# A note before instructions of MIDA across platforms

How MIDA works in different OS system:

Windows:

- Graphic (without install anything)
- Command line (without install anything)

Mac:

- Graphic (without install anything)
- Command line (without install anything)

Linux:

- Command line (install numpy, pandas, matplotlib packages)

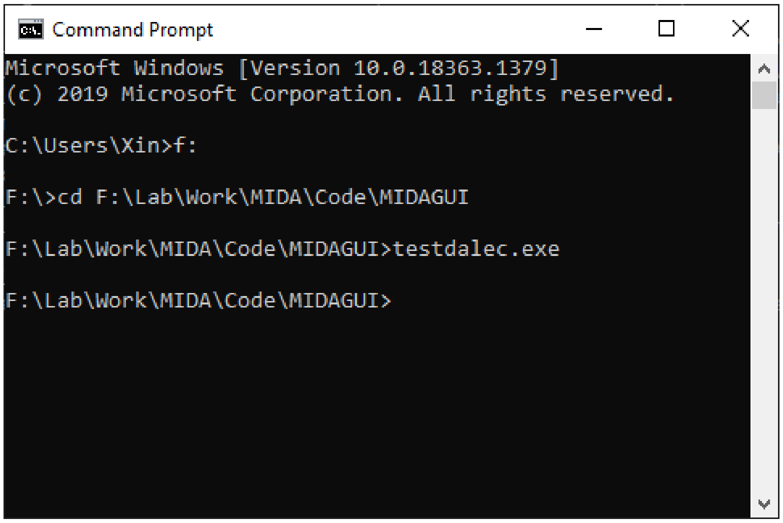
Although all executable MIDA across different OS systems have the same name, they are different files. For example, users cannot run MIDA.exe (executable MIDA in Windows) in Mac.

No matter which platform, users need to prepare both model (an executable file) and data (paramValue.txt, param.csv, config.txt, text files about observations and observation variances, namelist.txt, printDA.csv, etc.). The instructions from Jon and Yuan well introduce the data to be prepared for MIDA. The below shows some tips to be taken care of when users prepare a model.

In each iteration, MIDA will write new parameter values to paramValue.txt under the work path as indicated in namelist.txt. After model simulation, MIDA will read model outputs from text files in directories indicated in config.txt. Accordingly, users need to make sure that model executable is able to read parameter values from paramValue.txt and write model outputs to the files indicated in config.txt. The detailed information is available in step3 of Windows-Graphic version. In addition, model codes usually use relative directory to read file (e.g., ./paramValue.txt is relative directory and D:/work/paramValue.txt is absolute directory). Notice, the relative directory is referred to where MIDA.exe is rather than the directory of model executable. Taken ./paramValue.txt as an example, the paramValue.txt needs to be in the same folder as MIDA.

