

# Kriging-1-1D Example

August 14, 2024

## 1 KRIGING METAMODELING: ONE-DIMENSIONAL EXAMPLE

This example showcases how to perform Kriging metamodeling on a simple one-dimensional function using various types of correlation families.

### 1.1 Package imports

```
[1]: from uqpylab import sessions
import matplotlib.pyplot as plt
import numpy as np
```

### 1.2 Start a remote UQCloud session

```
[2]: # Start the session
mySession = sessions.cloud()
# (Optional) Get a convenient handle to the command line interface
uq = mySession.cli
# Reset the session
mySession.reset()
```

Processing .

.

done!

uqpylab.sessions :: INFO        :: This is UQ[py]Lab, version 1.00, running on  
https://uqcloud.ethz.ch.

UQ[py]Lab is free software, published under the  
open source BSD 3-clause license.

To request special permissions, please contact:  
- Stefano Marelli (marelli@ibk.baug.ethz.ch).

A new session  
(d397dbf453394cd9b490a73d561c62f7) started.

uqpylab.sessions :: INFO        :: Reset successful.

### 1.3 Set the random seed for reproducibility

```
[3]: uq.rng(100, 'twister');
```

### 1.4 Computational model

The computational model is a simple analytical function defined by:

$$y(x) = x \sin(x), \quad x \in [0, 15]$$

Specify this model using a string and create a MODEL object:

```
[4]: ModelOpts = {
    'Type': 'Model',
    'ModelFun': 'xsinx.model',
    'isVectorized': 'true'
}

myModel = uq.createModel(ModelOpts)
```

### 1.5 Probabilistic input model

The probabilistic input model consists of a single uniform random variable:

$$X \sim \mathcal{U}(0, 15)$$

Specify its marginal distribution and create a INPUT object:

```
[5]: InputOpts = {
    'Marginals': [
        {
            'Type': 'Uniform',
            'Parameters': [0, 15]
        }
    ]
}

myInput = uq.createInput(InputOpts)
```

### 1.6 Experimental design and model responses

Generate an experimental design  $X$  of size 8 using the latin hypercube sampling (LHS):

```
[6]: X = uq.getSample(N=8, Method='LHS')
```

```
[7]: Y = uq.evalModel(myModel, X)
```

## 1.7 Kriging metamodels

Three different correlation functions of the underlying Gaussian process to create Kriging metamodels are considered.

### 1.7.1 Matérn 5/2 correlation

Select the metamodeling tool and the Kriging module:

```
[8]: MetaOpts = {  
      "Type": "Metamodel",  
      "MetaType": "Kriging"  
    }
```

Use the experimental design and corresponding model responses generated earlier:

```
[9]: MetaOpts["ExpDesign"] = {  
      "X": X.tolist(),  
      "Y": Y.tolist()  
    }
```

Create the Kriging metamodel:

```
[10]: myKrigingMat = uq.createModel(MetaOpts)
```

Note that the various options that have not been explicitly specified have been automatically assigned to their default values. This includes the correlation family which is set to Matérn 5/2 by default.

Print out a report on the resulting Kriging object:

```
[11]: uq.print(myKrigingMat)
```

```
%----- Kriging metamodel -----%  
Object Name:          Model 2  
Input Dimension:      1  
Output Dimension:     1  
  
Experimental Design  
  X size:              [8x1]  
  Y size:              [8x1]  
  Sampling:            User  
  
Trend  
  Type:                ordinary  
  Degree:              0  
  Beta:                [ 2.64467   ]  
  
Gaussian Process (GP)  
  Corr. type:          ellipsoidal
```

```

    Corr. isotropy:      anisotropic
    Corr. family:       matern-5_2
    sigma^2:           4.59087e+02
    Estimation method:  Cross-validation (CV)

Hyperparameters
  theta:               [ 0.85210   ]
  Optim. method:       Hybrid Genetic Alg.

GP Regression
  Mode:                interpolation

Error estimates
  Leave-one-out:       5.40328e-01
%-----%

```

Plot a representation of the mean and the 95% confidence bounds of the Kriging predictor:

```
[12]: uq.display(myKrigingMat);
```

