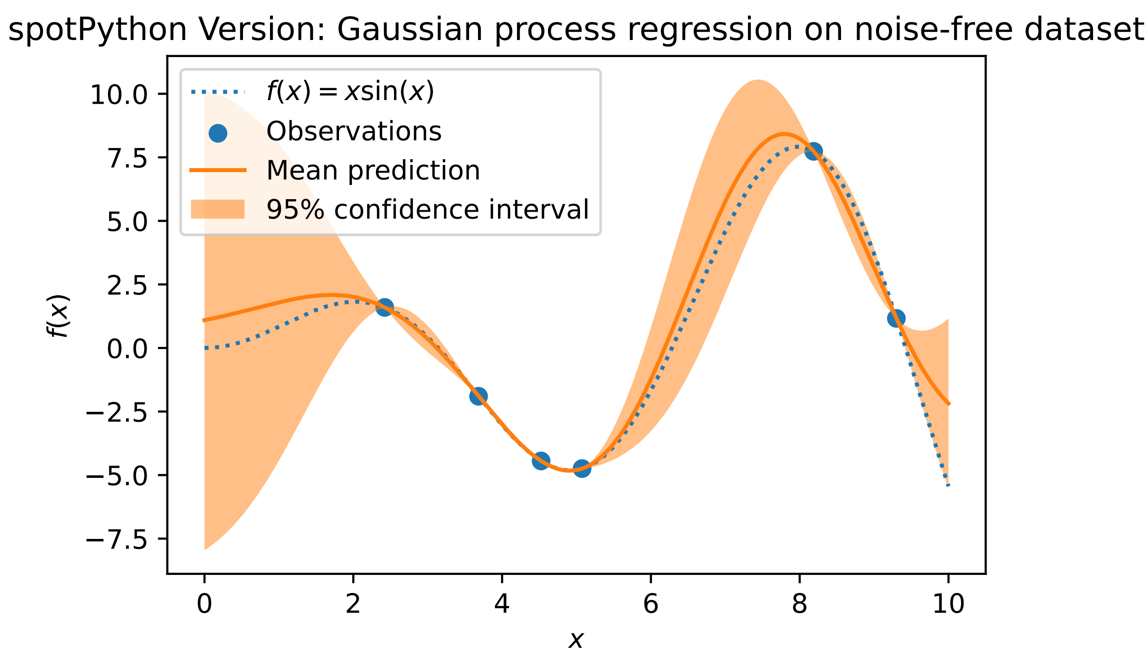
- Example from scikit-learn
- After fitting our model, we see that the hyperparameters of the kernel have been optimized. Now, we will use our kernel to compute the mean prediction of the full dataset and plot the 95% confidence interval.

```python
import numpy as np
import matplotlib.pyplot as plt
import math as m
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import RBF

X = np.linspace(start=0, stop=10, num=1_000).reshape(-1, 1)
y = np.squeeze(X * np.sin(X))
rng = np.random.RandomState(1)
training_indices = rng.choice(np.arange(y.size), size=6, replace=False)
X_train, y_train = X[training_indices], y[training_indices]

kernel = 1 * RBF(length_scale=1.0, length_scale_bounds=(1e-2, 1e2))
gaussian_process = GaussianProcessRegressor(kernel=kernel, n_restarts_optimizer=9)
gaussian_process.fit(X_train, y_train)
gaussian_process.kernel_

mean_prediction, std_prediction = gaussian_process.predict(X, return_std=True)

plt.plot(X, y, label=r"$f(x) = x \sin(x)$", linestyle="dotted")
plt.scatter(X_train, y_train, label="Observations")
plt.plot(X, mean_prediction, label="Mean prediction")
plt.fill_between(
    X.ravel(),
    mean_prediction - 1.96 * std_prediction,
    mean_prediction + 1.96 * std_prediction,
    alpha=0.5,
    label=r"95% confidence interval",
)
plt.legend()
plt.xlabel("$x$")
plt.ylabel("$f(x)$")
_ = plt.title("sk-learn Version: Gaussian process regression on noise-free dataset")
```




```
from spotPython.build.kriging import Kriging
import numpy as np
import matplotlib.pyplot as plt
rng = np.random.RandomState(1)
X = np.linspace(start=0, stop=10, num=1_000).reshape(-1, 1)
y = np.squeeze(X * np.sin(X))
training_indices = rng.choice(np.arange(y.size), size=6, replace=False)
X_train, y_train = X[training_indices], y[training_indices]


S = Kriging(name='kriging',  seed=123, log_level=50, cod_type="norm")
S.fit(X_train, y_train)

mean_prediction, std_prediction, ei = S.predict(X, return_val="all")

std_prediction

plt.plot(X, y, label=r"$f(x) = x \sin(x)$", linestyle="dotted")
plt.scatter(X_train, y_train, label="Observations")
plt.plot(X, mean_prediction, label="Mean prediction")
plt.fill_between(
```

```python
    X.ravel(),
    mean_prediction - 1.96 * std_prediction,
    mean_prediction + 1.96 * std_prediction,
    alpha=0.5,
    label=r"95% confidence interval",
)
plt.legend()
plt.xlabel("$x$")
plt.ylabel("$f(x)$")
_ = plt.title("spotPython Version: Gaussian process regression on noise-free dataset")
```



The Surrogate: Using scikit-learn models

- Default is the internal `kriging` surrogate.

```python
S_0 = Kriging(name='kriging', seed=123)
```

- Models from `scikit-learn` can be selected, e.g., Gaussian Process:

```python
# Needed for the sklearn surrogates:
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import RBF
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn import linear_model
from sklearn import tree
import pandas as pd
```

```python
kernel = 1 * RBF(length_scale=1.0, length_scale_bounds=(1e-2, 1e2))
S_GP = GaussianProcessRegressor(kernel=kernel, n_restarts_optimizer=9)
```

- and many more:

```python
S_Tree = DecisionTreeRegressor(random_state=0)
S_LM = linear_model.LinearRegression()
S_Ridge = linear_model.Ridge()
S_RF = RandomForestRegressor(max_depth=2, random_state=0)
```

- The scikit-learn GP model `S_GP` is selected.

```python
S = S_GP
```

```python
isinstance(S, GaussianProcessRegressor)
```

```
True
```

```python
from spotPython.fun.objectivefunctions import analytical
fun = analytical().fun_branin
lower = np.array([-5,-0])
upper = np.array([10,15])
design_control={"init_size": 5}
surrogate_control={
            "infill_criterion": None,
            "n_points": 1,
        }
spot_GP = spot.Spot(fun=fun, lower = lower, upper= upper, surrogate=S,
                    fun_evals = 15, noise = False, log_level = 50,
                    design_control=design_control,
                    surrogate_control=surrogate_control)
```

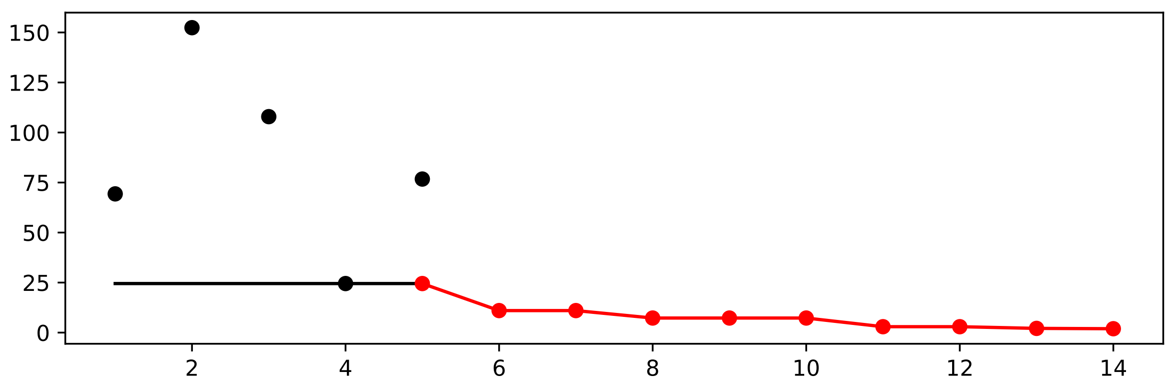
```python
spot_GP.run()
```

```
<spotPython.spot.spot.Spot at 0x2bbe67fa0>
```

```python
spot_GP.y
```

```
array([ 69.32459936, 152.38491454, 107.92560483,  24.51465459,
        76.73500031,  86.30425969,  11.00307951,  16.1174277 ,
         7.28130125,  21.82316247,  10.96088904,   2.95194767,
         3.02910042,   2.1049709 ,   1.94316396])
```

```python
spot_GP.plot_progress()
```




```python
spot_GP.print_results()
```

```
min y: 1.9431639643995577
x0: 10.0
x1: 2.9981294410720105
```

```
[['x0', 10.0], ['x1', 2.9981294410720105]]
```

Additional Examples

```python
# Needed for the sklearn surrogates:
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import RBF
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn import linear_model
from sklearn import tree
import pandas as pd
```

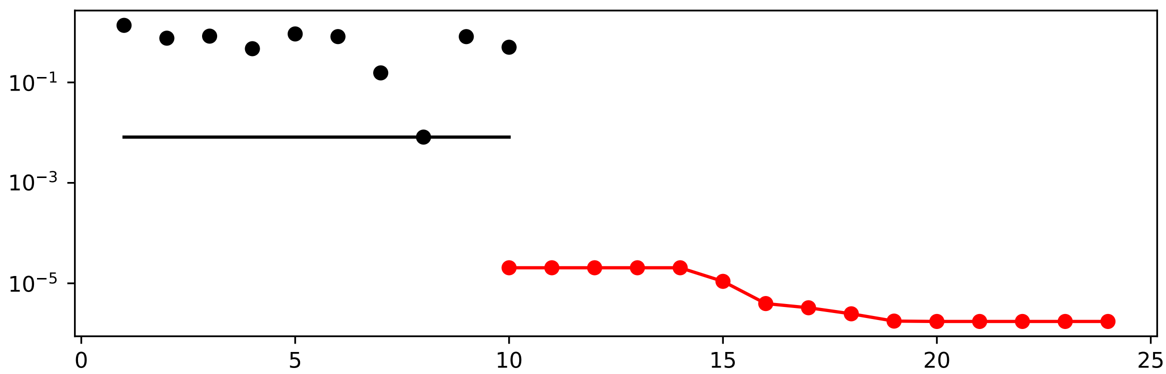
```python
kernel = 1 * RBF(length_scale=1.0, length_scale_bounds=(1e-2, 1e2))
S_GP = GaussianProcessRegressor(kernel=kernel, n_restarts_optimizer=9)
```

```python
from spotPython.build.kriging import Kriging
import numpy as np
import spotPython
from spotPython.fun.objectivefunctions import analytical
from spotPython.spot import spot

S_K = Kriging(name='kriging',
              seed=123,
              log_level=50,
              infill_criterion = "y",
              n_theta=1,
              noise=False,
              cod_type="norm")
fun = analytical().fun_sphere
lower = np.array([-1,-1])
upper = np.array([1,1])

design_control={"init_size": 10}
surrogate_control={
            "n_points": 1,
        }
spot_S_K = spot.Spot(fun=fun,
                     lower = lower,
                     upper= upper,
                     surrogate=S_K,
                     fun_evals = 25,
                     noise = False,
                     log_level = 50,
```

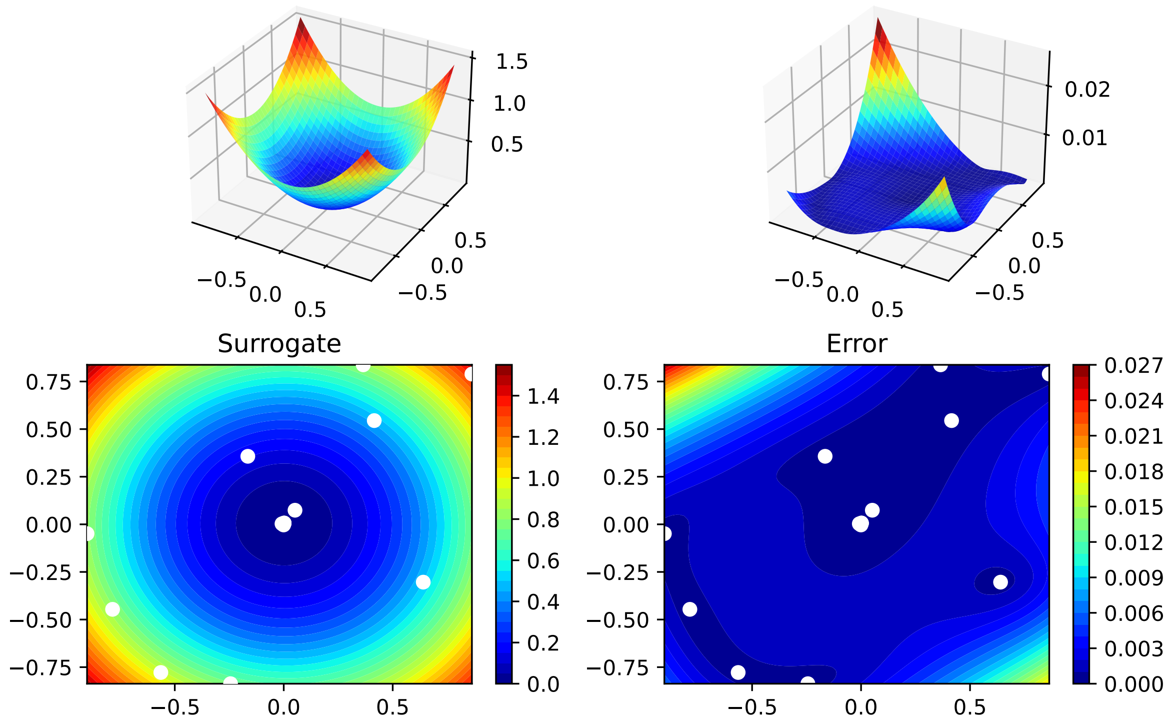
```
                              design_control=design_control,
                              surrogate_control=surrogate_control)

spot_S_K.run()
```

```
<spotPython.spot.spot.Spot at 0x2bbae1570>
```

```
spot_S_K.plot_progress(log_y=True)
```




```
spot_S_K.surrogate.plot()
```




```
spot_S_K.print_results()
```

```
min y: 1.7395335905335862e-06
x0: -0.0013044072412622557
x1: 0.00019507777780173277
```

```
[['x0', -0.0013044072412622557], ['x1', 0.00019507777780173277]]
```

4. Optimize on Surrogate

5. Evaluate on Real Objective

6. Impute / Infill new Points

Tests

```
import numpy as np
from spotPython.spot import spot
from spotPython.fun.objectivefunctions import analytical

fun_sphere = analytical().fun_sphere
spot_1 = spot.Spot(
    fun=fun_sphere,
    lower=np.array([-1, -1]),
    upper=np.array([1, 1]),
    n_points = 2
)

# (S-2) Initial Design:
spot_1.X = spot_1.design.scipy_lhd(
    spot_1.design_control["init_size"], lower=spot_1.lower, upper=spot_1.upper
)
print(spot_1.X)

# (S-3): Eval initial design:
spot_1.y = spot_1.fun(spot_1.X)
print(spot_1.y)

spot_1.surrogate.fit(spot_1.X, spot_1.y)
X0 = spot_1.suggest_new_X()
print(X0)
assert X0.size == spot_1.n_points * spot_1.k
```