### Windows - Graphic version:

1. Make sure model executable is able to work by running it in the windows terminal

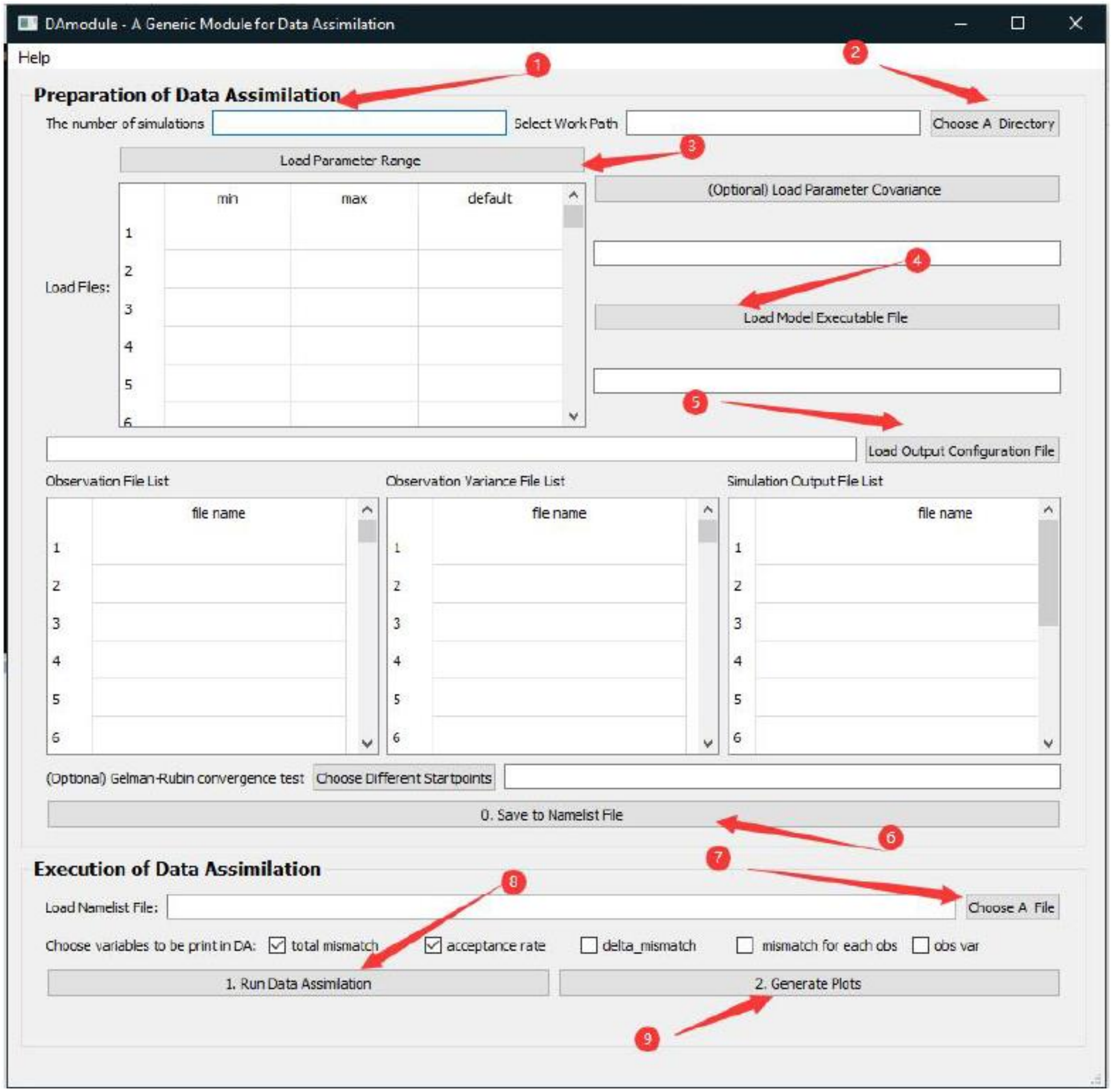


```
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Xin>f:

F:\>cd F:\Lab\Work\MIDA\Code\MIDAGUI

F:\Lab\Work\MIDA\Code\MIDAGUI>testdalec.exe

F:\Lab\Work\MIDA\Code\MIDAGUI>
```

   If there is no error and the simulation output files are newly generated, users can use the model executable for MIDA. Otherwise, users may encounter more problems in the following steps.

2. Double click MIDA.exe to start MIDA. Two windows will generate: one is the main window and the other is a black terminal. The main window is for users to interact with MIDA and the terminal window is to show print information. When either of these two windows is closed, MIDA will exit.

3. When users input in the upper panel to generate namelist.txt, MIDA will check the accuracy of related data file. Some checking rules are described below.


DAmodule - A Generic Module for Data Assimilation
Help
Preparation of Data Assimilation
The number of simulations
Select Work Path
Choose A Directory
Load Parameter Range
(Optional) Load Parameter Covariance
min
max
default
Load Files:
Load Model Executable File
Load Output Configuration File
Observation File List
Observation Variance File List
Simulation Output File List
file name
(Optional) Gelman-Rubin convergence test
Choose Different Startpoints
0. Save to Namelist File
Execution of Data Assimilation
Load Namelist File:
Choose A File
1. Run Data Assimilation
2. Generate Plots
1
2
3
4
5
6
7
8
9

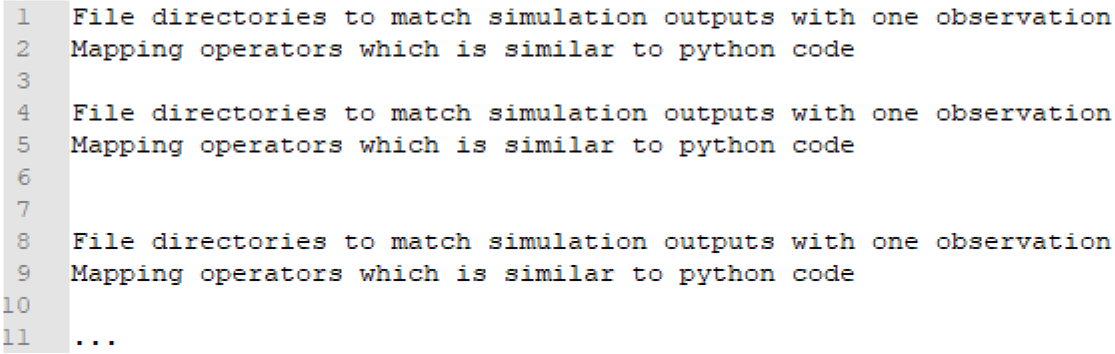
Choose variables to be print in DA: ☑ total mismatch ☑ acceptance rate □ delta_mismatch □ mismatch for each obs □ obs var

(1) The number of simulations: Check whether the input is a positive number

(2) Load parameter range (e.g., param.csv): (a) Check the whether the csv file is readable; (b) whether at least one parameter is selected for DA; (c) whether the parameters selected for DA have reasonable parameter range (min<default<max)

(3) Load model executable: Once loaded, the model executable will run for one time to check it is workable in the current computer platform. For this purpose, paramValue.txt are required to be prepared before using MIDA.

