High Performance Computing for Science and Engineering II

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

Spring semester 2019

HW 3 (Part 2 of 2): Applied Sampling with Korali

Issued: March 25, 2019
Due Date: April 1, 2019 10:00am

Warm-Up & Information

This homework requires the use of *Korali*, our new Uncertainty Quantification and Optimization framework developed at ETH Zürich. Along with this pdf, you should download the provided code which contains the following folders:

- *korali/* contains a ready-to-use, pre-compiled version of Korali. The library is compressed to save space, uncompress it with `tar -xvf libkorali.tar.gz`. The library and some of the task models will only work on Euler (or compatible ETH systems).
- *example/* contains examples based on the Ackley function. These examples contain all the Korali use cases required to solve this homework.
- *plotting/* Contains Python and MATLAB codes required to plot the distribution plots based on the output of Korali's TMCMC engine.
- *task1/* Contains the skeleton codes and model for Task 1.
- *task2/* Contains the skeleton codes and model for Task 2.

If you work on Euler you can load the same modules used for homework 1: `module purge && module load new gcc/6.3.0 intel/2018.1 mvapich2/2.3rc1`.

Guidelines for Moodle submission:

- For HW3 Part 1: Two solution codes (recommended: in python).
- For HW3 Part 2: 5 solution codes: task1a.cpp task1c.cpp task1d.cpp task2a.cpp task2b.cpp, containing your Korali-based solutions.
- The final version of your report PDF which include the solutions to both parts of Homework 3. Do not forget to answer every question in the HWs.

Example: The Ackley Function.

The *Ackley* function is a well-known test case for optimizer algortithms. It contains many local minima in which an optimizer can get 'stuck', but only one global maximum. In the *example/* folder we provide 6 use case for Korali. Feel free to take a look at them and use them as basis for your solution codes.

Case 1: We want to determine the values of the (x_1, x_2) parameters that represent the global minimum for the 2D Ackley function. The code in *minimize_function.cpp*, uses Korali's CMAES engine to find these values. Since CMAES maximizes $f(x)$, we use *maximize_ackley()* as model for Korali, which returns *-Ackley()*, instead. Note: In this case, we are given the values for (a, b, c) which define the 'shape' of the Ackley function, so these are not parameters to infer.

Case 2: The code in *sample_function.cpp*, uses Korali's TMCMC to generate tmcmc.txt, which will give a better idea of the distribution of Ackley around its global minimum, along with the marginal distributions of the parameters. Use **plotmatrix_hist** to plot the distribution graph.

Case 3: In this case, we do not know anything about the values of the a, b and c parameters that shape the Ackley. Instead, we are given a list of measurements with the (x_1, x_2) position and their $f(x_1, x_2)$. We need to use these measurements to infer the values for a, b and c. The code in *maximize_likelihood.cpp* will load this information from *data.in* and use it as reference data for Korali to use. The model, *likelihood_ackley*, will store the values of $f(x_1, x_2)$ for every point in the list, given a, b, c samples. This code will produce the values of these parameters that maximize their likelihood given the data.

Case 4: The code in *sample_likelihood.cpp*, uses Korali's TMCMC to sample and give a better understanding of the distribution of the parameters that give the maximum likelihood.

Case 5: In this case, we calculate the posterior distribution of the (a, b, c) parameters, given some prior information. We are told that parameter a follows a Gamma distribution, while parameter *c* follows a Gaussian distribution. No prior information given for b. The code in *maximize_posterior.cpp* uses Korali CMAES to find the parameters that maximize the posterior for (*a,b,c*).

Case 6: The code in *sample_posterior.cpp*, uses Korali's TMCMC to sample and give a better understanding of the distribution of the parameters that give the maximum posterior.

Task 1: Herr Küheli's Farm. (40 Points)

Herr Küheli's dairy farm is located deep within the hills of the Appenzell canton. His farm, famous for its delicious milk and cheeses, is home to hundreds of cows that freely graze around its grass fields and slopes. Once a year, during the month of September, the farm is visited by the Swiss Federal Food Safety and Veterinary authorities to ensure the good health and proper treatment of the cattle. Herr Küheli takes great pride in his cows, and eagerly awaits the opportunity for quality certification.