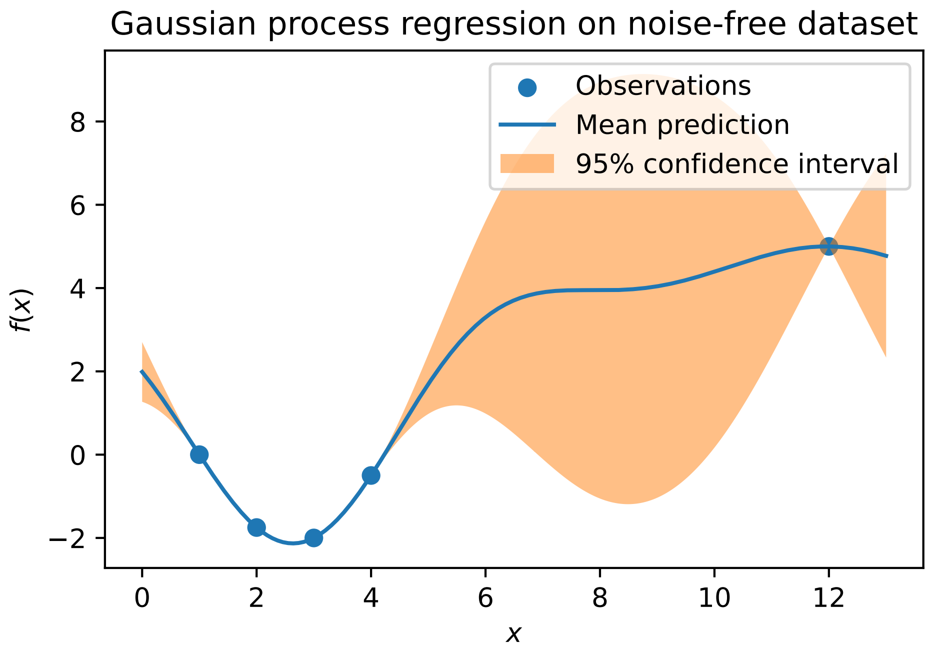
```
[[ 0.86352963  0.7892358 ]
 [-0.24407197 -0.83687436]
 [ 0.36481882  0.8375811 ]
 [ 0.415331    0.54468512]
 [-0.56395091 -0.77797854]
 [-0.90259409 -0.04899292]
 [-0.16484832  0.35724741]
```

```
 [ 0.05170659  0.07401196]
 [-0.78548145 -0.44638164]
 [ 0.64017497 -0.30363301]]
[1.36857656 0.75992983 0.83463487 0.46918172 0.92329124 0.8170764
 0.15480068 0.00815134 0.81623768 0.502017  ]
[[0.00160553 0.00428429]
 [0.00160553 0.00428429]]
```

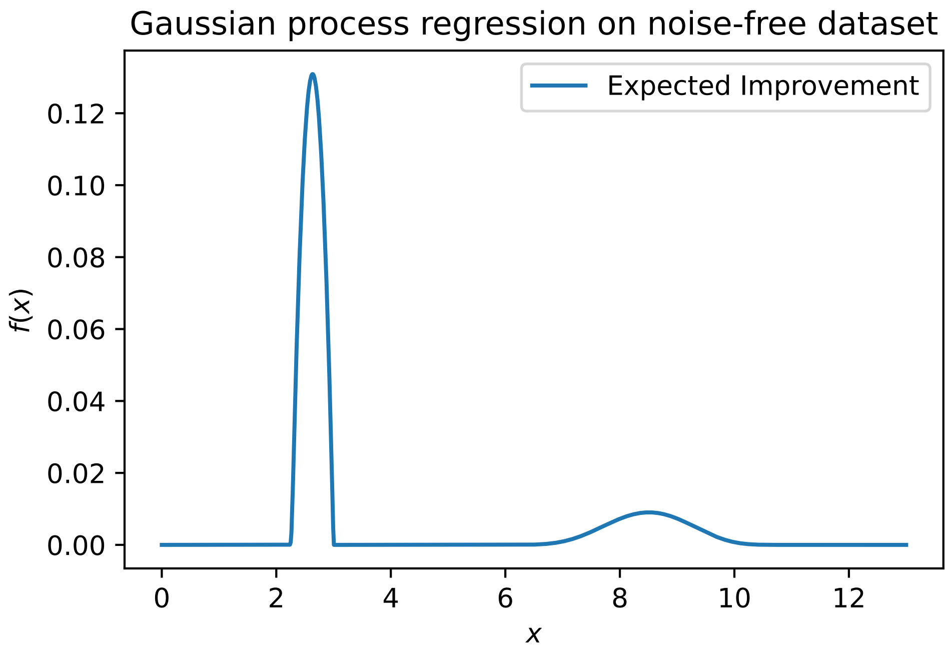
EI: The Famous Schonlau Example

```
X_train0 = np.array([1, 2, 3, 4, 12]).reshape(-1,1)
X_train = np.linspace(start=0, stop=10, num=5).reshape(-1, 1)
```

```
from spotPython.build.kriging import Kriging
import numpy as np
import matplotlib.pyplot as plt

X_train = np.array([1., 2., 3., 4., 12.]).reshape(-1,1)
y_train = np.array([0., -1.75, -2, -0.5, 5.])

S = Kriging(name='kriging',  seed=123, log_level=50, n_theta=1, noise=False, cod_type="nor
S.fit(X_train, y_train)

X = np.linspace(start=0, stop=13, num=1000).reshape(-1, 1)
mean_prediction, std_prediction, ei = S.predict(X, return_val="all")

plt.scatter(X_train, y_train, label="Observations")
plt.plot(X, mean_prediction, label="Mean prediction")
if True:
    plt.fill_between(
        X.ravel(),
        mean_prediction - 2 * std_prediction,
        mean_prediction + 2 * std_prediction,
        alpha=0.5,
        label=r"95% confidence interval",
    )
plt.legend()
plt.xlabel("$x$")
plt.ylabel("$f(x)$")
_ = plt.title("Gaussian process regression on noise-free dataset")
```




```
#plt.plot(X, y, label=r"$f(x) = x \sin(x)$", linestyle="dotted")
# plt.scatter(X_train, y_train, label="Observations")
plt.plot(X, -ei, label="Expected Improvement")
plt.legend()
plt.xlabel("$x$")
plt.ylabel("$f(x)$")
_ = plt.title("Gaussian process regression on noise-free dataset")
```



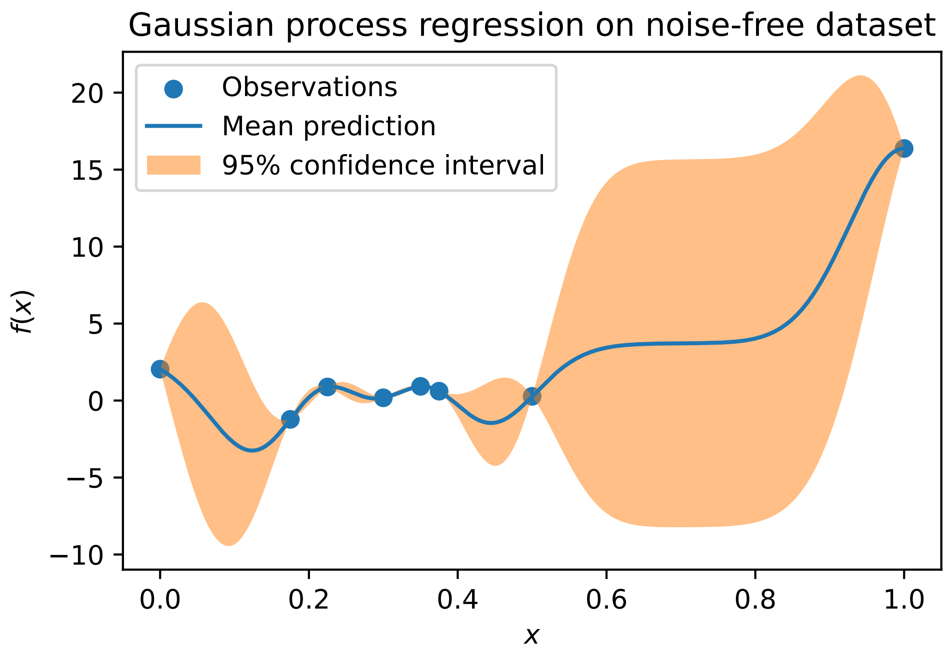

```
S.log
```

```
{'negLnLike': array([1.20788205]),
 'theta': array([1.09276]),
 'p': array([2.]),
 'Lambda': array([None], dtype=object)}
```

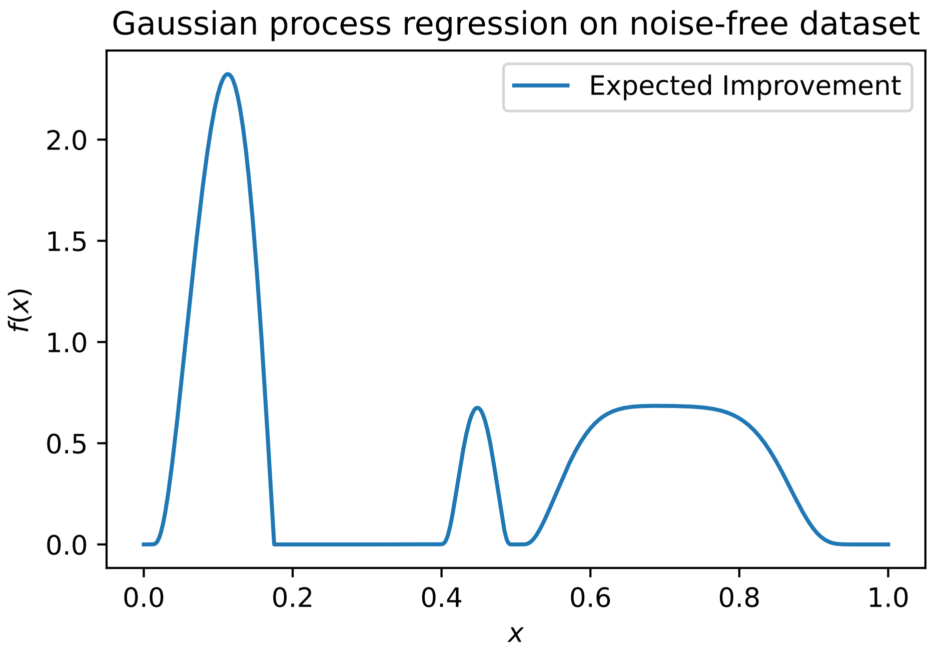
EI: The Forrester Example

```
from spotPython.build.kriging import Kriging
import numpy as np
import matplotlib.pyplot as plt
import spotPython
from spotPython.fun.objectivefunctions import analytical
from spotPython.spot import spot

# exact x locations are unknown:
X_train = np.array([0.0, 0.175, 0.225, 0.3, 0.35, 0.375, 0.5,1]).reshape(-1,1)

fun = analytical().fun_forrester
fun_control = {"sigma": 1.0,
               "seed": 123}
y_train = fun(X_train, fun_control=fun_control)

S = Kriging(name='kriging',  seed=123, log_level=50, n_theta=1, noise=False, cod_type="nor
S.fit(X_train, y_train)

X = np.linspace(start=0, stop=1, num=1000).reshape(-1, 1)
mean_prediction, std_prediction, ei = S.predict(X, return_val="all")

plt.scatter(X_train, y_train, label="Observations")
plt.plot(X, mean_prediction, label="Mean prediction")
if True:
    plt.fill_between(
        X.ravel(),
        mean_prediction - 2 * std_prediction,
        mean_prediction + 2 * std_prediction,
        alpha=0.5,
        label=r"95% confidence interval",
    )
plt.legend()
plt.xlabel("$x$")
plt.ylabel("$f(x)$")
```

```python
_ = plt.title("Gaussian process regression on noise-free dataset")
```




```python
#plt.plot(X, y, label=r"$f(x) = x \sin(x)$", linestyle="dotted")
# plt.scatter(X_train, y_train, label="Observations")
plt.plot(X, -ei, label="Expected Improvement")
plt.legend()
plt.xlabel("$x$")
plt.ylabel("$f(x)$")
_ = plt.title("Gaussian process regression on noise-free dataset")
```

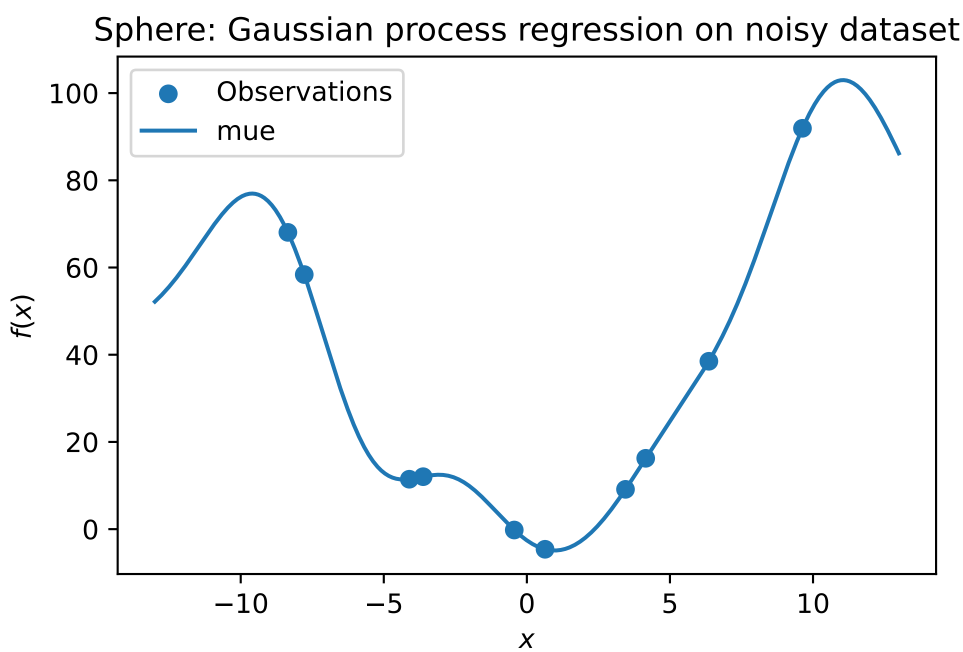

Gaussian process regression on noise-free dataset
Expected Improvement
2.0
1.5
1.0
0.5
0.0
f(x)
0.0
0.2
0.4
0.6
0.8
1.0
x

Noise

```
import numpy as np
import spotPython
from spotPython.fun.objectivefunctions import analytical
from spotPython.spot import spot
from spotPython.design.spacefilling import spacefilling
from spotPython.build.kriging import Kriging
import matplotlib.pyplot as plt

gen = spacefilling(1)
rng = np.random.RandomState(1)
lower = np.array([-10])
upper = np.array([10])
fun = analytical().fun_sphere
fun_control = {"sigma": 2,
               "seed": 125}
X = gen.scipy_lhd(10, lower=lower, upper = upper)
print(X)
y = fun(X, fun_control=fun_control)
print(y)
y.shape
X_train = X.reshape(-1,1)
y_train = y

S = Kriging(name='kriging',
            seed=123,
            log_level=50,
            n_theta=1,
            noise=False)
S.fit(X_train, y_train)

X_axis = np.linspace(start=-13, stop=13, num=1000).reshape(-1, 1)
mean_prediction, std_prediction, ei = S.predict(X_axis, return_val="all")

#plt.plot(X, y, label=r"$f(x) = x \sin(x)$", linestyle="dotted")
```

```python
plt.scatter(X_train, y_train, label="Observations")
#plt.plot(X, ei, label="Expected Improvement")
plt.plot(X_axis, mean_prediction, label="mue")
plt.legend()
plt.xlabel("$x$")
plt.ylabel("$f(x)$")
_ = plt.title("Sphere: Gaussian process regression on noisy dataset")
```

