ETH zürich

# High Performance Computing for Science and Engineering II

Spring semester 2022

*P. Koumoutsakos*
*ETH Zentrum, CLT E 13*
*CH-8092 Zürich*

## HW 2 - Korali

Issued: March 14, 2022
Due Date: March 28, 2022, 10:00am

You should be able to find all necessary documents for this homework in the public HPCSEII gitlab. https://gitlab.ethz.ch/hpcse-public-repos/hpcseii-spring-2022-lecture

### Task 1: Install Korali

Korali is a high-performance framework for uncertainty quantification, optimization, and deep reinforcement learning. Its engine provides support for large-scale HPC systems and a multi-language interface compatible with distributed computational models [1]. In this exercise we will be using Korali for gradient free optimization and bayesian uncertaintiy quantifiation. Your first task is to install Korali and get familiar with it.

For visualizing your results with Korali, use the following command:

```
python3 -m korali.plot (--dir <my_results>)
```

If you want to save the output directly as an image, use the following command:

```
python3 -m korali.plot (--dir <my_results>) --output <result>.png
```

If you need more help with Korali, use the official documentation under https://korali.readthedocs.io/en/master/index.html

a) First you have to log in to the Euler supercomputer of ETH. Remember that you need to be connected to the ETH network either directly or via VPN.

   `ssh <username>@euler.ethz.ch`

   Now you are on a login node of Euler. Here you can edit your code, but if you need serious computing power do not run the code on the login node, use the batch system instead. Visit the corresponding scicomp website (https://scicomp.ethz.ch/wiki/Using_the_batch_system) for more information.

b) To compile Korali certain modules are needed. Here a few helpful commands to work with modules:

[1] https://www.cse-lab.ethz.ch/korali/

```
man module //manual for the module command
module --help //shows usage, subcommands and arguments
module purge //removes all modules
module avail //shows all available modules
module load // loads a certain module
module unload // unload a certain module
module list // list all loaded modules
```

Now follow these steps to install the correct modules:

```
source /cluster/apps/local/env2lmod.sh
module purge
module load gcc/8.2.0 python/3.9.9 cmake/3.9.4 openmpi/4.0.2 gsl/2.6
```

For python you also need meson and ninja.

```
pip install meson
pip install ninja
```

c) Use git to download the Korali repository https://github.com/cselab/korali into a folder of your choice. Move to where you would like to install Korali and enter the given command.

```
git clone https://github.com/cselab/korali.git
```

d) Install Korali with the following commands:

```
cd korali
meson setup build --buildtype=release --prefix=$HOME/.local/
meson compile -C build (-j <number_of_threads>)
meson install -C build
```

e) Now add the following line to your bashrc (usually in your home directory: *~/.bashrc*):

```
export PYTHONPATH=$HOME/.local/lib64/python3.9/site-packages:$PYTHONPATH
```

To "reload" your bashrc either log out and log in again or simply use:

```
source ~/.bashrc
```

f) Now everything you need should be installed and you can continue solving the exercise. If everything is working as intended run the following command:

```
python -c "import korali"
```

g) **[5 points]** Run the python scripts in from the examples 1, 2 and 3 with

```
python3 <example>.py
```

The examples are found in the *examples* folder. Show the results with the help of Korali's plotter.

## Task 2: Function Optimization

In this task we are going to use Koralis optimizer to find the minimum of the `egg-crate` function. The function is defined as:

$$x^2 + y^2 + 25(sin(x)^2 + sin(y)^2) \tag{1}$$

We want to obtain the minimum in the following domain:

$$x \in [-5, 5] \tag{2}$$

$$y \in [-5, 5] \tag{3}$$

In figure 1 the surface of this function is shown.