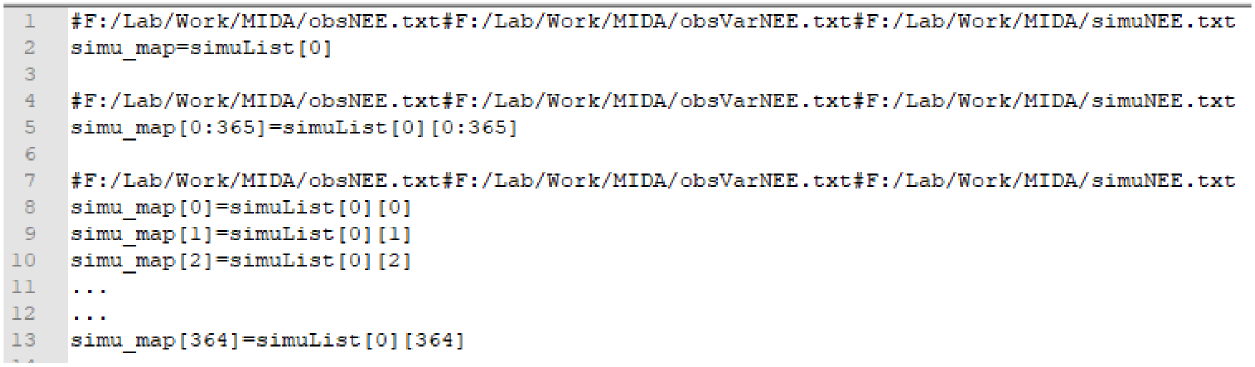
(4) Load output configuration (e.g. config.txt): The syntax of config.txt is like below:



```
File directories to match simulation outputs with one observation
Mapping operators which is similar to python code

File directories to match simulation outputs with one observation
Mapping operators which is similar to python code


File directories to match simulation outputs with one observation
Mapping operators which is similar to python code

...
```

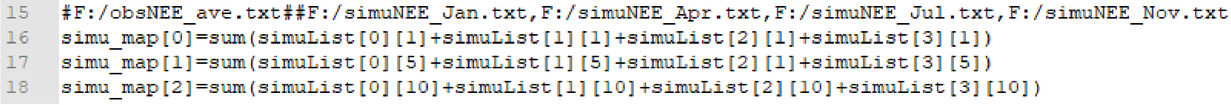
Each mapping block involves one line of file directories and several lines of mapping operators. File directory should start with # (i.e., #obsDir#obsVarDir#CorrespondingSimuOutputs). If there is no observation variance, please leave obsVarDir blank but do not forget to add # (i.e, #obsDir# #CorrespondingSimuOutputs).The mapping operators are similar to Python expressions. Therefore, the mapping operators support numerical calculations (e.g., +, -, *, /, sum(), math.pow(),etc.).

In mapping operators, *simu_map* saves simulation outputs after mapping and the length of *simu_map* is same with the observation indicated in the previous line about observation directory. *simuList* stores several simulation outputs before mapping and these simulation outputs are also indicated in the previous line. simuList[0] correspond to the first simulation output and simuList[0][0:5] indicate the first five elements in the first simulation output. Below are three ways to map a one-year daily NEE to observed NEE. Notice, for continuous elements (e.g., [0:365]), the last number is not selected (e.g. [0:365] equals 0, 1, 2, 3, …, 364). For more detailed information, please refer to slice in Python program.



```
#F:/Lab/Work/MIDA/obsNEE.txt#F:/Lab/Work/MIDA/obsVarNEE.txt#F:/Lab/Work/MIDA/simuNEE.txt
simu_map=simuList[0]

#F:/Lab/Work/MIDA/obsNEE.txt#F:/Lab/Work/MIDA/obsVarNEE.txt#F:/Lab/Work/MIDA/simuNEE.txt
simu_map[0:365]=simuList[0][0:365]

#F:/Lab/Work/MIDA/obsNEE.txt#F:/Lab/Work/MIDA/obsVarNEE.txt#F:/Lab/Work/MIDA/simuNEE.txt
simu_map[0]=simuList[0][0]
simu_map[1]=simuList[0][1]
simu_map[2]=simuList[0][2]
...
...
simu_map[364]=simuList[0][364]
```