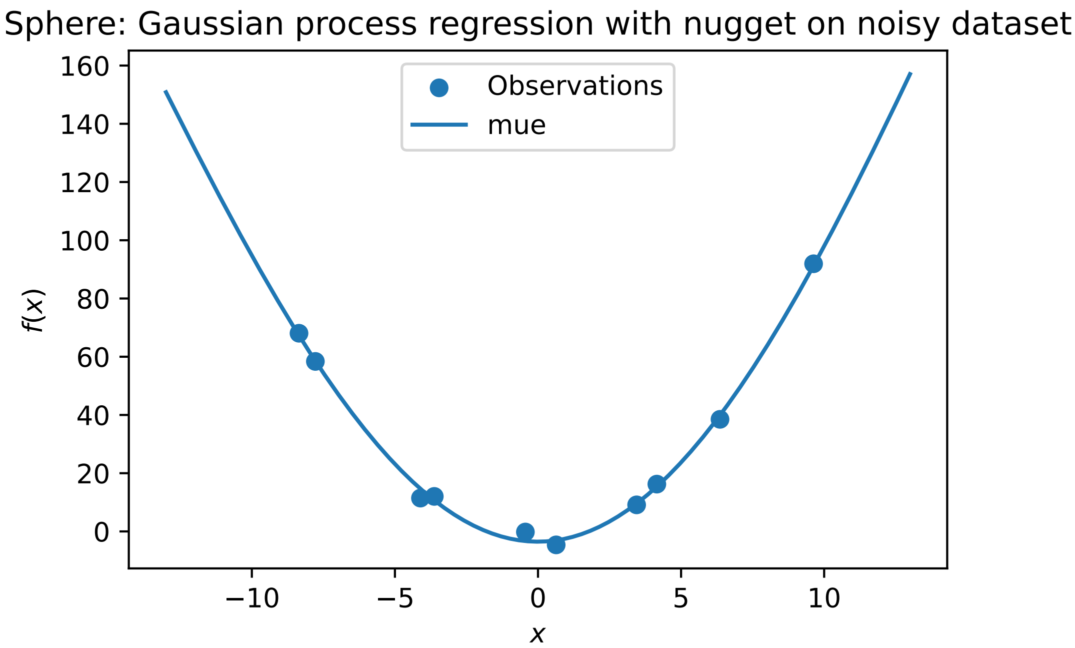
```
[[ 0.63529627]
 [-4.10764204]
 [-0.44071975]
 [ 9.63125638]
 [-8.3518118 ]
 [-3.62418901]
 [ 4.15331   ]
 [ 3.4468512 ]
 [ 6.36049088]
 [-7.77978539]]
[-4.61635371 11.44873209 -0.19988024 91.92791676 68.05926244 12.02926818
 16.2470957   9.12729929 38.4987029  58.38469104]
```



```python
S.log
```

```
{'negLnLike': array([24.69806131]),
 'theta': array([1.31023943]),
 'p': array([2.]),
 'Lambda': array([None], dtype=object)}
```

```python
S = Kriging(name='kriging',
            seed=123,
            log_level=50,
            n_theta=1,
            noise=True)
S.fit(X_train, y_train)

X_axis = np.linspace(start=-13, stop=13, num=1000).reshape(-1, 1)
mean_prediction, std_prediction, ei = S.predict(X_axis, return_val="all")

#plt.plot(X, y, label=r"$f(x) = x \sin(x)$", linestyle="dotted")
plt.scatter(X_train, y_train, label="Observations")
#plt.plot(X, ei, label="Expected Improvement")
plt.plot(X_axis, mean_prediction, label="mue")
plt.legend()
plt.xlabel("$x$")
plt.ylabel("$f(x)$")
_ = plt.title("Sphere: Gaussian process regression with nugget on noisy dataset")
```




```
S.log
```

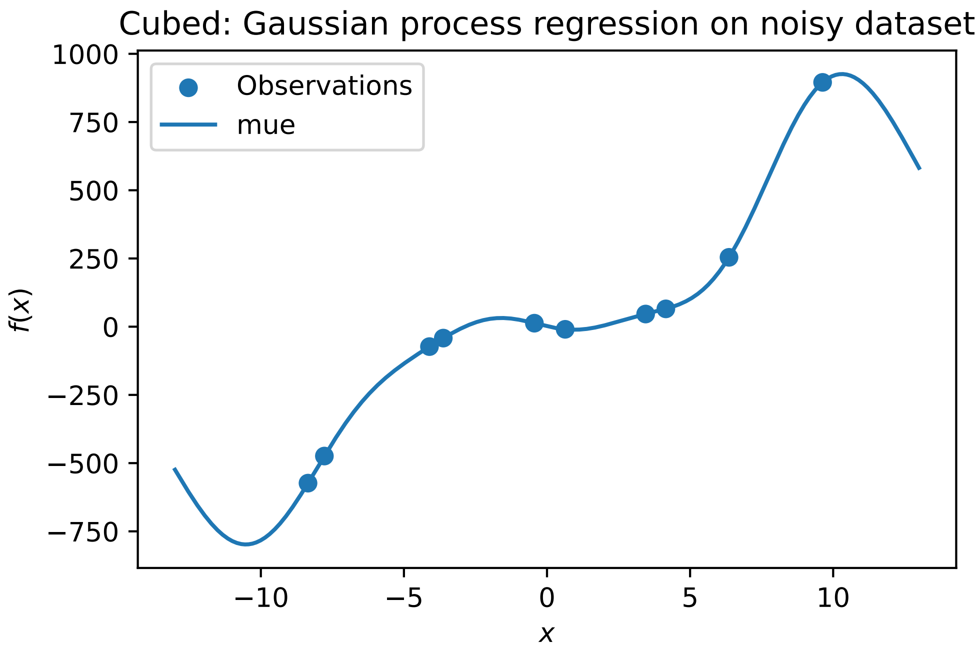
```
{'negLnLike': array([22.14095646]),
 'theta': array([-0.32527397]),
 'p': array([2.]),
 'Lambda': array([9.08815007e-05])}
```

Cubic Function

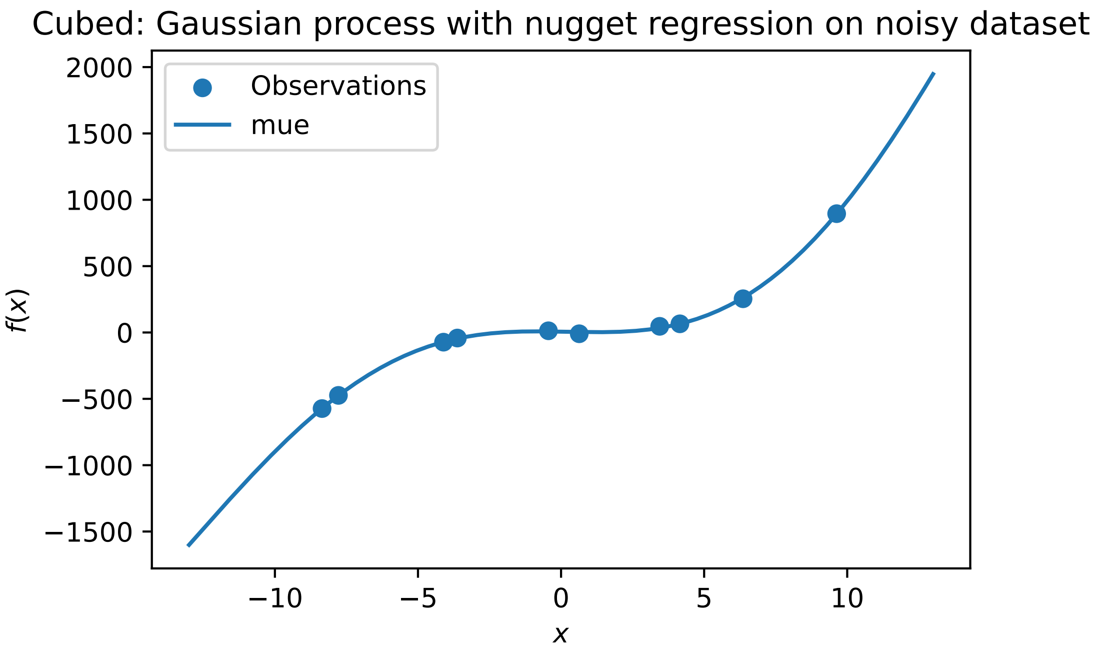
```
import numpy as np
import spotPython
from spotPython.fun.objectivefunctions import analytical
from spotPython.spot import spot
from spotPython.design.spacefilling import spacefilling
from spotPython.build.kriging import Kriging
import matplotlib.pyplot as plt

gen = spacefilling(1)
rng = np.random.RandomState(1)
```

```python
lower = np.array([-10])
upper = np.array([10])
fun = analytical().fun_cubed
fun_control = {"sigma": 10,
               "seed": 123}

X = gen.scipy_lhd(10, lower=lower, upper = upper)
print(X)
y = fun(X, fun_control=fun_control)
print(y)
y.shape
X_train = X.reshape(-1,1)
y_train = y

S = Kriging(name='kriging',  seed=123, log_level=50, n_theta=1, noise=False)
S.fit(X_train, y_train)

X_axis = np.linspace(start=-13, stop=13, num=1000).reshape(-1, 1)
mean_prediction, std_prediction, ei = S.predict(X_axis, return_val="all")

plt.scatter(X_train, y_train, label="Observations")
#plt.plot(X, ei, label="Expected Improvement")
plt.plot(X_axis, mean_prediction, label="mue")
plt.legend()
plt.xlabel("$x$")
plt.ylabel("$f(x)$")
_ = plt.title("Cubed: Gaussian process regression on noisy dataset")
```

```
[[ 0.63529627]
 [-4.10764204]
 [-0.44071975]
 [ 9.63125638]
 [-8.3518118 ]
 [-3.62418901]
 [ 4.15331   ]
 [ 3.4468512 ]
 [ 6.36049088]
 [-7.77978539]]
[  -9.63480707  -72.98497325   12.7936499   895.34567477 -573.35961837
  -41.83176425   65.27989461   46.37081417  254.1530734  -474.09587355]
```



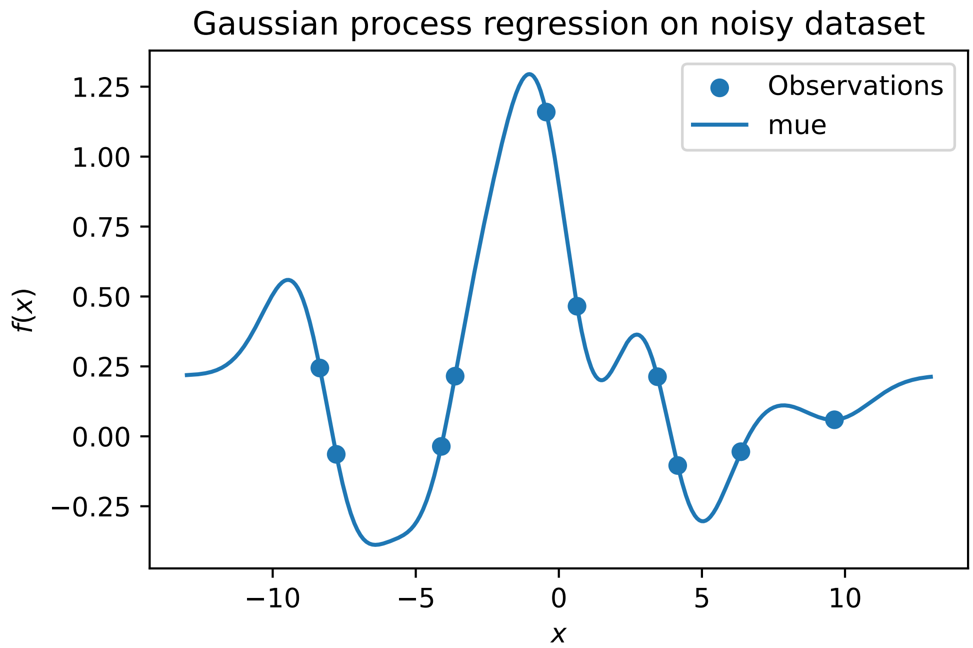

```
S = Kriging(name='kriging',  seed=123, log_level=0, n_theta=1, noise=True)
S.fit(X_train, y_train)

X_axis = np.linspace(start=-13, stop=13, num=1000).reshape(-1, 1)
mean_prediction, std_prediction, ei = S.predict(X_axis, return_val="all")

plt.scatter(X_train, y_train, label="Observations")
#plt.plot(X, ei, label="Expected Improvement")
plt.plot(X_axis, mean_prediction, label="mue")
plt.legend()
plt.xlabel("$x$")
plt.ylabel("$f(x)$")
_ = plt.title("Cubed: Gaussian process with nugget regression on noisy dataset")
```



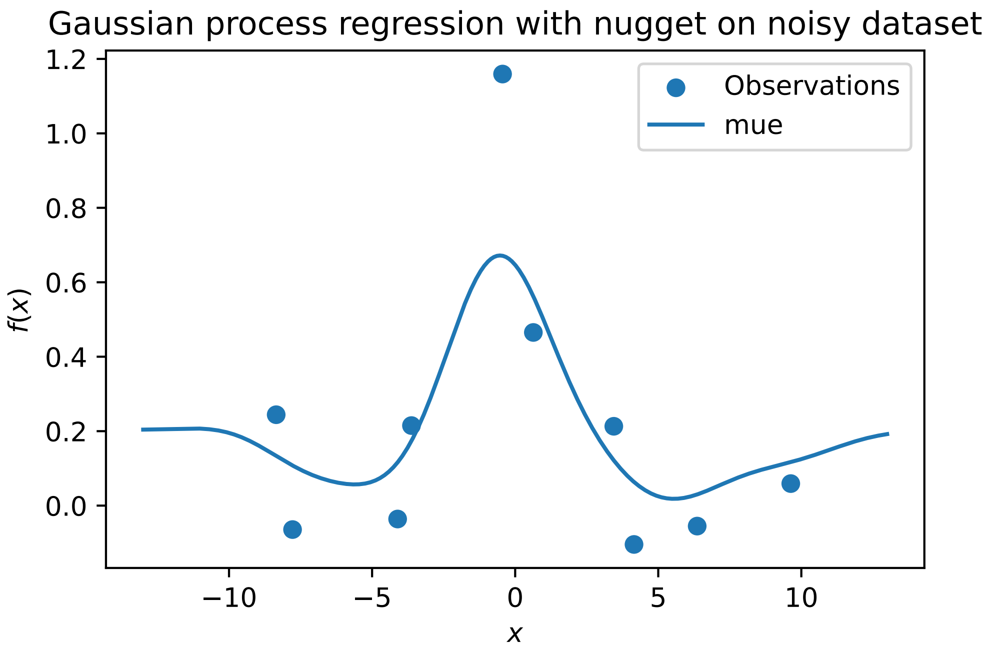

```
import numpy as np
import spotPython
from spotPython.fun.objectivefunctions import analytical
from spotPython.spot import spot
from spotPython.design.spacefilling import spacefilling
from spotPython.build.kriging import Kriging
import matplotlib.pyplot as plt

gen = spacefilling(1)
rng = np.random.RandomState(1)
lower = np.array([-10])
upper = np.array([10])
fun = analytical().fun_runge
fun_control = {"sigma": 0.25,
               "seed": 123}

X = gen.scipy_lhd(10, lower=lower, upper = upper)
print(X)
y = fun(X, fun_control=fun_control)
print(y)
y.shape
```

```python
X_train = X.reshape(-1,1)
y_train = y

S = Kriging(name='kriging',  seed=123, log_level=50, n_theta=1, noise=False)
S.fit(X_train, y_train)

X_axis = np.linspace(start=-13, stop=13, num=1000).reshape(-1, 1)
mean_prediction, std_prediction, ei = S.predict(X_axis, return_val="all")

plt.scatter(X_train, y_train, label="Observations")
#plt.plot(X, ei, label="Expected Improvement")
plt.plot(X_axis, mean_prediction, label="mue")
plt.legend()
plt.xlabel("$x$")
plt.ylabel("$f(x)$")
_ = plt.title("Gaussian process regression on noisy dataset")
```

```
[[ 0.63529627]
 [-4.10764204]
 [-0.44071975]
 [ 9.63125638]
 [-8.3518118 ]
 [-3.62418901]
 [ 4.15331   ]
 [ 3.4468512 ]
 [ 6.36049088]
 [-7.77978539]]
[ 0.46517267 -0.03599548  1.15933822  0.05915901  0.24419145  0.21502359
 -0.10432134  0.21312309 -0.05502681 -0.06434374]
```




```
S = Kriging(name='kriging',
            seed=123,
            log_level=50,
            n_theta=1,
            noise=True)
S.fit(X_train, y_train)

X_axis = np.linspace(start=-13, stop=13, num=1000).reshape(-1, 1)
mean_prediction, std_prediction, ei = S.predict(X_axis, return_val="all")

plt.scatter(X_train, y_train, label="Observations")
#plt.plot(X, ei, label="Expected Improvement")
plt.plot(X_axis, mean_prediction, label="mue")
plt.legend()
plt.xlabel("$x$")
plt.ylabel("$f(x)$")
_ = plt.title("Gaussian process regression with nugget on noisy dataset")
```