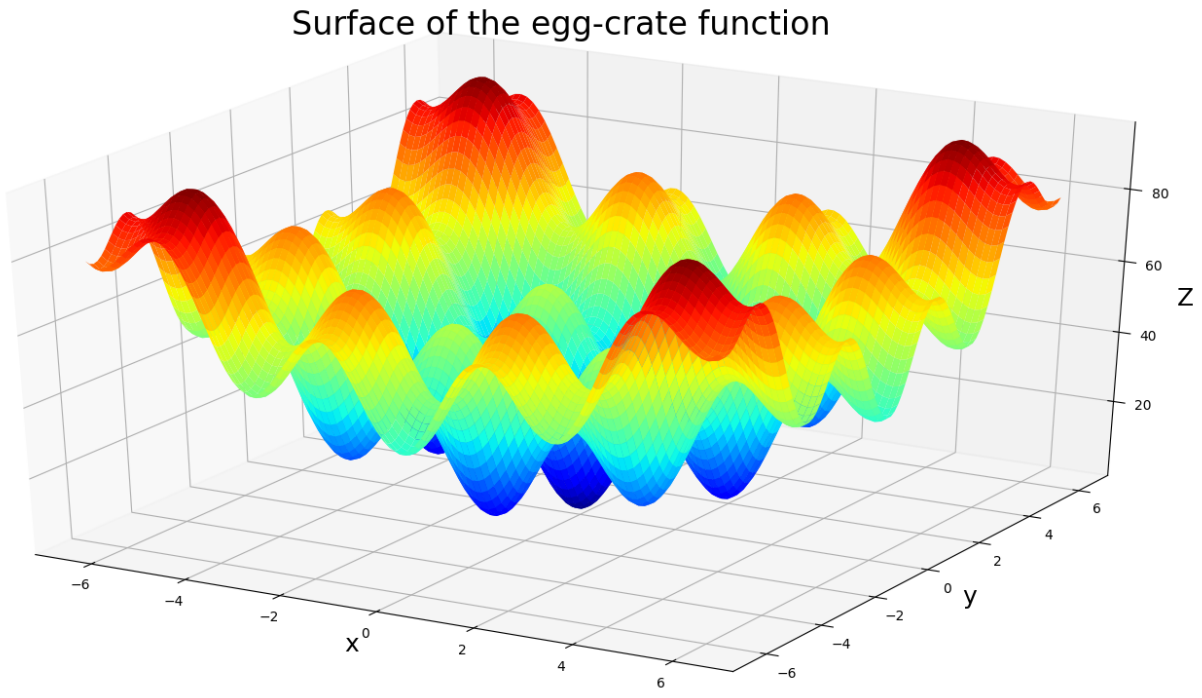


Figure 1: Surface of the `egg-crate` function.

We will perform the optimization using the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) algorithm. Evolution strategies (ES) are stochastic, derivative-free methods for numerical optimization of non-linear or non-convex continuous optimization problems.

The CMA-ES algorithm is implemented in Korali, which you have already installed on Euler, in Task 1. An example of the application of CMA-ES for the optimization of a function with Korali, is provided in *examples/example1*. In this example, the function $g(\theta) = \theta^2$, with $\theta \in [-10, 10]$ is optimized with respect to $\theta$, by finding the value of that minimizes the function within the given interval. Additional examples can be found in Korali's github repository. [2]

[2] https://www.cse-lab.ethz.ch/teaching/hpcse-ii_fs21/

a) **[5 points]**

Implement the model of the `egg-crate` function in *task2/skeleton_code/_model/model.py*. The model works similar to the examples in the *examples* folder, use the comments as hints to complete the code.

b) **[10 points]**

Implement the configuration for Korali in *task2/skeleton_code/cmaes.py*. Once again you can use *examples/example* and the comments as help. Remember that Korali by default maximizes functions when asked to find the optimum.

c) **[15 points]**

After implementing everything, run your code once and answer the following questions:

1. What are the $x_{min}$ and $y_{min}$ coordinates of the minimum?
2. What is the value at $f(x_{min}, y_{min})$
3. Change the number of samples CMA-ES makes. How does the number of samples correlate to the number generations?

## Task 3: Bayesian inference for sensor readout

You are working with a sensor that measures acceleration. The sensor returns a voltage $V$ that depends on the acceleration $a$ through an unknown function $f$.

$$V = f(a) \tag{4}$$

Unfortunately you do not yet know the relationship between acceleration and voltage so you perform some measurements that are shown in figure 2. Then, you decide to model these measurements as follows

$$f(a) = k_1 x + k_3 x^3 + k_5 x^5 \tag{5}$$

Note that your assumed model is an odd function, as negative and positive acceleration should not change the magnitude of measured voltage.

a) **[10 points]**
Write down the formula you would use to estimate the parameters with Bayesian inference, including a normally distributed measurement error $\epsilon$ with zero mean and standard distribution $\sigma$. Explain all components of the bayes formula you used and the variables or vectors they depend upon.

b) **[10 points]**
Implement the parameter optimization using CMA-ES by adapting Example 2. Use the skeleton files in *task3/skeleton_ code/part_ b*.

1. Plot the results using Korali's plotting function.
2. What can you say about the maximum likelyhood of the parameters $k_i$?

c) **[10 points]** Implement the parameter optimization using TMCMC by adapting Example 3. Use the skeleton files in *task3/skeleton_ code/part_ c*.