Below is a complex example of mapping. It is to map four simulation outputs (simuNEE_Jan.txt, simuNEE_Apr.txt, simuNEE_Jul.txt, simuNEE_Nov.txt) to one observation (obsNEE.ave.txt). The first line indicates the directories of these files and the observation variance is missing. Following are three mapping operators. *simu_map* saves the output values after mapping and it is to be compared with the observation (i.e., obsNEE_ave.txt). *simuList* saves all these four simulation outputs. *simuList[0]* saves the first output of the four output files (i.e., simuNEE_Jan.txt) while *simuList[3]* saves the last output (i.e., simuNEE_Nov.txt). Then *simuList[0][1]* corresponds the second element of the first output (i.e, the second element in simuNEE_Jan.txt). Similarly, *simuList[3][1]* is the second element in the fourth output file (i.e., simu_Nov.txt). Therefore, the first mapping operator is to average the second elements in all four output files as the first element in *simu_map*. Similarly, the second mapping operator is to average the sixth elements in all four outputs and final operator is to average the 11th elements of all outputs. As a result, *simu_map* becomes a three-element vector. Later, *simu_map* will be compared with observation in MIDA to estimate the discrepancy between observation and simulation outputs.



```
#F:/obsNEE_ave.txt##F:/simuNEE_Jan.txt,F:/simuNEE_Apr.txt,F:/simuNEE_Jul.txt,F:/simuNEE_Nov.txt
simu_map[0]=sum(simuList[0][1]+simuList[1][1]+simuList[2][1]+simuList[3][1])
simu_map[1]=sum(simuList[0][5]+simuList[1][5]+simuList[2][1]+simuList[3][5])
simu_map[2]=sum(simuList[0][10]+simuList[1][10]+simuList[2][10]+simuList[3][10])
```

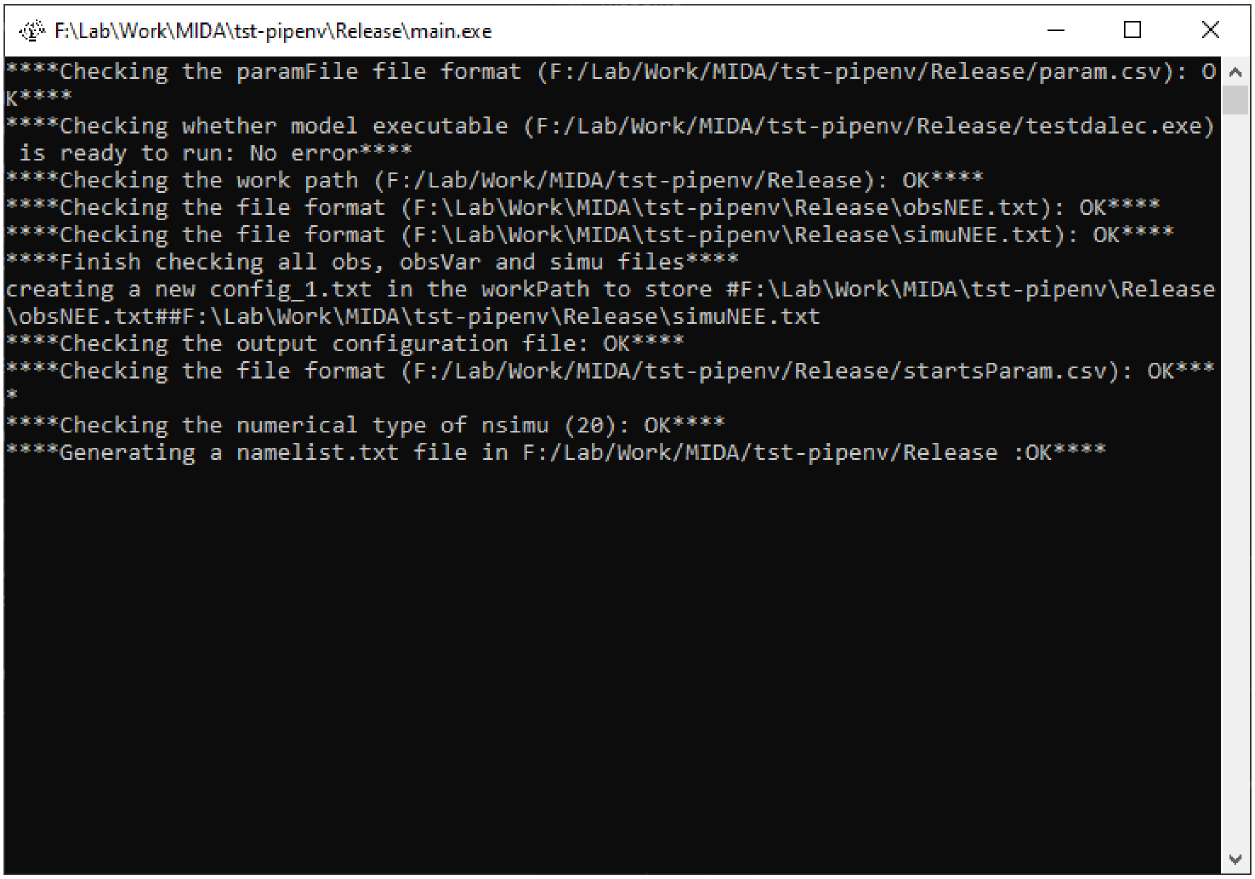
The format of config.txt will be checked:

a) Before loading config.txt, user need to select a work path and load model first.
b) The first line cannot be empty
c) The directories of observation and corresponding simulation outputs cannot be empty
d) Different mapping blocks should be separated by an empty line. Two empty lines indicate the end of config.txt no matter it is the real end or not
e) The length of observation is the same as that of corresponding observation variance
f) MIDA will read all files involved in config.txt to check whether they exist and are readable text file.

(5) Choose start-point file for G-R convergence test (e.g. startPoints.csv): (a) Check the file loaded is reasonable csv file; (b) The number of columns need to equal to the number of parameters selected for DA as indicated in param.csv; (c) The number of rows need to equal to the number of MCMC chains used for G-R convergence test. Please refer to step 5

for more information. (d) Check the default values for each parameter are within its parameter range.

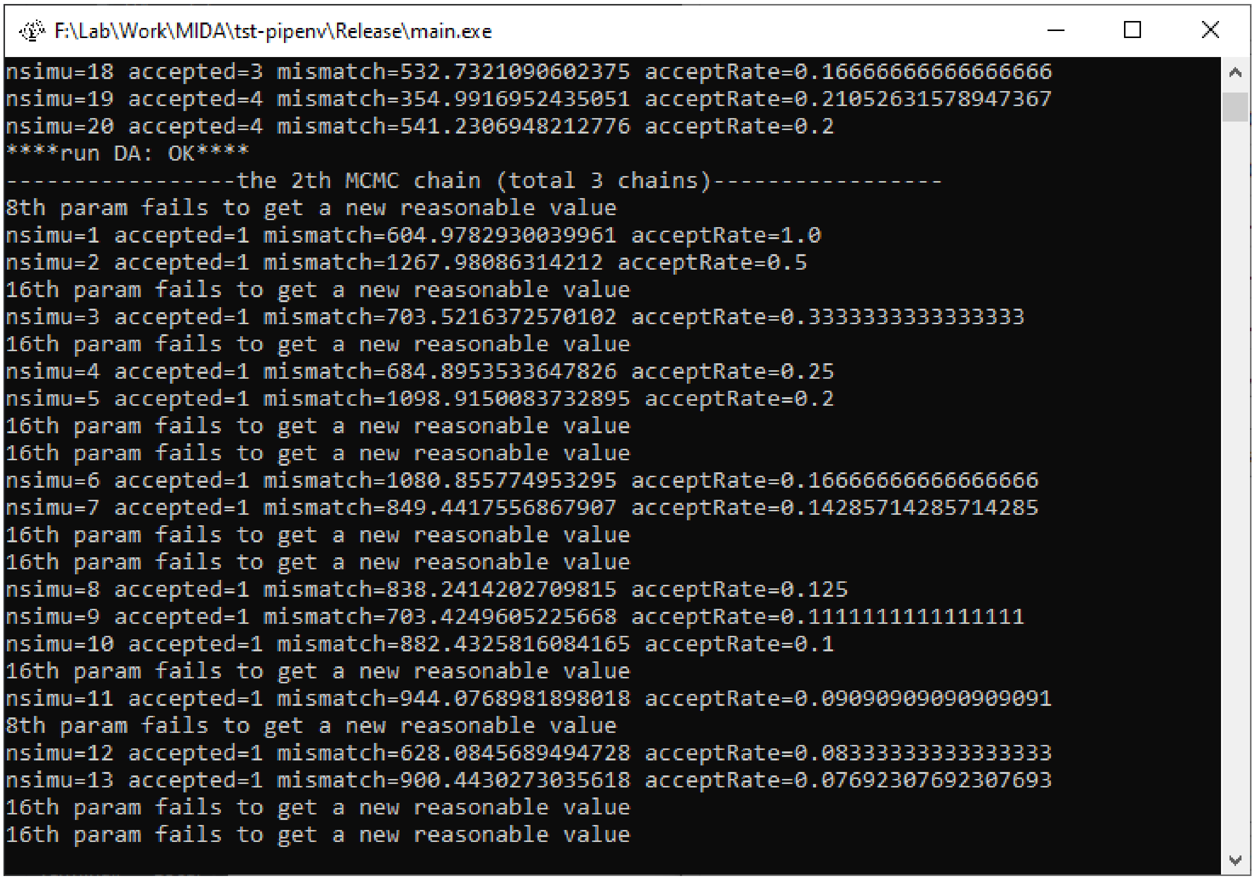
4. If users prepare a namlist.txt by hand, the checks in step3 will also be executed in MIDA when namelist.txt is loaded in the below panel.
5. Whenever users click a button, what happened behind in MIDA will be print in the other window (a black terminal). Figure below is an example of print information when a new namelist.txt is generated in MIDA. Detail errors will also be print to help users to debug.