Gaussian process regression with nugget on noisy dataset
Observations
mue
1.2
1.0
0.8
0.6
0.4
0.2
0.0
f(x)
−10
−5
0
5
10
x

Factors

```
["num"] * 3
```

```
['num', 'num', 'num']
```

```
from spotPython.design.spacefilling import spacefilling
from spotPython.build.kriging import Kriging
from spotPython.fun.objectivefunctions import analytical
import numpy as np
```

```
gen = spacefilling(2)
n = 30
rng = np.random.RandomState(1)
lower = np.array([-5,-0])
upper = np.array([10,15])
fun = analytical().fun_branin_factor
#fun = analytical(sigma=0).fun_sphere

X0 = gen.scipy_lhd(n, lower=lower, upper = upper)
X1 = np.random.randint(low=1, high=3, size=(n,))
X = np.c_[X0, X1]
y = fun(X)
S = Kriging(name='kriging',  seed=123, log_level=50, n_theta=3, noise=False, var_type=["nu
S.fit(X, y)
Sf = Kriging(name='kriging',  seed=123, log_level=50, n_theta=3, noise=False, var_type=["n
Sf.fit(X, y)
n = 50
X0 = gen.scipy_lhd(n, lower=lower, upper = upper)
X1 = np.random.randint(low=1, high=3, size=(n,))
X = np.c_[X0, X1]
y = fun(X)
s=np.sum(np.abs(S.predict(X)[0] - y))
sf=np.sum(np.abs(Sf.predict(X)[0] - y))
sf - s
```

```
-20.955318601116687
```

```python
vars(S)
```

```
{'name': 'kriging',
 'seed': 123,
 'rng': Generator(PCG64) at 0x2BD9EB220,
 'log': {'negLnLike': array([77.6885701]),
  'theta': array([ 0.77012209, -0.62247126, -3.        ]),
  'p': array([2.]),
  'Lambda': array([None], dtype=object)},
 'verbosity': 50,
 'noise': False,
 'var_type': ['num', 'num', 'num'],
 'cod_type': 'norm',
 'use_cod_y': False,
 'log_level': 50,
 'sigma': 0,
 'eps': 1.4901161193847656e-08,
 'min_theta': -3,
 'max_theta': 2,
 'min_p': 1,
 'max_p': 2,
 'min_Lambda': 1e-09,
 'max_Lambda': 1.0,
 'n_theta': 3,
 'n_p': 1,
 'optim_p': False,
 'cnd_Psi': 116616201.37817818,
 'inf_Psi': False,
 'model_optimizer': <function scipy.optimize._differentialevolution.differential_evolution(fu
 'model_fun_evals': 100,
 'nat_X': array([[-2.84117593,  5.97308949,  1.        ],
        [-3.61017994,  6.90781409,  2.        ],
        [ 9.91204705,  5.09395275,  2.        ],
        [-4.4616725 ,  1.3617128 ,  1.        ],
        [-2.40987728,  8.05505365,  1.        ],
        [ 5.24351477, 10.3775177 ,  1.        ],
        [ 2.5878792 , 11.89311852,  1.        ],
        [ 8.12926647,  3.6850299 ,  2.        ],
        [-1.96370363,  0.38404591,  2.        ],
        [ 0.10043744,  1.74091748,  1.        ],
```

```
       [ 1.38422219,  7.917048  ,  2.        ],
       [ 9.25110552, 13.20863768,  1.        ],
       [-1.09216899,  4.49255254,  1.        ],
       [ 3.26443339, 14.63587834,  1.        ],
       [ 4.54069975,  9.187233  ,  2.        ],
       [ 7.04143871,  8.56765487,  1.        ],
       [-0.10907144, 14.06693628,  1.        ],
       [ 1.63462403,  7.36106735,  1.        ],
       [ 4.10147822, 12.56738914,  2.        ],
       [ 0.85028105,  2.73647896,  2.        ],
       [ 5.9642566 , 10.70838079,  1.        ],
       [ 6.3810468 ,  4.61751818,  1.        ],
       [ 2.41318418, 12.34362887,  1.        ],
       [-3.00723724,  0.98372404,  1.        ],
       [ 8.75164908,  6.26584373,  1.        ],
       [ 6.93615484, 11.37121875,  2.        ],
       [-0.50159055,  2.30946613,  2.        ],
       [-4.78793654,  9.78635061,  1.        ],
       [ 7.58244883, 13.69175437,  1.        ],
       [ 3.86695804,  3.09448894,  2.        ]]),
'nat_y': array([ 42.09388125,  33.965223  ,  -3.74411425, 222.41884106,
        19.25981051, 108.97027586,  95.49463536,   1.74135821,
        81.69079805,  46.3681054 ,  16.77800101, 128.83487789,
        35.98089141, 165.59047908,  58.42915074,  81.277265  ,
        91.82566843,  32.4736378 , 113.78079756,  10.34983517,
       121.38146188,  41.89850532, 101.79776372, 130.84111259,
        30.98328188, 111.57203528,  28.861847  ,  66.90159224,
       174.66155808,  -5.44983081]),
'n': 30,
'k': 3,
'cod_X': array([[0.13243284, 0.39216316, 0.        ],
       [0.08011959, 0.45774943, 1.        ],
       [1.        , 0.33047728, 1.        ],
       [0.02219486, 0.06859938, 0.        ],
       [0.16177292, 0.53824712, 0.        ],
       [0.68241242, 0.7012061 , 0.        ],
       [0.50175673, 0.80755037, 0.        ],
       [0.87872227, 0.23161822, 1.        ],
       [0.1921249 , 0.        , 1.        ],
       [0.33254282, 0.09520682, 0.        ],
       [0.41987521, 0.52856376, 1.        ],
       [0.95503794, 0.89985564, 0.        ],
       [0.25141304, 0.28827918, 0.        ],
```

```
       [0.54778088, 1.        , 0.        ],
       [0.63460182, 0.61768809, 1.        ],
       [0.8047203 , 0.57421451, 0.        ],
       [0.3182905 , 0.96007938, 0.        ],
       [0.43690937, 0.48955259, 0.        ],
       [0.60472277, 0.85486153, 1.        ],
       [0.38355265, 0.1650618 , 1.        ],
       [0.73144253, 0.72442157, 0.        ],
       [0.75979563, 0.29704758, 0.        ],
       [0.4898727 , 0.83916107, 0.        ],
       [0.12113614, 0.04207727, 0.        ],
       [0.92106127, 0.41270467, 0.        ],
       [0.79755813, 0.77093054, 1.        ],
       [0.29158849, 0.13509984, 1.        ],
       [0.        , 0.65972602, 0.        ],
       [0.84152375, 0.93375421, 0.        ],
       [0.588769  , 0.19018207, 1.        ]]),
'cod_y': array([ 42.09388125,  33.965223  ,  -3.74411425, 222.41884106,
        19.25981051, 108.97027586,  95.49463536,   1.74135821,
        81.69079805,  46.3681054 ,  16.77800101, 128.83487789,
        35.98089141, 165.59047908,  58.42915074,  81.277265  ,
        91.82566843,  32.4736378 , 113.78079756,  10.34983517,
       121.38146188,  41.89850532, 101.79776372, 130.84111259,
        30.98328188, 111.57203528,  28.861847  ,  66.90159224,
       174.66155808,  -5.44983081]),
'num_mask': array([ True,  True,  True]),
'factor_mask': array([False, False, False]),
'int_mask': array([False, False, False]),
'ordered_mask': array([ True,  True,  True]),
'nat_range_X': [[-4.787936542018484, 9.912047049457485],
 [0.3840459056967906, 14.63587833590837],
 [1.0, 2.0]],
'nat_range_y': [-5.449830806521344, 222.4188410638947],
'nat_mean_X': array([2.50508374, 7.5128501 , 1.36666667]),
'nat_std_X': array([4.3289971 , 4.28784459, 0.48189441]),
'nat_mean_y': 72.90095819227218,
'nat_std_y': 56.137717999904446,
'mean_cod_y': array([ 42.09388125,  33.965223  ,  -3.74411425, 222.41884106,
        19.25981051, 108.97027586,  95.49463536,   1.74135821,
        81.69079805,  46.3681054 ,  16.77800101, 128.83487789,
        35.98089141, 165.59047908,  58.42915074,  81.277265  ,
        91.82566843,  32.4736378 , 113.78079756,  10.34983517,
       121.38146188,  41.89850532, 101.79776372, 130.84111259,
```

```
       30.98328188, 111.57203528,  28.861847  ,  66.90159224,
      174.66155808,  -5.44983081]),
'theta': array([ 0.77012209, -0.62247126, -3.        ]),
'x0_theta': array([0.1, 0.1, 0.1]),
'p': array([2.]),
'pen_val': 10241.66847530361,
'negLnLike': 77.68857009522574,
'gen': <spotPython.design.spacefilling.spacefilling at 0x2bda737c0>,
'LnDetPsi': -186.2872329870982,
'Psi': array([[1.00000001, 0.98201834, 0.01185246, 0.90796399, 0.98989083,
        0.16457297, 0.42974402, 0.03734253, 0.94301789, 0.77344827,
        0.61134398, 0.01747154, 0.91763466, 0.3314597 , 0.22347453,
        0.06924857, 0.75548743, 0.57792629, 0.25515195, 0.68062907,
        0.11768178, 0.09823385, 0.449241  , 0.97046012, 0.02564635,
        0.07129372, 0.84707926, 0.88658444, 0.04824031, 0.29016677],
       [0.98201834, 1.00000001, 0.00681969, 0.94470352, 0.95904584,
        0.11627379, 0.34050895, 0.02308207, 0.88349384, 0.66520996,
        0.50605378, 0.01049999, 0.83470724, 0.25683051, 0.16251036,
        0.04519333, 0.67347438, 0.47187375, 0.19040195, 0.56964405,
        0.08072827, 0.06534044, 0.35892873, 0.94921315, 0.01550091,
        0.04711652, 0.74958885, 0.95261945, 0.03112497, 0.21416985],
       [0.01185246, 0.00681969, 1.00000001, 0.00352142, 0.01576725,
        0.5337284 , 0.21926346, 0.91487826, 0.02085231, 0.07148745,
        0.13647158, 0.91371713, 0.03680473, 0.26915805, 0.44659807,
        0.78679795, 0.05884551, 0.15341424, 0.3731105 , 0.10594822,
        0.62950054, 0.71097792, 0.20280758, 0.01035427, 0.96144704,
        0.75001278, 0.05155896, 0.0026934 , 0.78998881, 0.36759489],
       [0.90796399, 0.94470352, 0.00352142, 1.00000001, 0.8458937 ,
        0.06974773, 0.2265373 , 0.01318677, 0.8418057 , 0.56695341,
        0.37419449, 0.00503999, 0.72543625, 0.15977277, 0.10208518,
        0.0255338 , 0.49363339, 0.34809062, 0.11681159, 0.46192883,
        0.04663074, 0.04007584, 0.23932752, 0.94381216, 0.00833571,
        0.02573999, 0.65082398, 0.91736624, 0.01604192, 0.1502759 ],
       [0.98989083, 0.95904584, 0.01576725, 0.8458937 , 1.00000001,
        0.20130486, 0.49751555, 0.04731017, 0.92725335, 0.80365337,
        0.67475783, 0.02381013, 0.93966414, 0.39514873, 0.2673126 ,
        0.08758349, 0.82966171, 0.63989835, 0.30710254, 0.72330097,
        0.14664501, 0.11998877, 0.51909764, 0.9338429 , 0.03338969,
        0.09118875, 0.87020472, 0.85413465, 0.0633618 , 0.3316047 ],
       [0.16457297, 0.11627379, 0.5337284 , 0.06974773, 0.20130486,
        1.00000001, 0.82289282, 0.75534025, 0.21563886, 0.44548467,
        0.66094248, 0.63942134, 0.32148116, 0.87980368, 0.98400171,
        0.91214365, 0.4507145 , 0.69371694, 0.95869507, 0.55120434,
```

```
 0.98581354, 0.92845625, 0.80019947, 0.14097283, 0.70095402,
 0.9228823 , 0.37639616, 0.06435452, 0.85042846, 0.89141985],
[0.42974402, 0.34050895, 0.21926346, 0.2265373 , 0.49751555,
 0.82289282, 1.00000001, 0.39966956, 0.48614739, 0.74849769,
 0.94265432, 0.29753206, 0.64826262, 0.97891483, 0.89266363,
 0.57486506, 0.81561821, 0.95228694, 0.93802312, 0.83380349,
 0.73170093, 0.63486155, 0.99893037, 0.37043715, 0.34206293,
 0.59648935, 0.69140694, 0.2258011 , 0.50471525, 0.87240119],
[0.03734253, 0.02308207, 0.91487826, 0.01318677, 0.04731017,
 0.75534025, 0.39966956, 1.00000001, 0.0614532 , 0.17161124,
 0.28333204, 0.86778234, 0.09831129, 0.45524392, 0.67938278,
 0.94058643, 0.13840716, 0.31142183, 0.58575559, 0.23568344,
 0.82970071, 0.91821115, 0.375444  , 0.03370499, 0.98080637,
 0.89746809, 0.13098265, 0.01012481, 0.88096224, 0.6092012 ],
[0.94301789, 0.88349384, 0.02085231, 0.8418057 , 0.92725335,
 0.21563886, 0.48614739, 0.0614532 , 1.00000001, 0.88754317,
 0.68924498, 0.02671878, 0.95932389, 0.37360102, 0.2881708 ,
 0.10126247, 0.73006958, 0.66292353, 0.30819555, 0.80064423,
 0.15891496, 0.14658804, 0.50100259, 0.96937392, 0.04194834,
 0.10018152, 0.93929639, 0.72453228, 0.06768236, 0.39247008],
[0.77344827, 0.66520996, 0.07148745, 0.56695341, 0.80365337,
 0.44548467, 0.74849769, 0.17161124, 0.88754317, 1.00000001,
 0.91327571, 0.08743626, 0.95345771, 0.6261664 , 0.54688143,
 0.25463252, 0.83559427, 0.90370766, 0.5627079 , 0.98266211,
 0.35641096, 0.33792746, 0.75744071, 0.76803659, 0.12693235,
 0.2506798 , 0.98880435, 0.48317956, 0.18385345, 0.67716046],
[0.61134398, 0.50605378, 0.13647158, 0.37419449, 0.67475783,
 0.66094248, 0.94265432, 0.28333204, 0.68924498, 0.91327571,
 1.00000001, 0.17892304, 0.83366022, 0.86038687, 0.76073354,
 0.4173402 , 0.89924772, 0.99693263, 0.7971983 , 0.96147398,
 0.5588212 , 0.49939361, 0.94850352, 0.55815851, 0.22679265,
 0.42562366, 0.87471021, 0.35222033, 0.33711029, 0.82256577],
[0.01747154, 0.01049999, 0.91371713, 0.00503999, 0.02381013,
 0.63942134, 0.29753206, 0.86778234, 0.02671878, 0.08743626,
 0.17892304, 1.00000001, 0.04952254, 0.37556907, 0.53537846,
 0.85352338, 0.09172254, 0.19762329, 0.48465439, 0.12829017,
 0.73947438, 0.73256638, 0.27932784, 0.01396187, 0.93856352,
 0.85981478, 0.06501655, 0.00457984, 0.92665801, 0.40200061],
[0.91763466, 0.83470724, 0.03680473, 0.72543625, 0.93966414,
 0.32148116, 0.64826262, 0.09831129, 0.95932389, 0.95345771,
 0.83366022, 0.04952254, 1.00000001, 0.52825703, 0.40993947,
 0.16158132, 0.87459641, 0.80869431, 0.44360695, 0.89810568,
 0.24595278, 0.21820298, 0.66543594, 0.89187868, 0.07100681,
```

```
 0.16309403, 0.98402531, 0.66683439, 0.11642732, 0.5098488 ],
[0.3314597 , 0.25683051, 0.26915805, 0.15977277, 0.39514873,
