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

HW 2 - Applied Sampling with Korali

Issued: March 15, 2021
Due Date: March 29, 2021, 10:00am

1-Week Milestone: Solve tasks 1, 2 and 3 (Part I).

Task 1: Installing korali on Euler

Euler is the computing cluster of ETH Zurich. The cluster works with a queueing system: you submit your program with its parameters (called job) to a queue and wait for it to finish. In this homework you will use `korali`, our new Uncertainty Quantification and Optimization framework developed at ETH Zürich.

a) Your first task is to acquire and install `korali` on the Euler cluster, by following the steps:

1. **Login to Euler**
   Login from within the ETH network[1] on Euler via ssh:
   `$ ssh username@euler.ethz.ch`
   and insert your password when asked to.
   Congratulations! You are now on a login node of the Euler cluster. In this environment you can write code, compile and run small tests. Keep in mind that there are other people working on the same nodes, so be mindful of how you use them!
2. **Acquire korali**
   Clone the source code for `korali` from the software's github repository, using the following command:
   `$ git clone https://github.com/cselab/korali.git`
   This creates a directory named `korali` inside your current working directory. Its contents are those of the repository.
3. **Install korali**
   To build and install Korali, there are a few software requirements. Euler provides the necessary software packages as "modules", which can be loaded and unloaded as needed.

   The basic commands to use the module system are:
   `$ module load <modulename>`
   sets the environment variables related to the specified module.
   `$ module unload <modulename>`
   unsets the environment variables related to `<modulename>`.
   `$ module list`
   lists all the modules currently loaded.

[1] You may use VPN `https://sslvpn.ethz.ch` to connect to the ETH Network from home.

`$ module avail`
outputs a list of all the modules that can be loaded.

To load the modules required to install `korali`, follow the steps:
```
$ source /cluster/apps/local/env2lmod.sh
$ module purge
$ module load gcc/8.2.0 python/3.6.4 cmake/3.9.4 openmpi/4.0.2
```

Additionally, set the value of the environment variable `MPICXX`, required to compile `korali` with MPI support.
```
$ export MPICXX=`which mpicxx`
```

Finally, install `korali`:
```
$ cd korali
$ ./install
```

b) Along with this pdf, you should download the provided resources from the course's gitlab repository. These contain examples and skeleton codes for the 3 coding tasks of the homework.
Contents of the examples:

- `example_1`: Function optimization with CMA-ES (python).
- `example_2`: Bayesian optimization of a model, with respect to a set of reference data (python).
- `example_3`: Bayesian inference of the posterior distribution of a model, with respect to a set of reference data (python).
- `example_4`: Bayesian inference of the posterior distribution of a model, with respect to a set of reference data (c++).

c) Additional notes on using Euler:

1. **Running on Euler:**
Performance measurements and long computations should not be performed on the login nodes but rather they should be submitted to the queue. To submit a job to the queue, you can use the following command from the folder where your program is stored:
`$ bsub -n 24 -W 08:00 -o output_file ./program_name program_args`
This command will submit a job for your executable `program_name` with arguments `program_args` by requesting 24 cores from a single node and a wall-clock time of 8 hours, after which, if the job is not already finished running, it will be terminated. The report of the job, along with the information that would usually appear on the terminal, will be appended in the file `output_file`, in the folder from where the job started.

   You can also allocate an interactive job for continuous development:
   `$ bsub -n 1 -W 01:00 -Is bash`

   While your job is running you can always use the following command to get the status and IDs of your jobs:
   `$ bjobs`

In order to terminate a job you can use the command:
`$ bkill <jobid>`

2. **I/O performance and `$SCRATCH`:**
   Since your simulations might involve a lot of I/O (input/output), you must never run your simulations in your `$HOME` directory, but setup the runs in your `$SCRATCH` space. The disks associated with this space are designed for heavy loads[2]. Lastly, your quota in `$HOME` is much smaller compared to `$SCRATCH`. However, please note that the memory in `$SCRATCH` is temporary and **any files older than 15 days are deleted automatically** (see `$SCRATCH/__USAGE_RULES__`). Follow the links below for more information on the Euler cluster. Use the following command to change to your scratch directory:
   `$ cd $SCRATCH`

   Additional information on the Euler cluster, its instruments and on how to use it can be found at:
   - https://scicomp.ethz.ch/wiki/Euler
   - https://scicomp.ethz.ch/wiki/User_documentation

[2]However, `$SCRATCH` is not designed for frequent storing. If you are logging temporary results into a file, open the file once at the beginning, and do not flush e.g. more than once per second (or per minute). Related to it, note that `std::endl` not only prints the newline character `'\n'`, but also flushes the stream.