1. Plot the results using Korali's plotting function.
2. What can you say about the correlation between the parameters $k_i$?

d) **[15 points]** As you observe your experimental equipment you notice that you made an error in the electrical circuit leading to a nonzero offset voltage. Copy your files from Subquestion c) into *task3/skeleton_ code/part_ d* and adapt your model.

1. Plot the results using Korali's plotting function.
2. What can you say about the maximum likelyhood of the parameters?
3. Compare the two models using the evidence with the help of Korali. What can you conclude by comparing the two models.

**Guidelines for reports submissions:**

- Submit a pdf file of your solution via Moodle until April 4, 2022, 10:00am.
- There are 80 available points in this homework. To get a grade of $6/6$ you need to collect 60 points. Collecting $50$ points results in a grade of $5/6$ and so on.

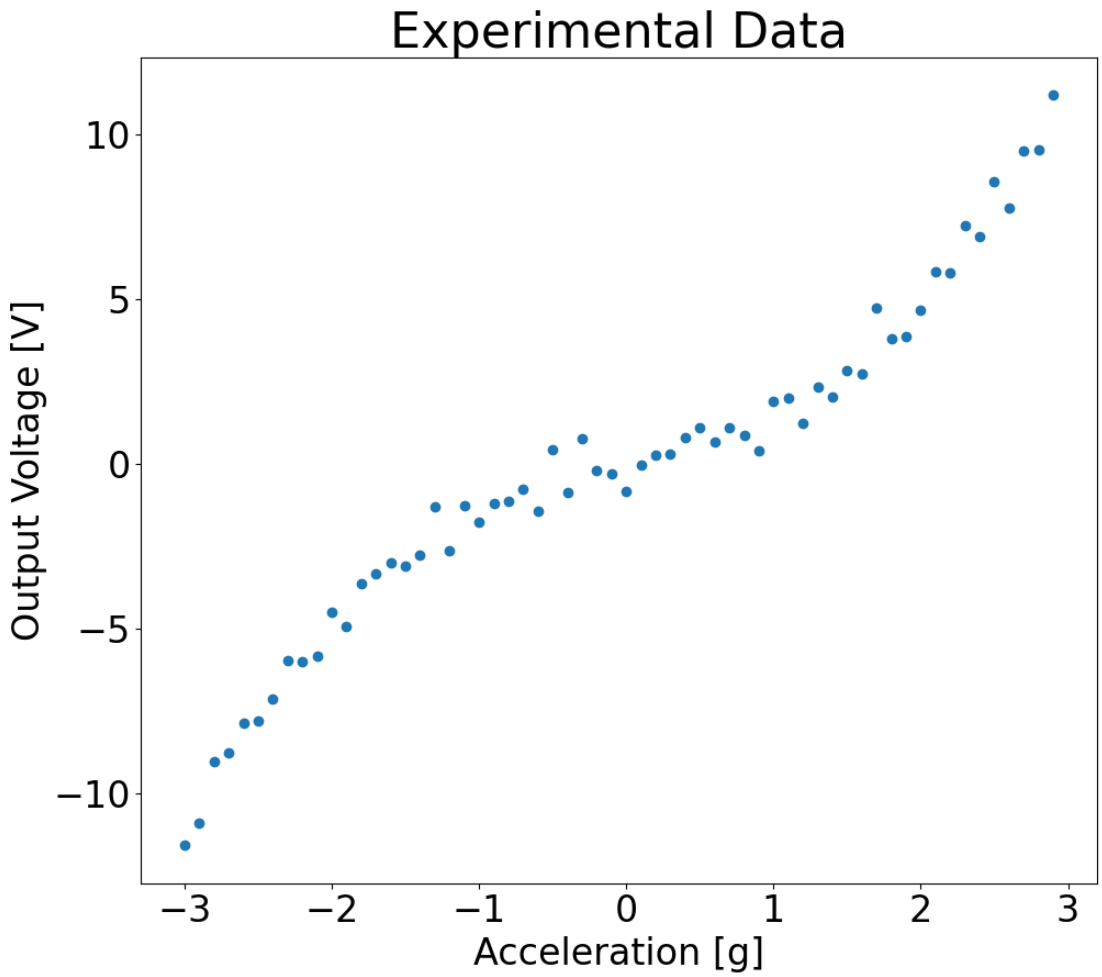


Figure 2: Sample points from your experiment. It shows the output voltage of the sensor depending on the applied acceleration as multiples of the gravitational acceleration on earth. One $g$ is approximately $9.81\,m/s^2$.