 0.87980368, 0.97891483, 0.45524392, 0.37360102, 0.6261664 ,
 0.86038687, 0.37556907, 0.52825703, 1.00000001, 0.92287475,
 0.64914849, 0.73301674, 0.87410706, 0.97519052, 0.72170402,
 0.80509384, 0.68206863, 0.97441114, 0.27499013, 0.40535777,
 0.68318406, 0.5677804 , 0.16612386, 0.60093252, 0.84592925],
[0.22347453, 0.16251036, 0.44659807, 0.10208518, 0.2673126 ,
 0.98400171, 0.89266363, 0.67938278, 0.2881708 , 0.54688143,
 0.76073354, 0.53537846, 0.40993947, 0.92287475, 1.00000001,
 0.84205342, 0.53886894, 0.79048026, 0.98149772, 0.65698554,
 0.94274894, 0.8888375 , 0.87277587, 0.19535553, 0.6099627 ,
 0.85043163, 0.4730439 , 0.09315632, 0.75803776, 0.94557162],
[0.06924857, 0.04519333, 0.78679795, 0.0255338 , 0.08758349,
 0.91214365, 0.57486506, 0.94058643, 0.10126247, 0.25463252,
 0.4173402 , 0.85352338, 0.16158132, 0.64914849, 0.84205342,
 1.00000001, 0.23950237, 0.44998093, 0.77461857, 0.33765605,
 0.96366726, 0.97024063, 0.54847012, 0.05961145, 0.9176441 ,
 0.98952299, 0.20232527, 0.02201526, 0.96193195, 0.7328152 ],
[0.75548743, 0.67347438, 0.05884551, 0.49363339, 0.82966171,
 0.4507145 , 0.81561821, 0.13840716, 0.73006958, 0.83559427,
 0.89924772, 0.09172254, 0.87459641, 0.73301674, 0.53886894,
 0.23950237, 1.00000001, 0.87311859, 0.61453826, 0.83790931,
 0.36108093, 0.28564836, 0.8378682 , 0.65053758, 0.10953312,
 0.25602687, 0.84574641, 0.5388942 , 0.19934591, 0.56366618],
[0.57792629, 0.47187375, 0.15341424, 0.34809062, 0.63989835,
 0.69371694, 0.95228694, 0.31142183, 0.66292353, 0.90370766,
 0.99693263, 0.19762329, 0.80869431, 0.87410706, 0.79048026,
 0.44998093, 0.87311859, 1.00000001, 0.81979268, 0.95802275,
 0.59207503, 0.53637919, 0.95535143, 0.52989444, 0.25105976,
 0.45566638, 0.85611116, 0.3226246 , 0.36372852, 0.8536701 ],
[0.25515195, 0.19040195, 0.3731105 , 0.11681159, 0.30710254,
 0.95869507, 0.93802312, 0.58575559, 0.30819555, 0.5627079 ,
 0.7971983 , 0.48465439, 0.44360695, 0.97519052, 0.98149772,
 0.77461857, 0.61453826, 0.81979268, 1.00000001, 0.66923798,
 0.90516245, 0.80504247, 0.92426886, 0.21523913, 0.52884836,
 0.80195232, 0.49603454, 0.11486152, 0.71693524, 0.89863533],
[0.68062907, 0.56964405, 0.10594822, 0.46192883, 0.72330097,
 0.55120434, 0.83380349, 0.23568344, 0.80064423, 0.98266211,
 0.96147398, 0.12829017, 0.89810568, 0.72170402, 0.65698554,
 0.33765605, 0.83790931, 0.95802275, 0.66923798, 1.00000001,
 0.45452972, 0.43216431, 0.83864469, 0.66350844, 0.17953826,
 0.33383081, 0.95120181, 0.39618595, 0.25225537, 0.78020046],
```

```
[0.11768178, 0.08072827, 0.62950054, 0.04663074, 0.14664501,
 0.98581354, 0.73170093, 0.82970071, 0.15891496, 0.35641096,
 0.5588212 , 0.73947438, 0.24595278, 0.80509384, 0.94274894,
 0.96366726, 0.36108093, 0.59207503, 0.90516245, 0.45452972,
 1.00000001, 0.95284716, 0.70690268, 0.0997628 , 0.79060529,
 0.97310514, 0.29422686, 0.042756  , 0.9214309 , 0.82780785],
[0.09823385, 0.06534044, 0.71097792, 0.04007584, 0.11998877,
 0.92845625, 0.63486155, 0.91821115, 0.14658804, 0.33792746,
 0.49939361, 0.73256638, 0.21820298, 0.68206863, 0.8888375 ,
 0.97024063, 0.28564836, 0.53637919, 0.80504247, 0.43216431,
 0.95284716, 1.00000001, 0.60699107, 0.08910034, 0.85524105,
 0.93897817, 0.27294921, 0.03233316, 0.87280898, 0.83860974],
[0.449241  , 0.35892873, 0.20280758, 0.23932752, 0.51909764,
 0.80019947, 0.99893037, 0.375444  , 0.50100259, 0.75744071,
 0.94850352, 0.27932784, 0.66543594, 0.97441114, 0.87277587,
 0.54847012, 0.8378682 , 0.95535143, 0.92426886, 0.83864469,
 0.70690268, 0.60699107, 1.00000001, 0.38581419, 0.32030363,
 0.57136484, 0.70411432, 0.24143473, 0.48167099, 0.85293906],
[0.97046012, 0.94921315, 0.01035427, 0.94381216, 0.9338429 ,
 0.14097283, 0.37043715, 0.03370499, 0.96937392, 0.76803659,
 0.55815851, 0.01396187, 0.89187868, 0.27499013, 0.19535553,
 0.05961145, 0.65053758, 0.52989444, 0.21523913, 0.66350844,
 0.0997628 , 0.08910034, 0.38581419, 1.00000001, 0.02233237,
 0.05944512, 0.84013328, 0.83742395, 0.03891567, 0.27409508],
[0.02564635, 0.01550091, 0.96144704, 0.00833571, 0.03338969,
 0.70095402, 0.34206293, 0.98080637, 0.04194834, 0.12693235,
 0.22679265, 0.93856352, 0.07100681, 0.40535777, 0.6099627 ,
 0.9176441 , 0.10953312, 0.25105976, 0.52884836, 0.17953826,
 0.79060529, 0.85524105, 0.32030363, 0.02233237, 1.00000001,
 0.88563509, 0.0950599 , 0.00666132, 0.90301239, 0.51520836],
[0.07129372, 0.04711652, 0.75001278, 0.02573999, 0.09118875,
 0.9228823 , 0.59648935, 0.89746809, 0.10018152, 0.2506798 ,
 0.42562366, 0.85981478, 0.16309403, 0.68318406, 0.85043163,
 0.98952299, 0.25602687, 0.45566638, 0.80195232, 0.33383081,
 0.97310514, 0.93897817, 0.57136484, 0.05944512, 0.88563509,
 1.00000001, 0.20102529, 0.02350291, 0.98146493, 0.7137573 ],
[0.84707926, 0.74958885, 0.05155896, 0.65082398, 0.87020472,
 0.37639616, 0.69140694, 0.13098265, 0.93929639, 0.98880435,
 0.87471021, 0.06501655, 0.98402531, 0.5677804 , 0.4730439 ,
 0.20232527, 0.84574641, 0.85611116, 0.49603454, 0.95120181,
 0.29422686, 0.27294921, 0.70411432, 0.84013328, 0.0950599 ,
 0.20102529, 1.00000001, 0.56697086, 0.14450017, 0.59397707],
[0.88658444, 0.95261945, 0.0026934 , 0.91736624, 0.85413465,
```

```
        0.06435452, 0.2258011 , 0.01012481, 0.72453228, 0.48317956,
        0.35222033, 0.00457984, 0.66683439, 0.16612386, 0.09315632,
        0.02201526, 0.5388942 , 0.3226246 , 0.11486152, 0.39618595,
        0.042756  , 0.03233316, 0.24143473, 0.83742395, 0.00666132,
        0.02350291, 0.56697086, 1.00000001, 0.01516065, 0.1230233 ],
       [0.04824031, 0.03112497, 0.78998881, 0.01604192, 0.0633618 ,
        0.85042846, 0.50471525, 0.88096224, 0.06768236, 0.18385345,
        0.33711029, 0.92665801, 0.11642732, 0.60093252, 0.75803776,
        0.96193195, 0.19934591, 0.36372852, 0.71693524, 0.25225537,
        0.9214309 , 0.87280898, 0.48167099, 0.03891567, 0.90301239,
        0.98146493, 0.14450017, 0.01516065, 1.00000001, 0.60099592],
       [0.29016677, 0.21416985, 0.36759489, 0.1502759 , 0.3316047 ,
        0.89141985, 0.87240119, 0.6092012 , 0.39247008, 0.67716046,
        0.82256577, 0.40200061, 0.5098488 , 0.84592925, 0.94557162,
        0.7328152 , 0.56366618, 0.8536701 , 0.89863533, 0.78020046,
        0.82780785, 0.83860974, 0.85293906, 0.27409508, 0.51520836,
        0.7137573 , 0.59397707, 0.1230233 , 0.60099592, 1.00000001]]),
'psi': array([[0.34260635],
       [0.25895978],
       [0.31738888],
       [0.17984492],
       [0.39198835],
       [0.88725669],
       [0.9336983 ],
       [0.54485813],
       [0.43979046],
       [0.72736558],
       [0.88650985],
       [0.36896682],
       [0.57235105],
       [0.90438298],
       [0.95095395],
       [0.69139459],
       [0.65171732],
       [0.9116596 ],
       [0.93072782],
       [0.82757135],
       [0.81121153],
       [0.78646058],
       [0.91902965],
       [0.31603674],
       [0.46047993],
       [0.68574036],
```

```
       [0.64941243],
       [0.15551492],
       [0.5758457 ],
       [0.98813074]]),
'one': array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.]),
'mu': 359.9030195419506,
'U': array([[ 1.00000001e+00,  9.82018332e-01,  1.18524623e-02,
         9.07963984e-01,  9.89890822e-01,  1.64572970e-01,
         4.29744016e-01,  3.73425289e-02,  9.43017886e-01,
         7.73448268e-01,  6.11343972e-01,  1.74715391e-02,
         9.17634655e-01,  3.31459694e-01,  2.23474532e-01,
         6.92485703e-02,  7.55487427e-01,  5.77926288e-01,
         2.55151953e-01,  6.80629064e-01,  1.17681774e-01,
         9.82338538e-02,  4.49240998e-01,  9.70460108e-01,
         2.56463507e-02,  7.12937154e-02,  8.47079255e-01,
         8.86584433e-01,  4.82403089e-02,  2.90166768e-01],
       [ 0.00000000e+00,  1.88785622e-01, -2.55297418e-02,
         2.81092605e-01, -6.91000614e-02, -2.40165979e-01,
        -4.31746594e-01, -7.19809745e-02, -2.25478052e-01,
        -4.99669484e-01, -4.99493602e-01, -3.52642615e-02,
        -3.51863716e-01, -3.63740565e-01, -3.01642300e-01,
        -1.20825089e-01, -3.62464668e-01, -5.06714737e-01,
        -3.18678622e-01, -5.23059800e-01, -1.84534122e-01,
        -1.64880145e-01, -4.35595470e-01, -2.01099359e-02,
        -5.12977473e-02, -1.21276281e-01, -4.35724400e-01,
         4.34234768e-01, -8.60653152e-02, -3.74918550e-01],
       [ 0.00000000e+00,  0.00000000e+00,  9.99603805e-01,
        -6.39964678e-05,  2.27139604e-03,  5.25854769e-01,
         2.03228076e-01,  9.12959716e-01,  3.92038111e-03,
         4.95833908e-02,  1.16519867e-01,  9.12971477e-01,
         1.69522246e-02,  2.56044677e-01,  4.36421403e-01,
         7.83202848e-01,  4.06536008e-02,  1.33681060e-01,
         3.62094003e-01,  8.45610117e-02,  6.23641700e-01,
         7.05883923e-01,  1.86436188e-01, -1.66212980e-03,
         9.60213884e-01,  7.46367331e-01,  3.04071000e-02,
         3.27239439e-03,  7.87531831e-01,  3.54724666e-01],
       [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         3.10786682e-01, -1.07687542e-01, -3.90495102e-02,
        -1.36045125e-01, -1.37443971e-03,  1.57535336e-01,
         1.65605963e-02, -1.30225655e-01, -2.74333047e-03,
        -2.84297632e-02, -1.25229066e-01, -5.14960742e-02,
        -1.07091032e-02, -2.90981710e-01, -1.10053996e-01,
```

```
 -8.12641482e-02, -2.90366229e-02, -2.67353467e-02,
 -8.76871560e-03, -1.48373568e-01,  2.19835604e-01,
 -1.51031993e-03, -1.56200604e-02,  1.34734789e-02,
 -3.11482633e-02, -1.13126528e-02, -2.50189084e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  6.11522015e-02,  2.68192026e-01,
  4.44304522e-01,  5.15034533e-02, -7.94127005e-02,
  8.45036275e-02,  3.39982236e-01,  2.79517456e-02,
  6.36472951e-02,  4.55227104e-01,  3.06071443e-01,
  1.26794201e-01,  4.14340989e-01,  3.37605411e-01,
  3.75058581e-01,  1.64999604e-01,  2.14317418e-01,
  1.44019855e-01,  4.56189065e-01, -7.38962674e-02,
  3.45736048e-02,  1.44854621e-01,  4.84368345e-02,
  5.16219395e-02,  1.08827097e-01,  2.44708157e-01],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  7.51838407e-01,
  5.54823804e-01,  3.16499646e-01,  4.21359590e-02,
  1.99646888e-01,  3.76187376e-01,  1.86719312e-01,
  7.82925296e-02,  6.33481255e-01,  7.46421744e-01,
  5.65885955e-01,  1.26977389e-01,  4.14680599e-01,
  7.26217601e-01,  2.97560974e-01,  7.12467320e-01,
  6.15200272e-01,  5.26007699e-01,  7.59272886e-03,
  2.26310539e-01,  5.98643557e-01,  1.38180858e-01,
  7.91844124e-03,  5.02852426e-01,  6.65680481e-01],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  2.52723339e-01, -1.25301039e-01,  6.36359397e-02,
  1.75070444e-01,  2.49712658e-01, -1.07356599e-01,
  9.08836597e-02,  2.24063403e-01,  8.14068330e-02,
 -1.50314099e-01,  1.26885288e-01,  2.49046584e-01,
  1.44736913e-01,  2.21342817e-01, -7.69174411e-02,
 -1.12791736e-01,  2.57996976e-01,  1.41281123e-02,
 -1.08330480e-01, -1.45686598e-01,  1.45312511e-01,
  1.75361514e-04, -1.66619434e-01,  1.27728306e-01],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  2.03474755e-01,  2.64083322e-02,
  7.82075407e-02,  6.24537663e-02, -2.10868920e-01,
  3.20994311e-02, -6.44482800e-02,  4.42985584e-02,
  4.81065990e-02,  4.71958421e-03,  6.75355808e-02,
 -4.14247998e-02,  1.00427486e-01, -1.74215435e-02,
  2.06206753e-01, -3.67766414e-03,  4.96726298e-03,
  6.16083931e-02, -5.17716476e-02,  6.00748243e-02,
```

```
  4.95629090e-04, -1.55644879e-01,  1.97609081e-01],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  1.49048410e-01,
  1.86463542e-01,  9.24401854e-02,  3.95645182e-02,
  9.51167481e-02, -3.59664530e-02,  1.55829992e-02,
  1.24316809e-02,  6.29754414e-03,  8.18146238e-02,
 -1.22234471e-02,  1.94102649e-01,  3.02729543e-03,
  1.76850841e-02, -3.16100412e-03,  5.25635362e-02,
  7.69474611e-03,  1.37994390e-02,  1.82410501e-01,
 -3.34564296e-02,  2.44695686e-02,  9.98980398e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  1.75615936e-01,  9.99765158e-02, -3.70768688e-03,
  5.90090354e-02, -8.53754534e-02,  3.99767058e-02,
 -8.04891224e-03, -6.35865426e-02,  1.32444595e-01,
 -3.97576352e-02,  1.99376728e-01, -1.92686634e-02,
  5.71537689e-02, -1.32149245e-02, -2.33751593e-02,
 -8.61368796e-03, -3.75012311e-02,  1.04499384e-01,