Figure 1: A painting of Appenzell by Ulrich Martinelli Druck

During the last few years, however, certification visits have been problematic. Given the vast square territory ($5\text{km} \times 5\text{km}$) enclosed by the Farm, the task of finding the herd has proven to be quite a challenge, taking hours to locate the majority of the cows. Authorities have warned Herr Küheli that such delays should not repeat during their next visit. This year, they have given him a maximum of one hour to locate at least 150 cows for examination. If he fails to achieve this goal, he may lose his certification.

To ensure a successful visit, Herr Küheli has hired you, an expert in optimization and sampling methods (and skilled Korali user) to help him find a way to quickly locate his cows. Your job is to determine the best spot in the farm to look for cows. He will personally go to each spot you suggest and let you know how many cows he has seen (given his old age, his sight is limited to a 200m^2 radius). Each trip takes him 5 minutes. Knowing that you only have a limited number of trials, you inquire about cow grazing behavior and their possible location. He gives you a hunch: "Cows will always graze where grass grows tall. They must surely be all around the spot of the farm where grass grows taller!".

Upon consultation with local agricultural engineers, you find that they have a model for grass growth at the local area. The model considers the capricious topography of the farm and outputs the estimated height of grass given a (x,y) coordinate pair. To produce correct estimations, the model requires two input parameters: *ph* – the current pH acidity of the ground, and; *mm*, the amount of rain fallen in the last month, measured in millimeters. Fortunately, the exact values for this parameters are well known: "Last month we had 80mm of rain and the soil under the farm currently has a pH of 6.0"

a) (10 Points) Use Korali to find the X and Y coordinates of the spot where the grass grows taller. Send Herr Küheli to check for cows to that spot using the *$check_cows* command. Write down the result of this first attempt along Herr Küheli's feedback.

   Syntax: $check_cows X Y, the coordinates of the farm to check, in km, from 0.0 to 5.0.

   The procedure in this task is the same you encountered in **Case 1** of the example: you have to find the global maximum of the grass height. In order to do so, you should use Korali's CMAES engine to maximise the provided, but unknown, grass height model. You can see the coded solution in the provided cpp file. The output of your task1a code should look something like the following:

```
⋮
[Korali] Gen 67 - Elapsed Time: 0.000083, Fitness Change: 6.96e-14
[Korali] Finished - Reason: Function value changes (6.96e-14) < (1.00e-13)
[Korali] Parameter 'x1' Value: 4.085200
[Korali] Parameter 'x2' Value: 3.747600
[Korali] Total Elapsed Time: 0.004440s
```

   with Korali finding the maximum grass height at (4.085200, 3.747600). If you send Herr Küheli to this point you will receive the following feedback:

```
./check_cows 4.085200 3.747600
Searching for cows near (4.085200, 3.747600)...
New Cow Found! Id:#95, Position: (4.153557, 3.972554), Distance: 0.235111
⋮
New Cow Found! Id:#449, Position: (4.097446, 3.971806), Distance: 0.224540
New cows found: 9
Total cows found so far: 9
Herr Kueheli says: "I knew we should look around the spot with the tallest grass!."
Herr Kueheli says: "The rest of the cows should be around here. Lets try these nearby
    points:"
[4.08, 3.95]
[4.28, 3.75]
[3.88, 3.75]
[4.08, 3.55]
>>> Time until deadline: 55 minutes. <<<
```

b) (10 Points) Seems like you have found a bunch of cows in your first attempt. Confident you are on your way to success, you decide to follow Herr Küheli's suggestion as to where to look for the rest of the cows. Write in your report the location of these spots and try them. Answer in your report: why do you think they were suggested? Did you achieve the results you were expecting?

Proceeding the search with the four points suggested by Herr Küheli:

```
./check_cows 4.08 3.95
Searching for cows near (4.080000, 3.950000)...
New Cow Found! Id:#20, Position: (4.274126, 4.013635), Distance: 0.204289
:
New Cow Found! Id:#495, Position: (4.171655, 4.122416), Distance: 0.195264
New cows found: 15
Total cows found so far: 24
Herr Kueheli says: "That was good, but we need to find them faster!."
>>> Time until deadline: 50 minutes. <<<

./check_cows 4.28 3.75
Searching for cows near (4.280000, 3.750000)...
New cows found: 0
Total cows found so far: 24
Herr Kueheli says: "That was not good enough, perhaps this strategy does not really
   work."
>>> Time until deadline: 45 minutes. <<<

./check_cows 3.88 3.75
Searching for cows near (3.880000, 3.750000)...
New Cow Found! Id:#73, Position: (3.890320, 3.468693), Distance: 0.281496
New cows found: 1
Total cows found so far: 25
Herr Kueheli says: "That was not good enough, perhaps this strategy does not really
   work."
>>> Time until deadline: 40 minutes. <<<

./check_cows 4.08 3.55
Searching for cows near (4.080000, 3.550000)...
New cows found: 0
Total cows found so far: 25
Herr Kueheli says: "That was not good enough, perhaps this strategy does not really
   work."
>>> Time until deadline: 35 minutes. <<<
```