F:\Lab\Work\MIDA\tst-pipenv\Release\main.exe

```
****Checking the paramFile file format (F:/Lab/Work/MIDA/tst-pipenv/Release/param.csv): O
K****
****Checking whether model executable (F:/Lab/Work/MIDA/tst-pipenv/Release/testdalec.exe)
 is ready to run: No error****
****Checking the work path (F:/Lab/Work/MIDA/tst-pipenv/Release): OK****
****Checking the file format (F:\Lab\Work\MIDA\tst-pipenv\Release\obsNEE.txt): OK****
****Checking the file format (F:\Lab\Work\MIDA\tst-pipenv\Release\simuNEE.txt): OK****
****Finish checking all obs, obsVar and simu files****
creating a new config_1.txt in the workPath to store #F:\Lab\Work\MIDA\tst-pipenv\Release
\obsNEE.txt##F:\Lab\Work\MIDA\tst-pipenv\Release\simuNEE.txt
****Checking the output configuration file: OK****
****Checking the file format (F:/Lab/Work/MIDA/tst-pipenv/Release/startsParam.csv): OK***
*
****Checking the numerical type of nsimu (20): OK****
****Generating a namelist.txt file in F:/Lab/Work/MIDA/tst-pipenv/Release :OK****
```

6. Run DA. If users click ‘Run DA’ button, the main window will be frozen, and users cannot type anything in this window. Information about the process of DA will be print out in the terminal.



```
nsimu=18 accepted=3 mismatch=532.7321090602375 acceptRate=0.16666666666666666
nsimu=19 accepted=4 mismatch=354.9916952435051 acceptRate=0.21052631578947367
nsimu=20 accepted=4 mismatch=541.2306948212776 acceptRate=0.2
****run DA: OK****
-----------------the 2th MCMC chain (total 3 chains)-----------------
8th param fails to get a new reasonable value
nsimu=1 accepted=1 mismatch=604.9782930039961 acceptRate=1.0
nsimu=2 accepted=1 mismatch=1267.98086314212 acceptRate=0.5
16th param fails to get a new reasonable value
nsimu=3 accepted=1 mismatch=703.5216372570102 acceptRate=0.3333333333333333
16th param fails to get a new reasonable value
nsimu=4 accepted=1 mismatch=684.8953533647826 acceptRate=0.25
nsimu=5 accepted=1 mismatch=1098.9150083732895 acceptRate=0.2
16th param fails to get a new reasonable value
16th param fails to get a new reasonable value
nsimu=6 accepted=1 mismatch=1080.855774953295 acceptRate=0.16666666666666666
nsimu=7 accepted=1 mismatch=849.4417556867907 acceptRate=0.14285714285714285
16th param fails to get a new reasonable value
16th param fails to get a new reasonable value
nsimu=8 accepted=1 mismatch=838.2414202709815 acceptRate=0.125
nsimu=9 accepted=1 mismatch=703.4249605225668 acceptRate=0.1111111111111111
nsimu=10 accepted=1 mismatch=882.4325816084165 acceptRate=0.1
16th param fails to get a new reasonable value
nsimu=11 accepted=1 mismatch=944.0768981898018 acceptRate=0.09090909090909091
8th param fails to get a new reasonable value
nsimu=12 accepted=1 mismatch=628.0845689494728 acceptRate=0.08333333333333333
nsimu=13 accepted=1 mismatch=900.4430273035618 acceptRate=0.07692307692307693
16th param fails to get a new reasonable value
16th param fails to get a new reasonable value
```