  8.54456183e-02, -4.63081493e-02,  2.17513964e-01],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  6.57789127e-02,  1.27355301e-02,
  1.14824727e-02, -4.89572082e-02,  3.80366686e-02,
  3.04555006e-03, -1.72464063e-02,  3.98617501e-02,
  1.56660577e-03,  4.59183322e-02, -5.90144286e-03,
  1.72467104e-02, -4.79865548e-03, -6.72536022e-03,
 -1.90890109e-03,  1.48658176e-03,  2.47199806e-02,
 -5.14112523e-02, -1.63756651e-02,  6.80973294e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  2.67486289e-01,
  7.34317915e-03,  1.50766687e-02, -3.19392343e-02,
  6.43845771e-02,  2.57521765e-02, -6.85246294e-03,
 -3.02363990e-03, -7.79905238e-03,  3.88847553e-02,
 -2.86720960e-02,  5.95270951e-03, -1.68539758e-03,
  6.53518593e-02,  1.11216420e-01,  4.09853362e-03,
  3.09133749e-03,  2.06295503e-01, -7.46914625e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
```

```
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  2.92746869e-02,  8.70581839e-03, -3.11398195e-02,
  1.59819221e-02,  2.86839059e-05, -2.77905138e-03,
 -5.11868304e-03, -2.06329165e-02,  1.57241966e-02,
 -3.11948721e-02,  3.15545440e-03, -1.06713606e-02,
  1.26274933e-02,  2.67506335e-02,  1.36996076e-02,
 -3.59783771e-03,  3.84736941e-02, -1.11982777e-01],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  4.91469918e-02, -1.68882161e-03,
 -5.05507259e-03,  5.96578543e-03, -2.59480871e-02,
  4.60563231e-02,  7.55891391e-03,  1.22469312e-02,
 -3.97797885e-02,  4.49727091e-03, -7.51842909e-03,
 -7.22365802e-03,  4.01543617e-02,  1.33902573e-02,
  1.02403985e-02,  5.02694717e-02, -3.98856521e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  3.55048098e-02,
 -2.14107587e-02, -6.37004590e-03, -2.41184983e-03,
  2.46637422e-02,  2.81839992e-04, -1.19672300e-02,
  6.55876213e-03, -1.44340906e-03,  4.16730682e-04,
 -2.89512628e-02,  1.11665471e-03, -2.90958897e-03,
  8.36487749e-03, -3.40042524e-02,  5.14532924e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  2.75486672e-02,  2.13258419e-02, -7.01114495e-03,
  3.03926551e-03, -5.57142148e-03,  1.08991464e-02,
 -8.14491243e-03,  3.06181232e-03,  7.72250791e-04,
  2.50308904e-02,  3.01433821e-02, -4.97690270e-04,
  1.87408447e-02,  1.31624099e-02, -3.27014081e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
```

```
  0.00000000e+00,  4.51299543e-02, -4.52750911e-04,
 -9.21163440e-03,  8.97837486e-04, -3.27524895e-05,
  2.17886374e-02,  3.22580098e-03,  2.06478116e-03,
 -1.12114674e-02, -3.77122365e-03, -1.54647969e-03,
 -1.90544659e-02, -3.56079966e-03,  3.01534739e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  1.13532622e-02,
 -6.40733867e-03,  4.66602083e-03, -6.47823666e-04,
  9.81498795e-03, -6.71070492e-04,  4.03399637e-03,
 -6.07198861e-05,  1.41366805e-03, -4.90301845e-03,
 -4.29090947e-02,  6.78204962e-03,  3.20409418e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  1.63481781e-02, -9.11962670e-03,  8.20080939e-03,
 -3.78604038e-02, -7.26110881e-05,  5.33342270e-03,
 -3.40975762e-03,  3.73424149e-02, -5.58119683e-03,
  4.31991494e-03,  3.60675356e-02, -4.60521308e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  9.85730808e-03,  9.89645781e-04,
 -9.51253797e-03, -1.37672810e-04, -8.11761700e-03,
  1.89111044e-03,  6.82703671e-03,  6.83654004e-03,
 -2.14984627e-02,  4.86343724e-03,  3.40182087e-03],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  6.14730317e-03,
 -1.87358615e-02,  7.84322495e-04,  1.58191515e-04,
```

```
 -1.65912295e-02,  2.69289095e-02,  7.02847328e-05,
  1.69639054e-02,  2.64675763e-02, -9.07902721e-03],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  2.34434019e-02,  7.34743580e-04, -7.26017069e-04,
 -3.89616558e-03, -5.42451092e-03, -1.22162717e-03,
  2.35938284e-03,  3.44115320e-04,  2.18749536e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  8.16693155e-04, -2.26694989e-03,
 -5.72159713e-03,  3.76760892e-03,  1.30825054e-03,
 -1.22022904e-03, -9.74940281e-03, -7.49430314e-03],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  8.97806303e-03,
  1.05757942e-03, -1.66177039e-03, -3.55718010e-03,
 -2.10254054e-02, -2.33562066e-03,  5.44803638e-03],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  1.43345029e-02, -9.34299672e-03, -9.15681010e-04,
  1.06161191e-02,  3.10121931e-03,  3.42719197e-03],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
```

```
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  1.08265973e-02,  6.87137584e-04,
-2.75849961e-03,  4.20932296e-03,  4.54729205e-03],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  3.36354296e-03,
 2.66908742e-02, -3.13633110e-03, -1.32510293e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 7.63728962e-02,  2.89145347e-04,  5.28301608e-04],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  1.06033857e-02,  3.81400062e-03],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
```

```
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  1.06614839e-02]]),
 'SigmaSqr': 88328.00018837576,
 'Lambda': None,
 'de_bounds': [[-3, 2], [-3, 2], [-3, 2]]}
```

```
vars(Sf)
```

```
{'name': 'kriging',
 'seed': 123,
 'rng': Generator(PCG64) at 0x2BD9E8F20,
 'log': {'negLnLike': array([98.72749633]),
  'theta': array([ 0.8710282 , -0.19383022, -2.57843188]),
  'p': array([2.]),
  'Lambda': array([None], dtype=object)},
 'verbosity': 50,
 'noise': False,
 'var_type': ['num', 'num', 'factor'],
 'cod_type': 'norm',
 'use_cod_y': False,
 'log_level': 50,
 'sigma': 0,
 'eps': 1.4901161193847656e-08,
 'min_theta': -3,
 'max_theta': 2,
 'min_p': 1,
 'max_p': 2,
 'min_Lambda': 1e-09,
 'max_Lambda': 1.0,
 'n_theta': 3,
 'n_p': 1,
 'optim_p': False,
 'cnd_Psi': 3662668.077523232,
 'inf_Psi': False,
 'model_optimizer': <function scipy.optimize._differentialevolution.differential_evolution(fu
 'model_fun_evals': 100,
 'nat_X': array([[-2.84117593,  5.97308949,  1.        ],
```

```
       [-3.61017994,  6.90781409,  2.        ],
       [ 9.91204705,  5.09395275,  2.        ],
       [-4.4616725 ,  1.3617128 ,  1.        ],
       [-2.40987728,  8.05505365,  1.        ],
       [ 5.24351477, 10.3775177 ,  1.        ],
       [ 2.5878792 , 11.89311852,  1.        ],
       [ 8.12926647,  3.6850299 ,  2.        ],
       [-1.96370363,  0.38404591,  2.        ],
       [ 0.10043744,  1.74091748,  1.        ],
       [ 1.38422219,  7.917048  ,  2.        ],
       [ 9.25110552, 13.20863768,  1.        ],
       [-1.09216899,  4.49255254,  1.        ],
       [ 3.26443339, 14.63587834,  1.        ],
       [ 4.54069975,  9.187233  ,  2.        ],
       [ 7.04143871,  8.56765487,  1.        ],
       [-0.10907144, 14.06693628,  1.        ],
       [ 1.63462403,  7.36106735,  1.        ],
       [ 4.10147822, 12.56738914,  2.        ],
       [ 0.85028105,  2.73647896,  2.        ],
       [ 5.9642566 , 10.70838079,  1.        ],
       [ 6.3810468 ,  4.61751818,  1.        ],
       [ 2.41318418, 12.34362887,  1.        ],
       [-3.00723724,  0.98372404,  1.        ],
       [ 8.75164908,  6.26584373,  1.        ],
       [ 6.93615484, 11.37121875,  2.        ],
       [-0.50159055,  2.30946613,  2.        ],
       [-4.78793654,  9.78635061,  1.        ],
       [ 7.58244883, 13.69175437,  1.        ],
       [ 3.86695804,  3.09448894,  2.        ]]),
'nat_y': array([ 42.09388125,  33.965223  ,  -3.74411425, 222.41884106,
        19.25981051, 108.97027586,  95.49463536,   1.74135821,
        81.69079805,  46.3681054 ,  16.77800101, 128.83487789,
        35.98089141, 165.59047908,  58.42915074,  81.277265  ,
        91.82566843,  32.4736378 , 113.78079756,  10.34983517,
       121.38146188,  41.89850532, 101.79776372, 130.84111259,
        30.98328188, 111.57203528,  28.861847  ,  66.90159224,
       174.66155808,  -5.44983081]),
'n': 30,
'k': 3,
'cod_X': array([[0.13243284, 0.39216316, 0.        ],
       [0.08011959, 0.45774943, 1.        ],
       [1.        , 0.33047728, 1.        ],
       [0.02219486, 0.06859938, 0.        ],
```

```
       [0.16177292, 0.53824712, 0.        ],
       [0.68241242, 0.7012061 , 0.        ],
       [0.50175673, 0.80755037, 0.        ],
       [0.87872227, 0.23161822, 1.        ],
       [0.1921249 , 0.        , 1.        ],
       [0.33254282, 0.09520682, 0.        ],
       [0.41987521, 0.52856376, 1.        ],
       [0.95503794, 0.89985564, 0.        ],
       [0.25141304, 0.28827918, 0.        ],
       [0.54778088, 1.        , 0.        ],
       [0.63460182, 0.61768809, 1.        ],
       [0.8047203 , 0.57421451, 0.        ],
       [0.3182905 , 0.96007938, 0.        ],
       [0.43690937, 0.48955259, 0.        ],
       [0.60472277, 0.85486153, 1.        ],
       [0.38355265, 0.1650618 , 1.        ],
       [0.73144253, 0.72442157, 0.        ],
       [0.75979563, 0.29704758, 0.        ],
       [0.4898727 , 0.83916107, 0.        ],
       [0.12113614, 0.04207727, 0.        ],
       [0.92106127, 0.41270467, 0.        ],
       [0.79755813, 0.77093054, 1.        ],
       [0.29158849, 0.13509984, 1.        ],
       [0.        , 0.65972602, 0.        ],
       [0.84152375, 0.93375421, 0.        ],
       [0.588769  , 0.19018207, 1.        ]]),
'cod_y': array([ 42.09388125,  33.965223  ,  -3.74411425, 222.41884106,
        19.25981051, 108.97027586,  95.49463536,   1.74135821,
        81.69079805,  46.3681054 ,  16.77800101, 128.83487789,
        35.98089141, 165.59047908,  58.42915074,  81.277265  ,
        91.82566843,  32.4736378 , 113.78079756,  10.34983517,
       121.38146188,  41.89850532, 101.79776372, 130.84111259,
        30.98328188, 111.57203528,  28.861847  ,  66.90159224,
       174.66155808,  -5.44983081]),
'num_mask': array([ True,  True, False]),
'factor_mask': array([False, False,  True]),
'int_mask': array([False, False, False]),
'ordered_mask': array([ True,  True, False]),
'nat_range_X': [[-4.787936542018484, 9.912047049457485],
 [0.3840459056967906, 14.63587833590837],
 [1.0, 2.0]],
'nat_range_y': [-5.449830806521344, 222.4188410638947],
'nat_mean_X': array([2.50508374, 7.5128501 , 1.36666667]),
```

```
'nat_std_X': array([4.3289971 , 4.28784459, 0.48189441]),
'nat_mean_y': 72.90095819227218,
'nat_std_y': 56.137717999904446,
'mean_cod_y': array([ 42.09388125,  33.965223  ,  -3.74411425, 222.41884106,
        19.25981051, 108.97027586,  95.49463536,   1.74135821,
        81.69079805,  46.3681054 ,  16.77800101, 128.83487789,
        35.98089141, 165.59047908,  58.42915074,  81.277265  ,
        91.82566843,  32.4736378 , 113.78079756,  10.34983517,
       121.38146188,  41.89850532, 101.79776372, 130.84111259,
        30.98328188, 111.57203528,  28.861847  ,  66.90159224,
       174.66155808,  -5.44983081]),
'theta': array([ 0.8710282 , -0.19383022, -2.57843188]),
'x0_theta': array([0.1, 0.1, 0.1]),
'p': array([2.]),
'pen_val': 10241.66847530361,
'negLnLike': 98.72749632571899,
'gen': <spotPython.design.spacefilling.spacefilling at 0x2bda72f50>,
'LnDetPsi': -95.5418041057407,
'Psi': array([[1.00000001e+00, 3.59483038e-01, 1.36677525e-03, 8.54445211e-01,
        9.80145512e-01, 9.93868688e-02, 3.24986342e-01, 5.77051620e-03,
        3.24684584e-01, 7.01882153e-01, 1.96743893e-01, 5.55454230e-03,
        8.93957187e-01, 2.19073572e-01, 5.46737836e-02, 3.40591794e-02,
        6.29334806e-01, 4.99105495e-01, 6.11464290e-02, 2.22774746e-01,
        6.47727047e-02, 5.33764946e-02, 3.40533883e-01, 9.23684320e-01,
        9.83640732e-03, 1.25368458e-02, 2.92142369e-01, 8.38501414e-01,
        1.97631243e-02, 7.62687083e-02],
       [3.59483038e-01, 1.00000001e+00, 1.83988163e-03, 3.25676787e-01,
        3.48649099e-01, 2.39099874e-02, 9.07798001e-02, 8.46547730e-03,
        7.96667259e-01, 2.10644864e-01, 4.22755675e-01, 1.09923790e-03,
        2.90425788e-01, 6.00054772e-02, 1.00165067e-01, 7.37154541e-03,
        2.05358372e-01, 1.42763725e-01, 1.16961675e-01, 4.77601213e-01,
        1.50291434e-02, 1.16879338e-02, 9.62607950e-02, 3.25276464e-01,
        1.91862937e-03, 2.04970705e-02, 6.71046246e-01, 3.41705340e-01,
        4.28412235e-03, 1.39692275e-01],
       [1.36677525e-03, 1.83988163e-03, 1.00000001e+00, 2.89203777e-04,
        1.93315554e-03, 1.59225932e-01, 5.02724804e-02, 8.90878585e-01,
