# Eigendecomposition of Mercer Kernels using Matlab®

Automatic Control Group
Department of Information Engineering - University of Padova

April 26, 2018 — ver. 1.0

## Contents



## 1 License terms



## 2 Objective

The objective is to compute eigenfunctions and eigenvalues of a generic Mercer Kernel $K$, where

$$K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R} \tag{1}$$

is symmetric, continuous and non-negative definite.

This package consider domains $\mathcal{X}$ that are intervals in $\mathbb{R}$ and $\mathbb{R}^2$. Higher dimensionalities are not yet fully supported.

## 3 Technical requirements

Any Matlab®distribution supporting classes should work, and no additional packages are required. Please use the contacts in Section 8 in case of difficulties.

We assume the reader to be familiar with Matlab®'s coding rules.

# 4 Installation

The source code is managed in classes scoped within a package. Please consult Section 5 to check the necessary `addpath` operations to be performed.

# 5 Detailed description of the input parameters

All the parameters that users need to modify are located in `./MainSoftware/LoadParameters.m`. No modifications are required in other files. In `./MainSoftware/LoadParameters.m` you will find several parameters, divided by scope. Please refer to Section 7 for examples.

## 5.1 Specification of the filepaths

- `addpath` *path of the package (note that* `+EigendecompositionPackage` *is excluded)*
- `tParameters.strResultingMatFilesDirectory` = *directory where .mat results should be placed*
- `tParameters.strResultingTxtFilesDirectory` = *directory where .txt results should be placed*

## 5.2 Specification of the input domain

- `tParameters.cInputDomain` = *cells composed by vectors; each vector contains the input locations of its corresponding dimensionality*

## 5.3 Specification of the measure

- `tParameters.strMeasureKind` = *[ 'exponential' | 'gaussian' | 'uniform' ]*
- `tParameters.cExponentialMeasureParameters` = *needed only if exponential measures are selected. Cells of bidimensional vectors [weight, decay]. The number of cells must be equal to the dimensionality of the domain. Each cell refers to the corresponding dimension*
- `tParameters.cGaussianMeasureParameters` = *needed only if Gaussian measures are selected. Cells of tridimensional vectors [weight, mean, variance]. The number of cells must be equal to the dimensionality of the domain. Each cell refers to the corresponding dimension*

## 5.4 Specification of the kernel

- `tParameters.strKernelKind` = *[ 'gaussian' | 'stablespline' | 'cubicspline' | 'laplacian' ]*
- `tParameters.fGaussianKernelVariance` = *scalar positive value. Needed only if Gaussian kernels are selected*
- `tParameters.fLaplacianKernelScale` = *scalar positive value. Needed only if Laplacian kernels are selected*
- `tParameters.fStableSplineKernelExponentialDecay` = *scalar positive value. Needed only if stable splines kernels are selected*
- `tParameters.bCompute2DKernels` = *[false | true]; skip the computation of 2D kernels if true (due to computational and memory constraints)*
- `tParameters.bCompute3DKernels` = *[false | true]; skip the computation of 3D kernels if true (due to computational and memory constraints)*

### 5.5 Specification of the accuracy of the decomposition

- `tParameters.fPercentageOfVarianceToBeCaptured =` *scalar value in [0,1). Determines how many eigenfunctions will be computed.*
- `tParameters.iMinimalNumberOfEigenfunctionsToBeSaved =` *positive integer. Possibly overrides the previous setting*

### 5.6 Specification of the outputs

- `tParameters.iNumberOfSamplesPerDimensionWhenExportingTxtFiles =`
- integer value, the higher its value the better the accuracy of the .txt files that will be exported
- `tParameters.bPrintDebugInformation =` *[true | false], if one desires the output to be shown in the command window*
- `tParameters.bPlotDebugInformation =` *[true | false], if one desires to see figures about the computations*
- `tParameters.bSaveResultingFiles =` *[true | false], if one wants to save the computed results*
- `tParameters.bEnableMailAlert =` *[true | false], if one wants to receive an email stating that computations are finished*
- `tParameters.strEMailAddress =` *list of email addresses*
- `tParameters.strSmtpServer =` *SMTP server's addres*
- `tParameters.bSendFiguresViaMail =` *[true | false], if with the email one wants to receive also the resulting figures as an attachment*
- `tParameters.bSendMatFilesViaMail =` *[true | false], if with the email one wants to receive also the resulting .mat files as an attachment*

## 6 Detailed description of the outputs

- .txt files containing the tabulated eigenfunctions (1 file per eigenfunction);
- .txt files containing the eigenvalues;
- .txt files containing the measure;
- .txt files containing some examples of realizations obtained from the considered kernel;
- .mat file containing the structure `tKernelParameters` with fields:
  - `strKernelKind`
  - `strMeasureKind`
  - `afEigenvalues`
  - `tHyperparameters`

  Only for 1D kernels it contains also
  - `afDomain`
  - `afMeasure`
  - `aafEigenfunctions`
  - `aafKernel`

Only for 2D kernels it contains also

- `aafDomain`
- `aafMeasure`
- `aaafEigenfunctions`
- `aaaafKernel`

Note: all the .txt files are optimized in order to be plotted using the LaTeX pgfplots package.

# 7 Example of usage

1. change the current directory to `./MainSoftware/`
2. edit `LoadParameters.m` (see the .m file for examples)
3. run `main.m`

# 8 Contacts

For comments, suggestions, indications and bugs reporting please contact `damiano.varagnolo@dei.unipd.it`.