### 1.7.2 Linear correlation

Create another Kriging metamodel with the same configuration options but use a linear correlation family instead:

```
[13]: MetaOpts['Corr'] = {
      "Family": "Linear"
    }
```

```
[14]: myKrigingLin = uq.createModel(MetaOpts)
```

### 1.7.3 Exponential correlation

Finally, create a Kriging metamodel using the exponential correlation family:

```
[15]: MetaOpts['Corr']['Family'] = 'Exponential'
```

```
[16]: myKrigingExp = uq.createModel(MetaOpts)
```

## 1.8 Metamodels validation

Create a validation set of size  $10^3$  over a regular grid:

```
[17]: Xval = uq.getSample(N=1e3, Method='grid')
```

Evaluate the true model responses for the validation set:

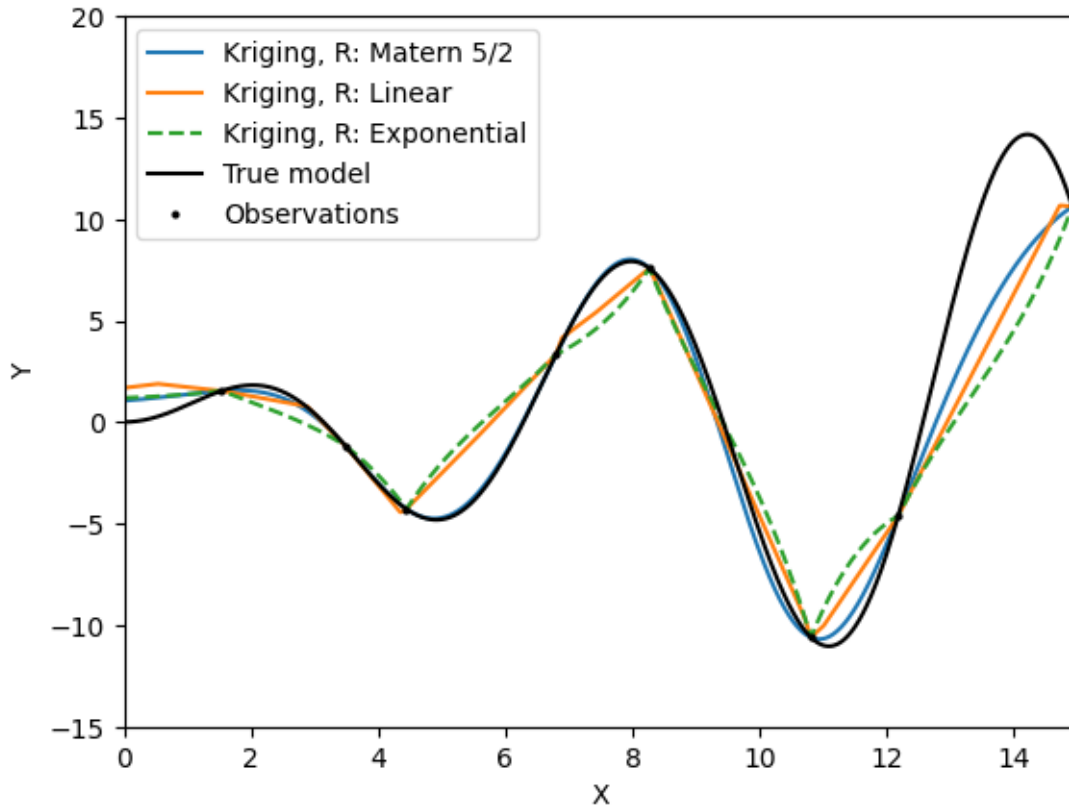
```
[18]: Yval = uq.evalModel(myModel, Xval)
```