Task 2: Optimization: The Rosenbrock function (30 Points)

In this task we are interested to optimize the two dimensional Rosenbrock function, by finding the location in space where the function takes its minimum value. The Rosenbrock function, shown in Figure 1, is a non-convex function commonly used as a benchmark problem for optimization algorithms. It is defined as:

$$f(x, y) = (a - x)^2 + b(y - x^2)^2 \tag{1}$$

with a and b being the parameters that define the "shape" of the function. The global minimum is located at $(x, y) = (a, a^2)$ and corresponds to $f(x, y) = 0$.

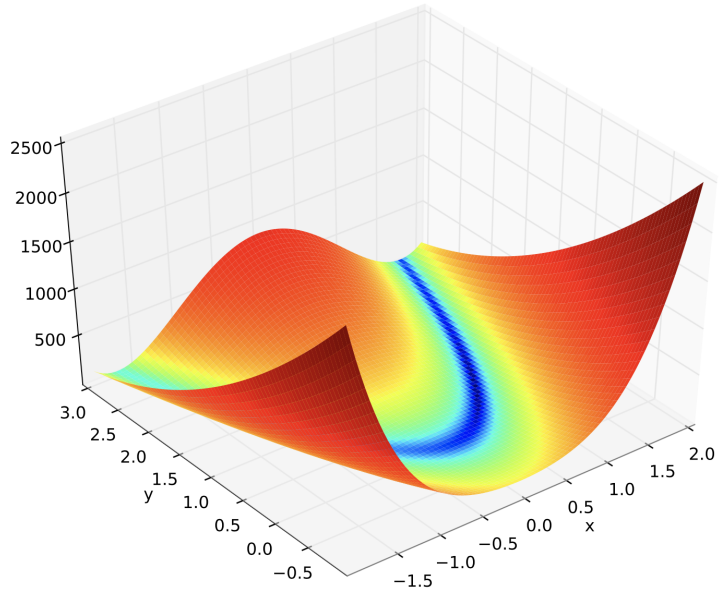


Figure 1: Plot of the two-dimensional Rosenbrock function, with $a = 1$, $b = 100$. For this combination of a and b, the function is minimized at $(1, 1)$. Source: https://commons.wikimedia.org/wiki/File:Rosenbrock_function.svg.

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. For an explanation of the CMA-ES method, you can refer to the recording and slides of the tutorial given during the exercise session of March 15^{th}, 2021.

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/example_1`. In this example, the function $g(\vartheta) = \vartheta^2$, with $\vartheta \in [-10, 10]$ is optimized with respect to ϑ, by finding the value of ϑ that minimizes the function $g(\vartheta)$ within the given interval. Additional examples can be found in korali's github repository.

The optimization can be run as follows:

```
$ python3 ./run_cmaes
```

By default, the results are saved in the folder `_korali_result`. To change the folder containing the results, you can add the following lines to your code, before calling `k.run(e)`:

```
e["File Output"]["Enabled"] = True
e["File Output"]["Enabled"] = '<my_results>'
```

where `<my_results>` is the name of the output folder.

For a visualization of the results from CMA-ES you can use the command:

```
$ python3 -m korali.plotter --dir <my_results>
```

Details on the visualized quantities can be found in korali's documentation.

QUESTIONS:

In this task you will estimate the coordinates (x_0, y_0) at which the Rosenbrock function with shape parameters $a = 1$ and $b = 100$, is minimized.

a) (10 points) Describe the model, solver, and parameters you will use to represent this problem in `korali`.

b) (10 points) You are provided with a skeleton code. Fill-in the locations marked with `TODO`:

- Implement the Rosenbrock function in: `task1/skeleton_code/model/model.py`. Similarly to `examples/example_1`, the model should obtain the values of the parameters x and y from `p["Parameters"]`, use them to compute the value of the Rosenbrock function, and store the appropriate value in `p["F(x)"]`.
- Complete the `korali` configuration in: `task1/skeleton_code/run_cmaes.py`.

c) (5 points) What are the coordinates (x_0, y_0) corresponding to the minimum value of the function? How did you obtain these values?

d) (5 points) How does the number of samples used in CMA-ES affect the number of generations required for the algorithm to converge?