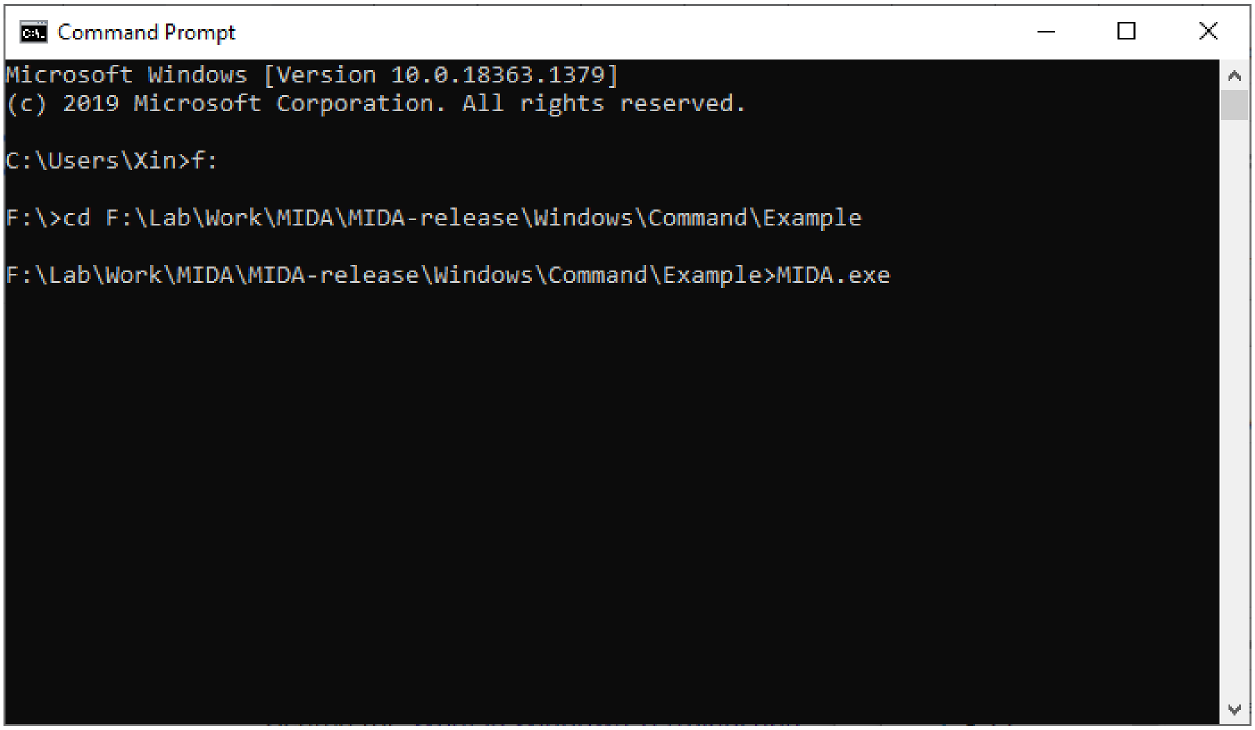
In the figure above, *nsimu* is the number of simulations executed so far and *accepted* is the number of simulations accepted. The *acceptRate=accepted/nsimu*. Generally, *acceptRate* is 20%~40%. ‘i-th param fails to get a new reasonable value’ means the new value of the i-th parameter is not within its parameter range. Then, MIDA will re-generate new parameter values until all parameters are reasonable. If users want to stop DA, users need to select the black terminal where print information is popping out and type CTLR+C. This hot key will terminate the execution of DA without exiting MIDA.

A G-R convergence test requires to run multiple MCMC chains. In the namelist.txt, *nChains_ConvergeTest* is the number of MCMC chains and *nChains_ConvergeTest=0* indicates no convergence test. *convergeTest_startsFile* is the csv file which saves the default values of parameters as the start points in multiple MCMC chains. After G-R convergence test, the G-R estimators will be saved in a file indicated as *outConvergenceTest*. All DA outputs will be saved in the DAresult/ folder in the work path as indicated in namelist.txt.

7. Visualization. Generally, there will be two figures generated: posterior distribution of parameters and the mismatches between observations and simulation outputs during DA. If convergence test is used, the number of figures = 2*number of MCMC chains + 1

**Windows - Command line version**

1. Prepare namelist.txt and printDA.csv and place them at the same folder as MIDA.exe. printDA.csv indicates what information (totally there are five choices) will be printed in each iteration during DA. For example, the mismatches between observations and simulation outputs. Specially, users must select at least one to be print in printDA.csv.
2. Open a windows command terminal, navigate to the location of MIDA.exe and type 'MIDA.exe' to start MIDA



```
Microsoft Windows [Version 10.0.18363.1379]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Xin>f:

F:\>cd F:\Lab\Work\MIDA\MIDA-release\Windows\Command\Example

F:\Lab\Work\MIDA\MIDA-release\Windows\Command\Example>MIDA.exe
```