you only manage to find a total of 25 cows, not enough for the Swiss Federal Food Safety and Veterinary authorities. Probably Herr Küheli suggested these four neighbouring points because he expected the grass height to change only gradually and the number of cows to be correlated to this variable. Evidently this is not the case: either the grass height is very irregular on his farm or the cows do not actually spend most of their time where the grass is higher. The maximum computed in a) provides us with information regarding only one point, but no knowledge about its surroundings. In order to get an objective idea of the grass height you proceed as instructed in point c).

c) (10 Points) Use Korali's sampling engine to gain a better understanding of grass growth at the Farm. What new points would you instruct Herr Küheli to look at? Explain your decision. Were you able to find all 150 cows within an hour? Explain why the initial strategy was not effective and why your current strategy provides better results.

The procedure is now that of **Case 2** of the example: you have to sample the model for the grass height. In order to do so, you should use Korali's TMCMC engine. You can see the coded solution in the provided cpp file. The output of your task1c code should look something like the folliwing:

```
[Korali] Starting TMCMC. Parameters: 2, Seed: 0x5C9CDE4E
[Korali] Generation 0 - Time: 0.012065s, Annealing: 0.00%, Acceptance: 100.00%
[Korali] Generation 1 - Time: 0.002635s, Annealing: 0.68%, Acceptance: 45.68%
[Korali] Generation 2 - Time: 0.002219s, Annealing: 5.16%, Acceptance: 46.00%
[Korali] Generation 3 - Time: 0.002227s, Annealing: 28.18%, Acceptance: 46.63%
[Korali] Finished. Evidence: -3.302056.
[Korali] Total Time: 0.131211s - Sampling Time: 0.019146s - Engine Time: 0.112000s.
[Korali] Saving results to file: tmcmc.txt.
```

and plotting the results with the provided script you will observe a picture similar to the following (remember that the distribution of the initial samples is random):

a) b)

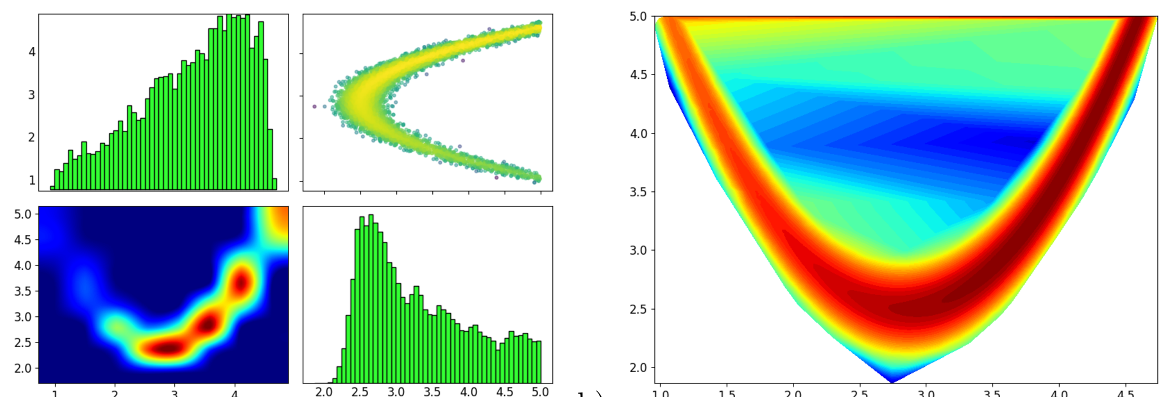


Figure 2: TMCMC samples on Herr Küheli's farm. a) correlation between x and y positions in the farm. b) grass height at the location of the samples provided by TMCMC (same as the top-right panel in a).