Task 3: Bayesian inference for a Red Blood Cell model (80 Points)

Blood transport is a fundamental process in the human body. The transport properties and rheology of blood are primarily governed by the dynamical motion of Red Blood Cells (RBCs), which are highly flexible objects with a biconcave resting shape. In turn, the flexibility of RBCs is controlled by their visco-elastic properties. It is therefore crucial that computational models of RBCs, model the mechanics and dynamics of RBCs accurately. This can be achieved by calibrating the model's parameters with respect to experimental data.

A popular experiment for the calibration of the elastic properties of RBC models, is the stretching experiment. In this experiment, two beads are attached on opposing sides of a RBC, and are pulled in opposite directions with a constant extensional force (see Figure 2 (left)). During this process, the RBC is elongated along the axis of the stretching force, until the elastic resistance of the RBC balances the stretching force. At that point, the elongation of the RBC is measured. The process is repeated for multiple stretching forces, and data such as the ones shown in Figure 2 (right) are generated.

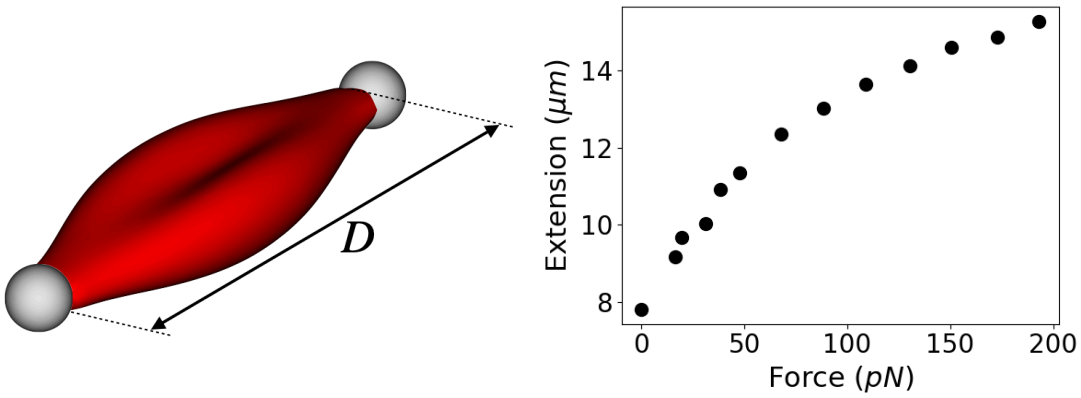


Figure 2: RBC elongation with respect to the extensional force. The data (right) are digitized from the experimental study of [1].

By analyzing the experimental data, you have realized that the stretching extensions can be described by an analytical function of the form:

$$D = D_0 + \sum_{p=1}^{M} k_p \, x^p \tag{2}$$

where D is the RBC extension along the force direction, D_0 is the RBC diameter at rest (i.e. at zero stretching force), k_i are a set of coefficients, and x is the stretching force.

QUESTIONS:

Part I: Bayesian Optimization.

a) (10 points) As a first step, you have decided to use $M = 2$ in Equation (2), therefore in this case: $D = D_0 + k_1 x + k_2 x^2$. You would now like to find the values of D_0, k_1 and k_2 with the highest probability of predicting the experimental data, using Bayesian optimization. In order

to proceed, you need to construct an error model: a relation between the experimental data, the computational model, and any modeling/experimental errors. Assuming that the error is normally distributed with zero mean and unknown variance, write down the functional form of the error model, and explain all terms and symbols used.

b) (10 points) After defining the error model, you want to find the parameter set that maximizes the likelihood function, $p(\vartheta|d)$. What are the components of the parameter vector ϑ?

c) (10 points) You now proceed to the Bayesian optimization with CMA-ES. Using the provided skeleton code, adapt the `korali` configuration from `examples/example_2` to obtain the maximum likelihood estimate for D_0, k_1, k_2 and σ^2, by sampling withing the region: $4 \leq D_0 \leq 12$, $-0.5 \leq k_1 \leq 0.5$, $-0.1 \leq k_2 \leq 0.1$ and $0 \leq \sigma^2 \leq 5$. Be careful to define reasonable initial values and initial standard deviations.

d) (10 points) Using `korali`'s plotter, provide a plot of your optimization results. What is the maximum likelihood estimate for each parameter? Check that your results have converged with respect to the number of samples.