        7.30193520e-03, 1.29613554e-02, 7.99889868e-02, 2.94493138e-01,
        5.71215595e-03, 6.04179667e-02, 3.51725233e-01, 2.66766208e-01,
        9.03246178e-03, 3.43124707e-02, 2.62637322e-01, 5.83543092e-02,
        1.94904814e-01, 2.39440277e-01, 4.50814151e-02, 1.12194054e-03,
        3.49717329e-01, 6.51364649e-01, 2.34358082e-02, 2.03457936e-04,
        2.41825865e-01, 2.81055002e-01],
       [8.54445211e-01, 3.25676787e-01, 2.89203777e-04, 1.00000001e+00,
```

```
 7.51318591e-01, 3.03471643e-02, 1.27662371e-01, 1.55164217e-03,
 2.95945056e-01, 4.88632404e-01, 9.92058370e-02, 9.99389888e-04,
 6.56193200e-01, 7.36931217e-02, 1.86899123e-02, 8.97103711e-03,
 3.13459166e-01, 2.48728324e-01, 1.98975307e-02, 1.38588195e-01,
 1.80767637e-02, 1.69736274e-02, 1.34625698e-01, 9.29422500e-01,
 2.28938261e-03, 3.07971496e-03, 2.13932235e-01, 7.96689042e-01,
 4.22289499e-03, 3.35488305e-02],
[9.80145512e-01, 3.48649099e-01, 1.93315554e-03, 7.51318591e-01,
 1.00000001e+00, 1.31178745e-01, 4.04413042e-01, 7.59959533e-03,
 3.03535873e-01, 7.10124309e-01, 2.24233429e-01, 8.56920304e-03,
 9.05111740e-01, 2.88332428e-01, 6.95789247e-02, 4.63038655e-02,
 7.43850425e-01, 5.68917406e-01, 8.02918425e-02, 2.33490023e-01,
 8.77195505e-02, 6.75644642e-02, 4.24069369e-01, 8.43812453e-01,
 1.36500789e-02, 1.76274520e-02, 2.92515498e-01, 8.15536891e-01,
 2.92014691e-02, 8.78314046e-02],
[9.93868688e-02, 2.39099874e-02, 1.59225932e-01, 3.03471643e-02,
 1.31178745e-01, 1.00000001e+00, 7.78996548e-01, 2.39911563e-01,
 4.50098778e-02, 3.18349847e-01, 2.16267327e-01, 5.61278932e-01,
 2.25497009e-01, 8.25453581e-01, 3.60072640e-01, 8.85610509e-01,
 3.57691991e-01, 6.20934782e-01, 3.46469939e-01, 1.57606976e-01,
 9.81956856e-01, 8.61539945e-01, 7.50026719e-01, 7.28804372e-02,
 6.20972086e-01, 3.32328569e-01, 9.63193562e-02, 3.13836824e-02,
 8.00332870e-01, 2.91624146e-01],
[3.24986342e-01, 9.07798001e-02, 5.02724804e-02, 1.27662371e-01,
 4.04413042e-01, 7.78996548e-01, 1.00000001e+00, 1.03498016e-01,
 1.18866903e-01, 5.84198303e-01, 3.32993648e-01, 2.16062464e-01,
 5.28209588e-01, 9.61324893e-01, 3.15308061e-01, 4.88271787e-01,
 7.67202307e-01, 9.08496481e-01, 3.39523354e-01, 2.54614839e-01,
 6.72716060e-01, 5.16049281e-01, 9.98312494e-01, 2.34219763e-01,
 2.45070529e-01, 1.91852049e-01, 1.98371721e-01, 1.51870045e-01,
 4.19789287e-01, 2.72481429e-01],
[5.77051620e-03, 8.46547730e-03, 8.90878585e-01, 1.55164217e-03,
 7.59959533e-03, 2.39911563e-01, 1.03498016e-01, 1.00000001e+00,
 2.90925990e-02, 3.96141835e-02, 1.97722555e-01, 2.64726903e-01,
 1.97194420e-02, 1.11729144e-01, 5.83786916e-01, 3.27650104e-01,
 2.53883276e-02, 8.26586549e-02, 4.46438311e-01, 1.61252485e-01,
 2.68043820e-01, 3.30272947e-01, 9.44418195e-02, 5.05322436e-03,
 3.55472730e-01, 7.90495704e-01, 7.67264275e-02, 1.05426654e-03,
 2.65592685e-01, 5.34824217e-01],
[3.24684584e-01, 7.96667259e-01, 7.30193520e-03, 2.95945056e-01,
 3.03535873e-01, 4.50098778e-02, 1.18866903e-01, 2.90925990e-02,
 1.00000001e+00, 3.15905217e-01, 5.68798327e-01, 2.89971303e-03,
 3.39831519e-01, 7.57816098e-02, 1.82866385e-01, 1.83238914e-02,
```

```
 1.81193935e-01, 2.02174985e-01, 1.76811850e-01, 7.48465197e-01,
 3.02830430e-02, 3.17137285e-02, 1.21307215e-01, 3.53957206e-01,
 6.36283669e-03, 4.48660572e-02, 9.18335326e-01, 2.11648813e-01,
 9.17148285e-03, 3.03555764e-01],
[7.01882153e-01, 2.10644864e-01, 1.29613554e-02, 4.88632404e-01,
 7.10124309e-01, 3.18349847e-01, 5.84198303e-01, 3.96141835e-02,
 3.15905217e-01, 1.00000001e+00, 3.08244677e-01, 3.71143371e-02,
 9.29818677e-01, 4.19719722e-01, 1.56816434e-01, 1.64717405e-01,
 6.18646624e-01, 8.34885356e-01, 1.46636560e-01, 3.59709838e-01,
 2.37936056e-01, 2.50949159e-01, 5.83832968e-01, 7.16122519e-01,
 7.14929073e-02, 5.50770580e-02, 3.62953099e-01, 3.58555906e-01,
 9.30130807e-02, 2.24560443e-01],
[1.96743893e-01, 4.22755675e-01, 7.99889868e-02, 9.92058370e-02,
 2.24233429e-01, 2.16267327e-01, 3.32993648e-01, 1.97722555e-01,
 5.68798327e-01, 3.08244677e-01, 1.00000001e+00, 4.01005398e-02,
 2.87127347e-01, 2.82577394e-01, 7.06315400e-01, 1.22228968e-01,
 3.02446957e-01, 3.66729750e-01, 7.24672671e-01, 9.09948686e-01,
 1.74491951e-01, 1.50636312e-01, 3.33488303e-01, 1.62900548e-01,
 5.64114632e-02, 3.33691730e-01, 8.01421113e-01, 9.81751551e-02,
 8.83761484e-02, 7.51833449e-01],
[5.55454230e-03, 1.09923790e-03, 2.94493138e-01, 9.99389888e-04,
 8.56920304e-03, 5.61278932e-01, 2.16062464e-01, 2.64726903e-01,
 2.89971303e-03, 3.71143371e-02, 4.01005398e-02, 1.00000001e+00,
 1.98766749e-02, 2.89715061e-01, 1.63012055e-01, 7.89966793e-01,
 4.90425074e-02, 1.22143513e-01, 1.47608278e-01, 2.29975823e-02,
 6.76249793e-01, 5.97016704e-01, 1.99848215e-01, 3.55954247e-03,
 8.51754629e-01, 3.02728918e-01, 9.60933737e-03, 1.09784974e-03,
 9.08025202e-01, 9.83555229e-02],
[8.93957187e-01, 2.90425788e-01, 5.71215595e-03, 6.56193200e-01,
 9.05111740e-01, 2.25497009e-01, 5.28209588e-01, 1.97194420e-02,
 3.39831519e-01, 9.29818677e-01, 2.87127347e-01, 1.98766749e-02,
 1.00000001e+00, 3.76496247e-01, 1.15265421e-01, 9.75668606e-02,
 7.24647806e-01, 7.54569680e-01, 1.18481371e-01, 3.19990920e-01,
 1.59777082e-01, 1.46529554e-01, 5.39697066e-01, 8.47972688e-01,
 3.53648842e-02, 3.45459068e-02, 3.58075953e-01, 5.72362722e-01,
 5.76001277e-02, 1.56948669e-01],
[2.19073572e-01, 6.00054772e-02, 6.04179667e-02, 7.36931217e-02,
 2.88332428e-01, 8.25453581e-01, 9.61324893e-01, 1.11729144e-01,
 7.57816098e-02, 4.19719722e-01, 2.82577394e-01, 2.89715061e-01,
 3.76496247e-01, 1.00000001e+00, 3.16778264e-01, 5.45211097e-01,
 6.75459177e-01, 7.72522783e-01, 3.54313043e-01, 1.92711788e-01,
 7.41373150e-01, 5.21913079e-01, 9.59374688e-01, 1.43727767e-01,
 2.84757296e-01, 2.23762132e-01, 1.39952654e-01, 9.98807345e-02,
```

```
 5.25206713e-01, 2.38785620e-01],
[5.46737836e-02, 1.00165067e-01, 3.51725233e-01, 1.86899123e-02,
 6.95789247e-02, 3.60072640e-01, 3.15308061e-01, 5.83786916e-01,
 1.82866385e-01, 1.56816434e-01, 7.06315400e-01, 1.63012055e-01,
 1.15265421e-01, 3.16778264e-01, 1.00000001e+00, 2.96337738e-01,
 1.62271132e-01, 2.72282384e-01, 9.58262299e-01, 5.49120484e-01,
 3.40623844e-01, 3.06584559e-01, 3.05123651e-01, 4.19565931e-02,
 1.94631970e-01, 8.08682676e-01, 3.59397484e-01, 1.84330876e-02,
 2.51053321e-01, 8.75838321e-01],
[3.40591794e-02, 7.37154541e-03, 2.66766208e-01, 8.97103711e-03,
 4.63038655e-02, 8.85610509e-01, 4.88271787e-01, 3.27650104e-01,
 1.83238914e-02, 1.64717405e-01, 1.22228968e-01, 7.89966793e-01,
 9.75668606e-02, 5.45211097e-01, 2.96337738e-01, 1.00000001e+00,
 1.56689431e-01, 3.64275429e-01, 2.59856271e-01, 8.84598467e-02,
 9.47110571e-01, 9.37853627e-01, 4.57709687e-01, 2.59007294e-02,
 8.89344950e-01, 3.58743800e-01, 4.59624245e-02, 8.09428831e-03,
 9.11380416e-01, 2.36711117e-01],
[6.29334806e-01, 2.05358372e-01, 9.03246178e-03, 3.13459166e-01,
 7.43850425e-01, 3.57691991e-01, 7.67202307e-01, 2.53883276e-02,
 1.81193935e-01, 6.18646624e-01, 3.02446957e-01, 4.90425074e-02,
 7.24647806e-01, 6.75459177e-01, 1.62271132e-01, 1.56689431e-01,
 1.00000001e+00, 7.81732559e-01, 1.98548232e-01, 2.37841039e-01,
 2.71466319e-01, 1.77323079e-01, 7.96028905e-01, 4.36851359e-01,
 5.54897022e-02, 6.52386081e-02, 2.36722617e-01, 4.44623522e-01,
 1.30711629e-01, 1.46173105e-01],
[4.99105495e-01, 1.42763725e-01, 3.43124707e-02, 2.48728324e-01,
 5.68917406e-01, 6.20934782e-01, 9.08496481e-01, 8.26586549e-02,
 2.02174985e-01, 8.34885356e-01, 3.66729750e-01, 1.22143513e-01,
 7.54569680e-01, 7.72522783e-01, 2.72282384e-01, 3.64275429e-01,
 7.81732559e-01, 1.00000001e+00, 2.73989882e-01, 3.36707245e-01,
 5.06661020e-01, 4.50045951e-01, 9.05682247e-01, 4.19337311e-01,
 1.74548570e-01, 1.33032842e-01, 2.90158549e-01, 2.37645988e-01,
 2.61120027e-01, 2.92667383e-01],
[6.11464290e-02, 1.16961675e-01, 2.62637322e-01, 1.98975307e-02,
 8.02918425e-02, 3.46469939e-01, 3.39523354e-01, 4.46438311e-01,
 1.76811850e-01, 1.46636560e-01, 7.24672671e-01, 1.47608278e-01,
 1.18481371e-01, 3.54313043e-01, 9.58262299e-01, 2.59856271e-01,
 1.98548232e-01, 2.73989882e-01, 1.00000001e+00, 5.12732763e-01,
 3.22965403e-01, 2.52126269e-01, 3.33480016e-01, 4.24048045e-02,
 1.54323120e-01, 7.55163150e-01, 3.46404169e-01, 2.37140018e-02,
 2.41555466e-01, 7.52288147e-01],
[2.22774746e-01, 4.77601213e-01, 5.83543092e-02, 1.38588195e-01,
 2.33490023e-01, 1.57606976e-01, 2.54614839e-01, 1.61252485e-01,
```

```
 7.48465197e-01, 3.59709838e-01, 9.09948686e-01, 2.29975823e-02,
 3.19990920e-01, 1.92711788e-01, 5.49120484e-01, 8.84598467e-02,
 2.37841039e-01, 3.36707245e-01, 5.12732763e-01, 1.00000001e+00,
 1.22510865e-01, 1.27068894e-01, 2.52887269e-01, 2.18411563e-01,
 4.13333579e-02, 2.21231586e-01, 9.38550369e-01, 1.05426037e-01,
 5.30430897e-02, 7.31003233e-01],
[6.47727047e-02, 1.50291434e-02, 1.94904814e-01, 1.80767637e-02,
 8.77195505e-02, 9.81956856e-01, 6.72716060e-01, 2.68043820e-01,
 3.02830430e-02, 2.37936056e-01, 1.74491951e-01, 6.76249793e-01,
 1.59777082e-01, 7.41373150e-01, 3.40623844e-01, 9.47110571e-01,
 2.71466319e-01, 5.06661020e-01, 3.22965403e-01, 1.22510865e-01,
 1.00000001e+00, 8.84382315e-01, 6.42717662e-01, 4.66209645e-02,
 7.19385736e-01, 3.55629492e-01, 6.99558442e-02, 1.87202417e-02,
 8.88617650e-01, 2.63443560e-01],
[5.33764946e-02, 1.16879338e-02, 2.39440277e-01, 1.69736274e-02,
 6.75644642e-02, 8.61539945e-01, 5.16049281e-01, 3.30272947e-01,
 3.17137285e-02, 2.50949159e-01, 1.50636312e-01, 5.97016704e-01,
 1.46529554e-01, 5.21913079e-01, 3.06584559e-01, 9.37853627e-01,
 1.77323079e-01, 4.50045951e-01, 2.52126269e-01, 1.27068894e-01,
 8.84382315e-01, 1.00000001e+00, 4.82164370e-01, 4.63065077e-02,
 8.17252303e-01, 3.15273599e-01, 7.09541146e-02, 1.26028772e-02,
 7.34122038e-01, 2.93858532e-01],
[3.40533883e-01, 9.62607950e-02, 4.50814151e-02, 1.34625698e-01,
 4.24069369e-01, 7.50026719e-01, 9.98312494e-01, 9.44418195e-02,
 1.21307215e-01, 5.83832968e-01, 3.33488303e-01, 1.99848215e-01,
 5.39697066e-01, 9.59374688e-01, 3.05123651e-01, 4.57709687e-01,
 7.96028905e-01, 9.05682247e-01, 3.33480016e-01, 2.52887269e-01,
 6.42717662e-01, 4.82164370e-01, 1.00000001e+00, 2.42456354e-01,
 2.23592873e-01, 1.81510159e-01, 2.00009376e-01, 1.64674661e-01,
 3.96693164e-01, 2.61264913e-01],
[9.23684320e-01, 3.25276464e-01, 1.12194054e-03, 9.29422500e-01,
 8.43812453e-01, 7.28804372e-02, 2.34219763e-01, 5.05322436e-03,
 3.53957206e-01, 7.16122519e-01, 1.62900548e-01, 3.55954247e-03,
 8.47972688e-01, 1.43727767e-01, 4.19565931e-02, 2.59007294e-02,
 4.36851359e-01, 4.19337311e-01, 4.24048045e-02, 2.18411563e-01,
 4.66209645e-02, 4.63065077e-02, 2.42456354e-01, 1.00000001e+00,
 7.88610805e-03, 8.73988997e-03, 2.94808663e-01, 7.02447013e-01,
 1.27139663e-02, 7.14339902e-02],
[9.83640732e-03, 1.91862937e-03, 3.49717329e-01, 2.28938261e-03,
 1.36500789e-02, 6.20972086e-01, 2.45070529e-01, 3.55472730e-01,
 6.36283669e-03, 7.14929073e-02, 5.64114632e-02, 8.51754629e-01,
 3.53648842e-02, 2.84757296e-01, 1.94631970e-01, 8.89344950e-01,
 5.54897022e-02, 1.74548570e-01, 1.54323120e-01, 4.13333579e-02,
```

```
 7.19385736e-01, 8.17252303e-01, 2.23592873e-01, 7.88610805e-03,
 1.00000001e+00, 3.02562674e-01, 1.84328570e-02, 1.75920629e-03,
 8.01909715e-01, 1.56896172e-01],
[1.25368458e-02, 2.04970705e-02, 6.51364649e-01, 3.07971496e-03,
 1.76274520e-02, 3.32328569e-01, 1.91852049e-01, 7.90495704e-01,
 4.48660572e-02, 5.50770580e-02, 3.33691730e-01, 3.02728918e-01,
 3.45459068e-02, 2.23762132e-01, 8.08682676e-01, 3.58743800e-01,
 6.52386081e-02, 1.33032842e-01, 7.55163150e-01, 2.21231586e-01,
 3.55629492e-01, 3.15273599e-01, 1.81510159e-01, 8.73988997e-03,
 3.02562674e-01, 1.00000001e+00, 1.15206942e-01, 3.23230974e-03,
 3.56532347e-01, 5.82881378e-01],
[2.92142369e-01, 6.71046246e-01, 2.34358082e-02, 2.13932235e-01,
 2.92515498e-01, 9.63193562e-02, 1.98371721e-01, 7.67264275e-02,
 9.18335326e-01, 3.62953099e-01, 8.01421113e-01, 9.60933737e-03,
 3.58075953e-01, 1.39952654e-01, 3.59397484e-01, 4.59624245e-02,
 2.36722617e-01, 2.90158549e-01, 3.46404169e-01, 9.38550369e-01,
 6.99558442e-02, 7.09541146e-02, 2.00009376e-01, 2.94808663e-01,
 1.84328570e-02, 1.15206942e-01, 1.00000001e+00, 1.63993379e-01,
 2.58495159e-02, 5.17787147e-01],