It would seem that the point where the grass is taller is not surrounded by a lot of other tall grass, and therefore not many other cows. An area that seems more promising is around (2.9, 2.5). Instructing Herr Küheli to search for cows at this point you find an additional 96 cows, bringing your total to 121.

```
./check_cows 2.9 2.5
Searching for cows near (2.900000, 2.500000)...
New Cow Found! Id:#5, Position: (3.075143, 2.435791), Distance: 0.186542
:
New Cow Found! Id:#477, Position: (3.056540, 2.460539), Distance: 0.161438
New cows found: 96
Total cows found so far: 121
Herr Kueheli says: "Great, this strategy is really effective!"
>>> Time until deadline: 30 minutes. <<<
```

Finally, moving to any other point with tall grass, e.g. (3.6, 2.75), you should be able to complete the task with time to spare. Good job!

```
Searching for cows near (3.600000, 2.750000)...
New Cow Found! Id:#0, Position: (3.484399, 2.730184), Distance: 0.117287
:
New Cow Found! Id:#490, Position: (3.690406, 2.615545), Distance: 0.162023
New cows found: 69
Total cows found so far: 190
Herr Kueheli says: "Thanks so much, you and Korali helped me find all my cows in time!"
```

Evidently, differently from what Herr Küheli thought, the cows do not simply go where the grass is taller, but where they can find a higher volume of grass (obviously you may say in hindsight). You can deduce this by the distribution of samples provided by the TMCMC analysis in figure 2a: you would find an additional 63 cows at (2.4, 2.6).

d) (10 Points) Frau Kleineblume is planning to build a pumpkin garden right besides Herr Küheli's farm. The optimal range of soil pH for pumpkins to grow is within $5.5 - 7.5$. Additionally, these plants require constant irrigation, and therefore can only thrive under a rain volume of $> 100mm$ per month. Unfortunately, none of the locals know the value of these parameters for the current month of March, not even Herr Küheli.

Local experts are able to give you some clues: "The pH of the soil never goes below 4 or above 9. Also, historical trends indicate that at this point of the year, rain volume averages 90mm with a 20mm standard deviation"

Once again, you are summoned to aid the locals with their endeavors. Armed with the knowledge you acquired on HPCSEII, you ask Herr Küheli to inform you about the height of the grass at many spots in his farm as possible. He assembles this information into a file called *grass.in*.

Use Korali to infer the likely values for the pH of the soil and rain volume. Write the rationale and steps you took to obtain these values. How did you integrate the expert's clues in your calculations? What can you say about the uncertainty of your results? Would you FULLY recommend (or not recommend) Frau Kleineblume to plant her pumpkins at this time of the year? Explain your decision.

For this problem the rationale is that of **Case 4** in the example: you have some sparse measurements of the height of the grass and you need to infer the most likely values of pH and rainfall by solving an inverse problem. In order to do this you can use Korali to solve a likelyhood problem with TMCMC for sampling. Given that we only know the upper and lower bounds for the pH of the soil, and do not have any information about a possible distribution, the most logical choice is to use a uniform distribution for this parameter. Regarding the volume of rainfall, instead, we can assume a Gaussian distribution with given mean and standard deviation. Solving the problem and plotting the results will produce a picture similar to the following:

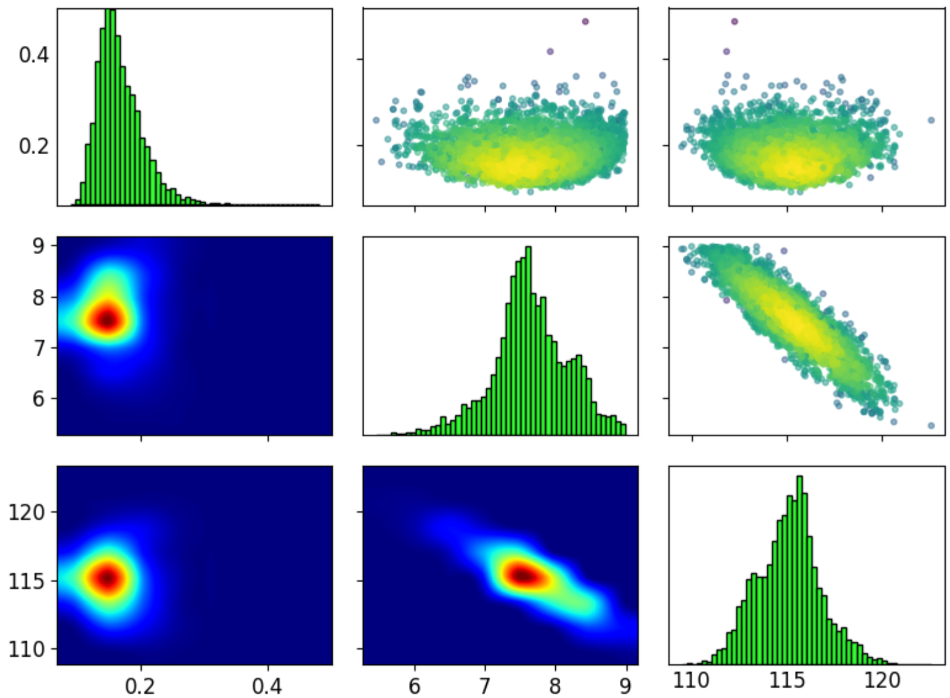


Figure 3: Likelyhood of pH and rainfall in Frau Kleineblume problem.

where the first column represents the estimation error, σ, the central column corresponds to the values of pH, and the rightmost column to the volume of rainfall in millimiters. You can see that the average rainfall is around 115mm, which is more than the necessary 100 even considering the estimation error, and that it has zero probability of falling below 100mm. The mean pH, instead, is around the upper limit of 7.5. By manually loading the `tmcmc.txt` file in either python or matlab, you can see that the mean is actually around 7.64. Moreover, at least for the analysis in figure 3 which was performed with $10\,000$ points, only 38% of the samples are actually below 7.5, which means that there is a higher chance that the pH level is higher than the upper limit for the pumpkins. For this reason you should not recommend Frau Kleineblume to plant her pumpkins at this time of the year.

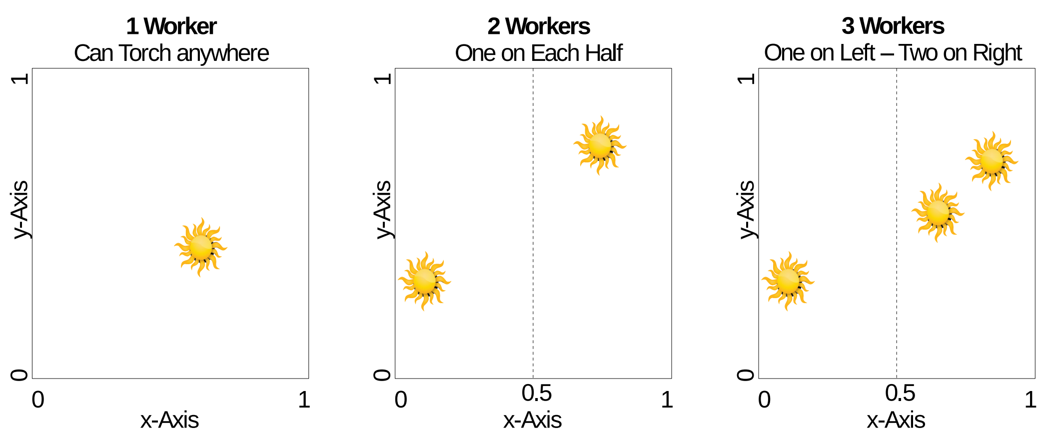
Task 2: n-Candle Problem. (40 Points)

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 it is severely understaffed. Indeed, they have only 3 employees to apply heat treatment to all the metal sheets used in the factory.

Metal treatment consists of heating one or more sections of the 2D metal sheet using a torch (or candle). When demand is very high, each worker will treat different metal sheets by applying a torch anywhere in the plate. When there's average demand, two employees may collaborate on treating the same plate, by each working on their half of the plate (as divided by the x-axis). Finally, when demand is slow, all three employees will work together on each metal sheet, with one employee working on the first half, and two on the second half. As a result, there are 3 possible configurations by which a metal sheet could be treated:



Unfortunately, the current treatment process does not keep track of how many workers treated each sheet, nor the exact position where they applied their 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*), use Korali to indicate:

   - Describe the model, solver, and parameters you will use to represent this problem in Korali. (Hint: the *p.nCandles* variable lets you choose which model to use, either 1-Candle, 2-Candles, or 3-Candles.)
   - What is the most likely number of workers employed in the treatment of this sheet?
   - What metrics did you use to verify which model fits the data better?
   - What is the most likely (x, y) position each of them applied their torches.

An example solution for the implementations in file *hw3b/solution_codes/task2/task2a.cpp*. The experiments have been carried out with the shell script *run_task2a.sh*. Please make sure that you loaded the necessary modules.