Part II: Bayesian Inference & Model Selection.

e) (10 points) Using TMCMC and the provided skeleton code in `task3/skeleton_code/part_II`, adapt the example from `examples/example_3`, to sample the posterior distribution of your parameter set ϑ.

f) (10 points) Plot the resulting posterior distribution using `korali`'s TMCMC plotter. Instructions and details on the visualized quantities can be found in the this link. What can you say about any correlation between the parameters D_0, k_1 and k_2?

g) (20 points) You begin to wonder whether the analytical model M_2 that you chose earlier, $D = D_0 + k_1 x + k_2 x^2$, has enough flexibility to explain the experimental data, or whether you should use a model M_3, with higher order terms, such as $D = D_0 + k_1 x + k_2 x^2 + k_3 x^3$.

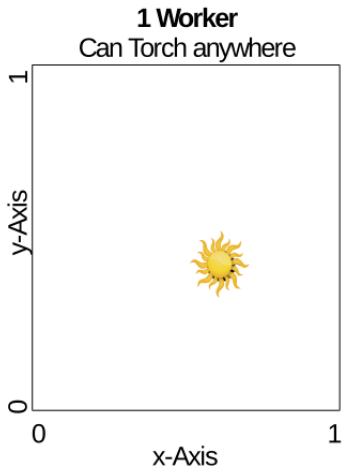
- Which of the two models has the highest probability to explain the data? Extend the code from sub-question (e) to answer this question.
- What metric(s) did you use to verify your answer?

Task 4: The Candle Problem. (40 Points)

As the newly appointed head of operations in an up-and-coming electric car startup company, it is your job to ensure that all manufacturing processes are as efficient as possible. After all, the success of the company depends on the quality of their cars.

Not long after you take your post, you receive concerning news from the technical director of the chassis department. Apparently, cars are leaving the factory with inconsistent build quality. Some of them never report a failure, while others suffer from irreparable structural damage.

"The problem is the metal sheets we are using to build the chassis. Some of them seem to have a better heat treatment that make them resistant, while others turn out brittle". After a visit to the metal sheet manufacturing section, you find out that the metal treatment consists of heating one section of the 2D metal sheet using a torch (or candle).



Unfortunately, the current treatment process does not keep track of the exact position where the worker applies the torch. We have, however, an automatic machine that registers the temperature at multiple points of each sheet, just after they leave the treatment facility.

You receive a new report indicating that Sheet #004392 seems to have successfully passed all structural integrity tests, with outstanding results over all other sheets. Your technical director proposes: "We should replicate the same heat treatment that this sheet received to all the new sheets from now on. That would solve our problem.".

a) (20 Points) Armed with the temperature measurements for Sheet #004392 (file: *data.in*; containing x- and y-position as well as the temperature of 50 points on the sheet), use Korali to indicate:

   - Describe the model, solver, and parameters you will use to represent this problem in Korali.
   - What is the most likely (x, y) position where the torch was applied? Adapt the example in `examples/example_4` to answer this question.
   - What algorithm did you use to answer the previous sub-question and why?

   The model given in *task4/skeleton_codes/_model* solves the heat equation on the metal sheet, given the necessary parameters (position of the candle) and returns the resulting temperature at the measurement points from *data.in*.

b) (20 Points) Thanks to your assessment, treating metal sheets with the new procedure has greatly improved the quality of the cars chassis. However, none of the new sheets seems to match the exceptional quality obtained by sheet #004392. It seems like there is some piece of information missing. Intrigued, you approach the metal workers in search for more clues. The workers report that: "We seldom torched the sheets close to the edge. Rather, we try to stay more or less close to the center of our section, horizontally speaking.". After analyzing their report, you realize that the x-axis position of their torch has a mean of 50cm, with a 5cm standard deviation. No additional information about their position in the y-axis. Answer:

- What would you change in your model to better reflect this new information?
- What is the new recommended values for the position of the torch?
- Use Korali to compare the *Evidence* with and without the prior information presented by the workers. What conclusion can you make from this comparison?

Guidelines for reports submissions:

- Submit a zip file of your solution via Moodle until March 29, 2021, 10:00am.

References

[1] JP Mills, L Qie, M Dao, CT Lim, and S Suresh. Nonlinear elastic and viscoelastic deformation of the human red blood cell with optical tweezers. *Molecular & Cellular Biomechanics*, 1(3):169, 2004.