[8.38501414e-01, 3.41705340e-01, 2.03457936e-04, 7.96689042e-01,
 8.15536891e-01, 3.13836824e-02, 1.51870045e-01, 1.05426654e-03,
 2.11648813e-01, 3.58555906e-01, 9.81751551e-02, 1.09784974e-03,
 5.72362722e-01, 9.98807345e-02, 1.84330876e-02, 8.09428831e-03,
 4.44623522e-01, 2.37645988e-01, 2.37140018e-02, 1.05426037e-01,
 1.87202417e-02, 1.26028772e-02, 1.64674661e-01, 7.02447013e-01,
 1.75920629e-03, 3.23230974e-03, 1.63993379e-01, 1.00000001e+00,
 4.94103743e-03, 2.43083925e-02],
[1.97631243e-02, 4.28412235e-03, 2.41825865e-01, 4.22289499e-03,
 2.92014691e-02, 8.00332870e-01, 4.19789287e-01, 2.65592685e-01,
 9.17148285e-03, 9.30130807e-02, 8.83761484e-02, 9.08025202e-01,
 5.76001277e-02, 5.25206713e-01, 2.51053321e-01, 9.11380416e-01,
 1.30711629e-01, 2.61120027e-01, 2.41555466e-01, 5.30430897e-02,
 8.88617650e-01, 7.34122038e-01, 3.96693164e-01, 1.27139663e-02,
 8.01909715e-01, 3.56532347e-01, 2.58495159e-02, 4.94103743e-03,
 1.00000001e+00, 1.60645414e-01],
[7.62687083e-02, 1.39692275e-01, 2.81055002e-01, 3.35488305e-02,
 8.78314046e-02, 2.91624146e-01, 2.72481429e-01, 5.34824217e-01,
 3.03555764e-01, 2.24560443e-01, 7.51833449e-01, 9.83555229e-02,
 1.56948669e-01, 2.38785620e-01, 8.75838321e-01, 2.36711117e-01,
 1.46173105e-01, 2.92667383e-01, 7.52288147e-01, 7.31003233e-01,
 2.63443560e-01, 2.93858532e-01, 2.61264913e-01, 7.14339902e-02,
 1.56896172e-01, 5.82881378e-01, 5.17787147e-01, 2.43083925e-02,
 1.60645414e-01, 1.00000001e+00]]),
```

```
'psi': array([[0.09531992],
       [0.18123704],
       [0.23503466],
       [0.04112177],
       [0.11175682],
       [0.30423134],
       [0.31527597],
       [0.4623973 ],
       [0.33980766],
       [0.2408812 ],
       [0.85042037],
       [0.09472998],
       [0.18190666],
       [0.28195918],
       [0.91704606],
       [0.22754675],
       [0.18967424],
       [0.32581282],
       [0.83960551],
       [0.77788883],
       [0.27019244],
       [0.27172271],
       [0.30596201],
       [0.08329111],
       [0.13832731],
       [0.58613627],
       [0.57052484],
       [0.03412543],
       [0.16397381],
       [0.97586783]]),
'one': array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.]),
'mu': 177.32613758509206,
'U': array([[ 1.00000001e+00,  3.59483035e-01,  1.36677524e-03,
         8.54445204e-01,  9.80145504e-01,  9.93868680e-02,
         3.24986339e-01,  5.77051615e-03,  3.24684582e-01,
         7.01882148e-01,  1.96743892e-01,  5.55454226e-03,
         8.93957180e-01,  2.19073571e-01,  5.46737832e-02,
         3.40591792e-02,  6.29334801e-01,  4.99105492e-01,
         6.11464285e-02,  2.22774745e-01,  6.47727042e-02,
         5.33764942e-02,  3.40533881e-01,  9.23684313e-01,
         9.83640724e-03,  1.25368457e-02,  2.92142367e-01,
         8.38501407e-01,  1.97631242e-02,  7.62687077e-02],
```

```
[ 0.00000000e+00,  9.33151629e-01,  1.44515540e-03,
  1.98448262e-02, -3.96139447e-03, -1.26645072e-02,
 -2.79132294e-02,  6.84891334e-03,  7.28658279e-01,
 -4.46549733e-02,  3.77247997e-01, -9.61822049e-04,
 -3.31528671e-02, -2.00907916e-02,  8.62783361e-02,
 -5.22117900e-03, -2.23723690e-02, -3.92821819e-02,
  1.01784714e-01,  4.25994511e-01, -8.84694901e-03,
 -8.03729012e-03, -2.80290546e-02, -7.25753041e-03,
 -1.73325761e-03,  1.71357866e-02,  6.06574541e-01,
  4.31637350e-02, -3.02243004e-03,  1.20318033e-01],
[ 0.00000000e+00,  0.00000000e+00,  9.99998029e-01,
 -9.07311414e-04,  5.99242953e-04,  1.59108708e-01,
  4.98687344e-02,  8.90862556e-01,  5.80515134e-03,
  1.20665974e-02,  7.91750562e-02,  2.94487516e-01,
  4.53823740e-03,  6.01476952e-02,  3.51526513e-01,
  2.66727728e-01,  8.20465028e-03,  3.36871410e-02,
  2.62407171e-01,  5.74343111e-02,  1.94829453e-01,
  2.39379411e-01,  4.46565761e-02, -1.30040298e-04,
  3.49707079e-01,  6.51324034e-01,  2.21599645e-02,
 -1.00496531e-03,  2.41803697e-01,  2.80777435e-01],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  5.19161600e-01, -1.65811335e-01, -1.04356297e-01,
 -2.87813021e-01, -5.21335441e-03,  7.83001302e-03,
 -2.12246655e-01, -1.46997818e-01, -6.66532931e-03,
 -2.06067234e-01, -2.17735613e-01, -5.66665063e-02,
 -3.81096111e-02, -4.31121093e-01, -3.40779935e-01,
 -6.57416820e-02, -1.15883559e-01, -7.11062329e-02,
 -5.44280853e-02, -2.99993478e-01,  2.70298832e-01,
 -1.11017436e-02, -1.42179925e-02, -9.18879015e-02,
  1.52896607e-01, -2.38543083e-02, -6.50116243e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  1.08652449e-01,  1.50168561e-01,
  3.49886480e-01,  5.26913797e-03, -9.68310526e-02,
 -1.21482306e-01,  7.79445481e-02,  1.69299133e-02,
 -4.96878464e-02,  3.44122835e-01,  6.19023843e-02,
  5.91002862e-02,  5.10181258e-01,  2.12056214e-01,
  8.93186786e-02, -2.23035360e-02,  1.13120872e-01,
  5.56605452e-02,  3.71979685e-01, -1.54093501e-01,
  1.79632218e-02,  2.44780033e-02, -6.14158642e-02,
  1.76775466e-01,  5.26315324e-02,  2.39818126e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  9.64989876e-01,
  6.79625851e-01,  9.98409359e-02,  3.77237460e-02,
```

```
  2.50987185e-01,  1.67720742e-01,  5.29146758e-01,
  1.25871379e-01,  7.45559708e-01,  2.94916284e-01,
  8.56867708e-01,  1.78190702e-01,  5.16136124e-01,
  2.89803205e-01,  1.27440626e-01,  9.53378512e-01,
  8.33177143e-01,  6.44106277e-01,  3.35283398e-02,
  5.80809239e-01,  2.30581381e-01,  7.36526032e-02,
 -6.40796295e-02,  7.76655168e-01,  2.38870884e-01],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  4.73254344e-01, -2.91828027e-02,  9.27492435e-02,
  3.48836247e-01,  1.94545365e-01, -3.54817632e-01,
  2.30457106e-01,  4.15835515e-01,  9.30093040e-02,
 -3.17462218e-01,  2.91497920e-01,  4.65841746e-01,
  1.31591750e-01,  1.67102028e-01, -1.40056771e-01,
 -2.42677020e-01,  4.86821350e-01,  9.03592614e-02,
 -3.79980214e-01, -2.87168163e-02,  1.35744427e-01,
 -1.97931708e-01, -3.20951815e-01,  1.00594236e-01],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  4.42050815e-01,  3.74351478e-02,
  2.21126918e-02,  2.51609852e-01, -1.37892516e-01,
  9.25400759e-03, -1.86205973e-02,  5.48277824e-01,
 -1.23366335e-02,  8.61776494e-04,  2.08258419e-02,
  4.20115168e-01,  2.20674575e-01, -1.37446369e-02,
  5.86394417e-02, -1.67285407e-03,  3.16520747e-03,
 -5.73306390e-02,  4.20773358e-01,  1.07673546e-01,
 -6.10210394e-03, -8.42085419e-02,  5.92798802e-01],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  5.85439062e-01,
  1.15693408e-01,  3.49280630e-01,  3.39723838e-02,
  7.52270729e-02, -2.05453775e-02,  1.13363778e-01,
  2.24182987e-02,  2.06672069e-02,  4.83331544e-02,
  8.80842634e-02,  5.73204689e-01,  6.17962170e-03,
  2.31419830e-02, -1.95022981e-04,  5.55798731e-02,
  3.36619827e-02,  8.92199425e-03,  6.09354798e-01,
 -9.41541210e-02,  2.12806332e-02,  2.59276132e-01],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  4.96850051e-01,  1.68660458e-02,  4.09465014e-02,
  2.61982085e-01, -1.39824026e-01, -3.65480238e-02,
  5.99977618e-02, -5.06195016e-03,  2.77236124e-01,
```

```
 -7.27111041e-02,  6.62164663e-02, -4.95629318e-03,
  1.54998571e-01, -1.38465023e-02,  1.20150692e-01,
  8.90934852e-02, -3.83468349e-02,  3.83238348e-02,
 -1.56888201e-01, -1.23800611e-02,  4.82621910e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  7.30380940e-01,  2.21431939e-02,
 -2.72765977e-03, -1.13109893e-02,  5.16899906e-01,
 -2.08235855e-03, -1.46035552e-02,  1.37892682e-02,
  5.85161831e-01,  5.13207013e-01, -1.20476530e-03,
 -8.36026721e-03, -2.16308330e-03, -1.33829315e-02,
  2.85383429e-03,  1.74932527e-01,  3.08475475e-01,
  1.64569837e-02,  6.86830839e-03,  4.89634951e-01],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  6.96194687e-01,
  2.33122906e-02,  2.96328892e-02, -4.88987648e-03,
  1.99723935e-01,  5.84393903e-02, -2.11131497e-02,
  9.25232386e-03, -9.19537611e-03,  8.62836617e-02,
 -1.33254376e-03,  1.26095014e-02,  1.59655756e-02,
  4.21460462e-01,  4.82252521e-02, -3.72127503e-03,
 -4.75458621e-02,  4.29234144e-01, -2.30772786e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  1.04421895e-01, -9.94969447e-04, -1.92102297e-02,
  3.99650814e-02,  6.25801828e-02, -9.61960314e-03,
  4.50145632e-05, -5.49004991e-02,  2.76156058e-02,
 -2.69906204e-02,  8.60300382e-03,  8.52405809e-02,
  2.71369771e-02,  1.59508654e-02, -1.50045749e-02,
 -3.10474525e-01,  4.86080657e-02, -9.32093679e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  1.79027904e-01,  3.68460449e-02,
 -6.22811066e-02,  6.16741334e-02, -1.36366902e-01,
  9.58096372e-02, -1.04979658e-02,  2.76607887e-02,
 -1.81293081e-01,  1.77600475e-02,  4.47131203e-02,
 -1.25300746e-01,  7.77330161e-02,  2.27389448e-04,
```

```
 -1.69972431e-02,  1.22505892e-01, -3.06214689e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  4.23786319e-01,
 -3.84633948e-02, -1.18034593e-02, -1.40814186e-03,
  4.67206992e-01, -6.57328339e-02, -6.14422592e-03,
 -6.63010816e-02, -2.15387219e-03,  4.79323140e-03,
 -7.02710626e-02,  4.34300861e-01, -9.18623451e-02,
 -3.64481003e-03, -7.07923386e-04,  1.67775058e-01],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  1.91351151e-01,  7.97306791e-02, -1.66413368e-02,
  2.53931498e-02, -1.91044635e-02,  2.74425265e-02,
  2.83293690e-01,  1.17150434e-02,  9.05162614e-03,
  3.83405914e-01, -2.64956431e-02, -1.45592027e-02,
 -1.73291002e-03, -3.15226146e-02, -2.23954113e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  1.48991278e-01, -2.85250573e-02,
 -4.61102545e-03, -1.43514859e-02,  3.93376130e-03,
  1.14964847e-02,  1.01186609e-02,  5.22698064e-02,
 -7.00937084e-02,  1.00372803e-02, -9.98534127e-03,
 -8.39465668e-02,  4.99788436e-03,  5.19249443e-03],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  6.05419324e-02,
 -4.96163628e-03,  2.02962016e-02,  2.83006309e-03,
  4.02030898e-02, -1.02075630e-03, -5.85333189e-02,
 -2.28895447e-02,  1.80246616e-02,  7.73716441e-03,
 -3.25859885e-04,  3.74988193e-02,  3.17323158e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
```

```
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  2.07957624e-01, -1.45404217e-01,  2.26566815e-03,
 -1.44462602e-03,  1.46513909e-04,  5.59228485e-05,
 -8.79356494e-03,  1.09971084e-01, -5.82541519e-02,
  2.17386233e-03,  1.10835275e-02, -2.71936829e-01],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  1.81187513e-01,  1.84177478e-03,
 -6.75539447e-03,  2.78298333e-04, -9.17937741e-03,
 -3.78699160e-03,  4.84686282e-02,  1.09679389e-01,
  9.07722362e-03,  5.28978616e-03,  1.28632916e-01],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  3.63349546e-02,
 -1.45902644e-01,  3.44323664e-03, -2.30394696e-02,
 -1.08899261e-01,  6.61216161e-02,  2.09763568e-03,
  6.75019226e-02,  1.30557456e-01, -3.89585038e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  1.06047719e-01,  1.81854528e-03,  5.29711470e-03,
 -4.55840321e-02,  2.97509507e-03, -3.96326377e-03,
 -1.05573899e-02,  2.72238609e-02,  2.58107100e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
```

```
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  3.96409228e-03, -3.88929393e-02,
-5.64542397e-02,  1.33513694e-02,  5.52231011e-03,
 4.70168713e-02, -2.57897670e-02, -6.46727498e-03],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  7.07521165e-02,
-2.31685092e-02,  7.31279247e-03, -7.22305154e-03,
-7.87670418e-02,  4.35081925e-03,  1.56829209e-02],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 1.17796251e-01, -3.23530116e-02, -1.56406005e-03,
-1.51675590e-03, -8.26503764e-03,  8.24693511e-03],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  3.04731240e-01,  3.72699111e-02,
-5.70038404e-04,  2.89981090e-04, -1.14721182e-01],
[ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
```

```
         0.00000000e+00,  0.00000000e+00,  6.82877401e-02,
         6.48119704e-03, -6.44254137e-05, -8.88985096e-02],
       [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         2.10076878e-01, -1.59689245e-03, -5.91404556e-03],
       [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  3.33849225e-02,  4.49333335e-03],
       [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00,  1.19110803e-01]]),
'SigmaSqr': 17440.666123806714,
'Lambda': None,
'de_bounds': [[-3, 2], [-3, 2], [-3, 2]]}
```