- The goal of this task is to evaluate the evidence of our data for each model and consequently select the model. The models describe the solution of the heat equation with 1, 2 or 3 heat sources. In order to sample the posterior distribution of the parameters and calculate the evidence of the models we set up a problem of type *Korali::Problem::Posterior* and we run it with the solver *Korali::Solver::TMCMC*. To correctly set up the problem we have to define a prior distribution for the x and y coordinate of the candles (*Korali::Parameter::Uniform*). The prior of the x coordinate of the first candle is bound at $x_{low} = 0.0$ and $x_{up} = 1.0$, respectively at $x_{low} = 0.0$ and $x_{up} = 0.5$ for the configuration with 2 or 3 candles. The priors of the x coordinate of the second and third candle is bound at $x_{low} = 0.5$ and $x_{up} = 1.0$.
- Comparing the evidence of the three models yields that the most likely number of workers is 3.
- Bayesian model selection is based on the evidence of the three models. Model evidence accounts for both the accuracy and the complexity of the model. Since TMCMC is a randomized algorithm the results strongly depend on the

configuration of the internal variables. In order to get a good estimator of the evidence of the models we carefully increase the population size of TMCMC and execute repeated runs. With 100'000 samples and 10 repeated runs we can clearly discriminate the three models models and report following mean logevidence (standard deviation in brackets):

1. -117 (2)
2. -115 (17)
3. -82 (11)

Note that there are other tuning parameters in TMCMC that improve the sampling quality, e.g. reducing TolCOV which mostly leads to smaller increments of the annealing parameter ρ and hence to more generations; or searching the optimal covariance scaling parameter (increasing the sample size is the most straight forward approach to improve the sampling quality but also the most costly computationally wise).

- In order to find the most likely coordinates of the three candles we maximize the posterior with the CMA-ES solver Korali::Solver::CMAES. The location of the MAP is found at

  1. PosX1 = 0.2086, PosY1 = 0.2194
  2. PosX2 = 0.8040, PosY2 = 0.8028
  3. PosX3 = 0.7058, PosY3 = 0.709230.

  Notice that the location of candles 2 & 3 are interchangeable. Which we can observe from repeated runs.
  Also we find that CMA-ES sometimes converges to a different maximum (PosX1 = 0.2673, PosY1 = 0.1642, PosX2 = 0.7016, PosY2 = 0.7107, PosX3 = 0.8040 and PosY3 = 0.7995).

b) (20 Points) Thanks to your assessment, treating metal sheets with the new procedure has the quality of the cars chassis has greatly improved. 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 torches has a mean of 25cm (if working in the first half of the plate), and 75cm (if working on the second half), with a 5cm standard deviation in either case. 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 torches?
- Use Korali to compare the *Evidence* with or without the prior information presented by the workers. What conclusion can you make from this comparison?

An example solution for the implementations in file *hw3b/solution_codes/task2/task2a.cpp*. The experiments have been carried out with the shell scripts *run_task2b.sh* and *run_task2b_alt.sh*. Please make sure that you loaded the necessary modules.

- From the feedback of the workers we update the priors as follows:
  - Either we decrease the width of the the uniform priors such that the upper and lower bound is symmetrically centred around the means 0.25 and 0.75 and such that the standard deviation equals 0.05 (we know that $\sigma = \frac{x_{up} - x_{low}}{\sqrt{12}}$ for a uniform distribution) (see *task2b.cpp*).
  - Or we select a normally distributed prior for the x coordinate with $\mu_1 = 0.25$, respectively $\mu_2 = 0.75$, and $\sigma = 0.05$ (see *task2b_alt.cpp*).
- We repeat the steps from task 2a (model selection based on evidence followed by posterior maximization with CMA-ES) and we find that model 3 returns the greatest evidence followed by model 1 (standard deviation in bracket):
  1. -93 (1)
  2. -120 (8)
  3. -80 (12)

  Optimizing the posterior distribution of the coordinates of the candles yields the same locations as found in previous task. This result is expected since the support of the updated priors includes the optima from task 2a (note that this holds only for uniform priors, if you choose to work with a Gaussian prior in task 2b you might find a different optimum).
- By comparing the results from task 2a and 2b we can see that in general the evidence increased and the standard deviation of the experiments reduced. We can expect an increased evidence since the support of the new "smaller" uniform prior includes the location of the maximal likelihood while reducing the number of possible inputs / locations (recommended read:

https://www.cs.princeton.edu/courses/archive/fall09/cos597A/papers/MacKay2003-Ch28.pdf).