TH LFNS toolbox, as well as examples and this very tutorial can be found on `https://github.com/Mijan/LFNS`, it can be downloaded using

```
git clone https://github.com/Mijan/LFNS.git}
```

This allows in particular to use

```
git pull
```

to get all latest versions and updates.

1 Installation

The first exercise is to get the LFNS toolbox ready to be used. This can be done in one of the following three ways:

1.1 Install the toolbox on a Unix-based system

The most straight forward way to use the toolbox is to install it using cmake. On Linux systems (and for the most part Mac systems) this can be done fairly easily by first cloning the git repository (by typing `git clone https://github.com/Mijan/LFNS.git`) and following the instructions on https://mijan.github.io/LFNS/.

1.2 Use LFNS in a Docker container (all system)

You can download and install Docker for any operating system and running the LFNS Docker image. For this follow the following steps:

- Download Docker from https://www.docker.com/get-started and install it using the instruction.
- Download the docker image file lfns.tar from https://polybox.ethz.ch/index.php/s/XOwXhgB10BRbY and the run script from https://polybox.ethz.ch/index.php/s/EPOkSU0Z7EhL0sJ.
- Put the lfns.tar file and the run_lfns.sh into the same folder, which we will refer to `/lfns`.
- Make sure the script run_lfns.sh is executable. This can be done in most OS by right clicking it and clicking a button.
- Load the docker image lfns.tar by typing `docker load < lfns.tar`.
- Run the docker image by running the provided script by typing `sh run_lfns.sh`. This will run the docker image in interactive mode and will mount the current folder on docker. Now you should be able to just type the lfns commands in the console. To test this you can type `simulate --help`. If this produces a help message, everything went well!

1.3 Install LFNS on Euler

It might be the easiest to install the LFNS toolbox on Euler and use it there.

- Log into Euler (from a console you can type `ssh your-user-name@euler.ethz.ch`)
- Clone the git repository by typing

```
git clone https://github.com/Mijan/LFNS.git
```

- The git repository comes with a script file to install all the required libraries and toolbox on euler. To run the script change the directory to `$HOME/LFNS/scripts` and call the script `euler_install_script.sh`

```
cd LFNS/scripts
source euler_installation_script.sh
```

- To make sure that Euler saves the location of the installation path, open the `.bashrc` file in the home folder and add the following lines

```
module load open_mpi boost/1.59.0
export LD_LIBRARY_PATH=$HOME/local/lib:$HOME/local/lib64:$LD_LIBRARY_PATH
export LIBRARY_PATH=$HOME/local/lib:$HOME/local/lib64:$LIBRARY_PATH
export CPATH=$HOME/local/include:$CPATH
export PATH=$HOME/local/bin:$PATH
export CPPFLAGS="${CPPFLAGS} -I${BOOST_INCLUDEDIR}"
export LDFLAGS="-L${BOOST_LIBRARYDIR} ${LDFLAGS}"
```

- You can test if everything has worked by typing `simulate --help`, which should produce a help message.

2 The antithetic controller

In this exercise you will write the model files for the antithetic controller and simulate it. For information about the files see the document doc.pdf (either online or in the folder LFNS/Documentation).

2.1 Write the model files

The model reaction for the antithetic controller has 3 species $z1$, $z2$ and $x1$, The reactions follow mass action kinetics and the reaction are

$$\begin{array}{lcrll} \emptyset & \rightarrow & z1 & \text{at rate} & \mu \\ z1 & \rightarrow & z1 + x1 & \text{at rate} & k \\ x1 & \rightarrow & \emptyset & \text{at rate} & \gamma \\ x1 & \rightarrow & x1 + z2 & \text{at rate} & \theta \\ z1 + z2 & \rightarrow & \emptyset & \text{at rate} & \eta \end{array}$$

Use the templates in the `antithetic/` folder and edit the files `antithetic_model.txt`, `antithetic_initial.txt` and `antithetic_measurement.txt` to represent the above model. As initial conditions, set all the species to 0, and for a measurement take the species $x1$ and add a Gaussion noise with mean 0 and standard deviation 0.1.

2.2 Write the config file

- Modify the `<model>` and the `<Simulation>` block of the file `antithetic_config_file.xml` to contain the relative paths of your model files and specify the parameter to be simulated to all be set to 1. Specify the model type to be deterministic `<type>DET</type>`
- Simulate the antithetic model by typing `simulate antithetic_config_file.xml`. This command should simulate the system and produce a number of output files in the same folder as the `simulate antithetic_config_file.xml` file.
- Take a look at the `results_model_summary.txt` file to make sure the simulated system is the correct one.
- You can use the script `LFNS/scripts/plotSystem.m` to plot the system using matlab. For this type `plotSystem(results_model_summary.txt)`.
- Change the model type in the `antithetic_config_file.xml` under `<type>` to STOCH to simulate the system using SSA. Call the simulate command with the option `-n 100` to simulate 100 SSA trajectories. Use again the `plotSystem` command to plot the simulation outcome.
- You can simulate the system again and play around with the option (you can see a list of them by typing `simulate --help`.

2.3 Add a perturbation to the antithetic model

Now modify the model files to contain a system perturbation.

- Modify the `antithetic_model.txt` file by adding another parameter `pert` that is added to the propensity of the production propensity of `x1`.

- Add an `<inputs>` block to the `antithetic_config_file.xml` that defines a new experiment and sets the parameter value of `pert` to 1 in the times between 10 and 50 minutes.
- Add an `<experiments>` entry to the `<Simulation>` block in the `antithetic_config_file.xml` and make sure the experiment name is the same as in the `<experiments>` entry in the `<input>` block.
- Simulate the new experiment and plot the system states.