Evaluate the Kriging surrogate predictions on the validation set. In each case, both the mean and the variance of the Kriging predictor are calculated:

```
[19]: [YMeanMat, YVarMat] = uq.evalModel(myKrigingMat, Xval, nargout=2)
[YMeanLin, YVarLin] = uq.evalModel(myKrigingLin, Xval, nargout=2)
[YMeanExp, YVarExp] = uq.evalModel(myKrigingExp, Xval, nargout=2)
```

```
[20]: plt.plot(Xval, YMeanMat, '-')
plt.plot(Xval, YMeanLin, '-')
plt.plot(Xval, YMeanExp, '--')
plt.plot(Xval, Yval, '-k')
plt.plot(X, Y, 'ko', markersize=2)

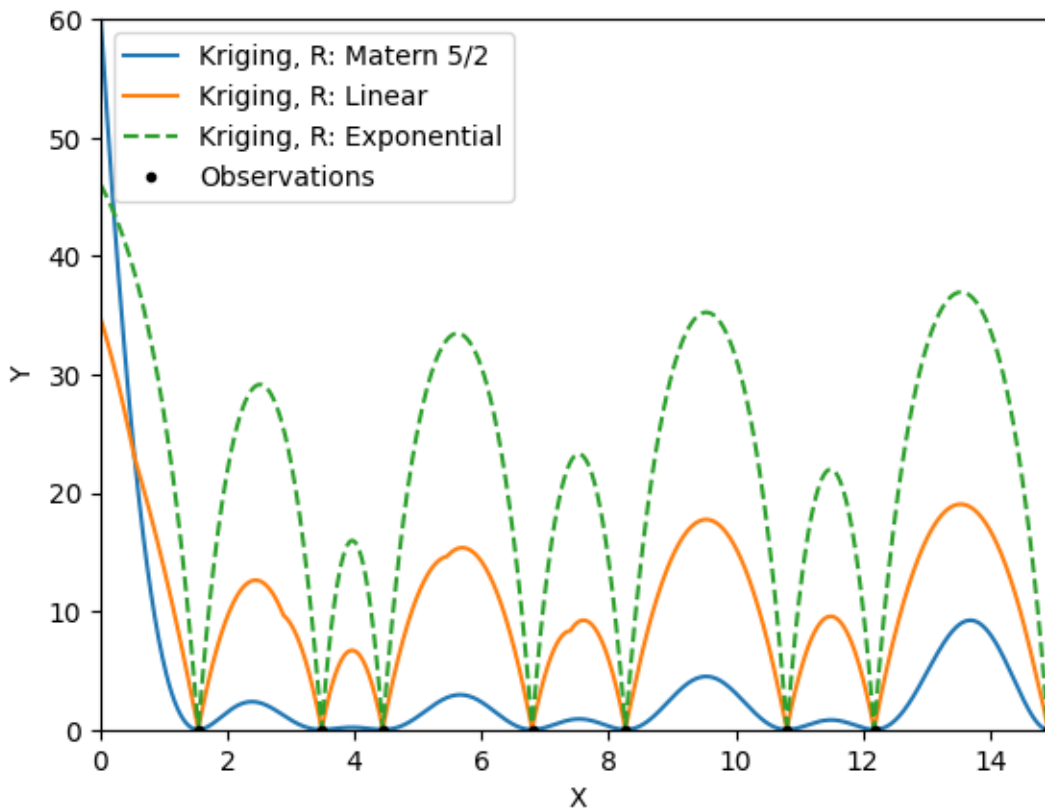
plt.xlim([0, 15])
plt.ylim([-15, 20])
plt.legend(['Kriging, R: Matern 5/2',
           'Kriging, R: Linear',
           'Kriging, R: Exponential',
           'True model', 'Observations'],
          loc='upper left')
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```



Finally, compare the variance that is predicted by each Kriging metamodel:

```
[21]: plt.plot(Xval, YVarMat, '-')
plt.plot(Xval, YVarLin, '-')
plt.plot(Xval, YVarExp, '--')
plt.plot(X, np.zeros(Y.shape), 'ko', markersize=3)

plt.xlim([0, 15])
plt.ylim([0, 60])
plt.legend(['Kriging, R: Matern 5/2',
            'Kriging, R: Linear',
            'Kriging, R: Exponential',
            'Observations'],
            loc='upper left')
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```



## 1.9 Terminate the remote UQCloud session

```
[22]: mySession.quit()
```

```
uqpylab.sessions :: INFO      :: Session d397dbf453394cd9b490a73d561c62f7  
terminated.
```

```
[22]: True
```