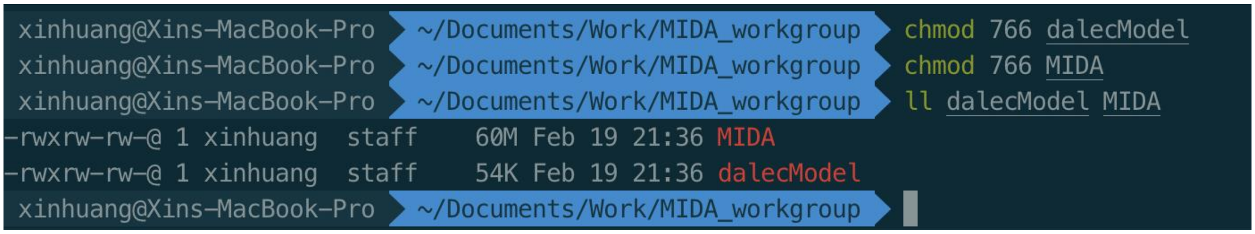
3. MIDA will automatically read namelist.txt, conduct DA and visualize DA results. If *display_plot* in namelist.txt is 1, plots will be visualized in new windows. If *display_plot* has a value of 0, MIDA will save plots to DAresults/ folder. Users may use the plotScript.py to plot DA results without MIDA (Type 'python plotScript.py' in the terminal). In this case, users need to install numpy, pandas, and matplolib Python packages before using

plotScript.py. All information regarding to DA in MIDA (e.g., checking file formats before DA and the ratio of accepted simulation during DA) will be printed in the terminal to help users to debug.

4. Please enter to exit MIDA.

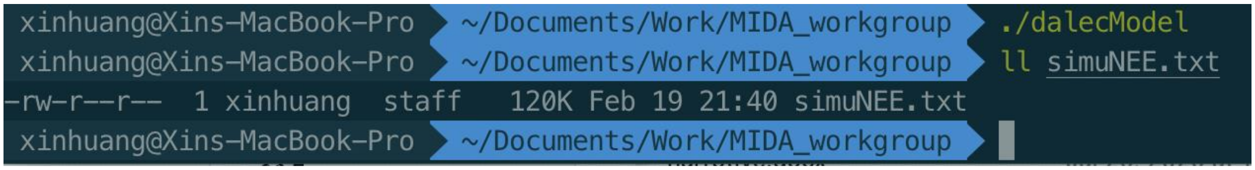
### Mac - Graphic version

1. Make sure the model and MIDA is executable. Users may directly type the commands below in mac terminal to make model and MIDA to be executable.



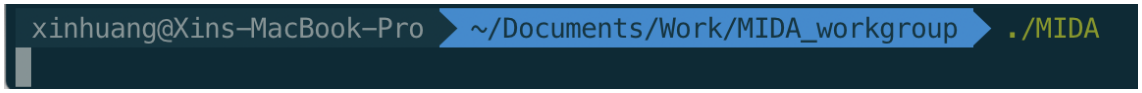
```
 xinhuang@Xins-MacBook-Pro  ~/Documents/Work/MIDA_workgroup  chmod 766 dalecModel
 xinhuang@Xins-MacBook-Pro  ~/Documents/Work/MIDA_workgroup  chmod 766 MIDA
 xinhuang@Xins-MacBook-Pro  ~/Documents/Work/MIDA_workgroup  ll dalecModel MIDA
-rwxrw-rw-@ 1 xinhuang  staff    60M Feb 19 21:36 MIDA
-rwxrw-rw-@ 1 xinhuang  staff    54K Feb 19 21:36 dalecModel
 xinhuang@Xins-MacBook-Pro  ~/Documents/Work/MIDA_workgroup 
```

2. Make sure model executable is able to run properly before using MIDA. Type the command below to check whether no error occurs, and the simulation outputs are just generated. Otherwise, users may encounter more errors in the following steps.



```
 xinhuang@Xins-MacBook-Pro  ~/Documents/Work/MIDA_workgroup  ./dalecModel
 xinhuang@Xins-MacBook-Pro  ~/Documents/Work/MIDA_workgroup  ll simuNEE.txt
-rw-r--r--  1 xinhuang  staff   120K Feb 19 21:40 simuNEE.txt
 xinhuang@Xins-MacBook-Pro  ~/Documents/Work/MIDA_workgroup 
```

3. Different from windows-graphic version, users with Mac have to type './MIDA' to start MIDA in the terminal. Two windows will generate: one is the main window and the other is the terminal.



```
 xinhuang@Xins-MacBook-Pro  ~/Documents/Work/MIDA_workgroup  ./MIDA
```

4. The following steps are similar to those in windows-Graphic MIDA.

### Mac - Command line version

1. Similar to Mac-graphic version, users need to make sure model and MIDA are executable. Moreover, model executable can run accurately before MIDA (Please see step 1 in Mac-Graphic section to get more detailed information)

2. Similar to Windows-command version, users need to prepare a namelist.txt and printDA.csv and place them under the same folder of MIDA (Please see step 1 in Windows-Command section to get more detailed information)
3. In the terminal, navigate to the folder of MIDA executable, type ‘./MIDA’ to start MIDA. MIDA will automatically read namelist.txt and related data, conduct DA and visualize DA results. If *display_plot* in namelist.txt is 1, plots will be visualized in new windows. If *display_plot* has a value of 0, MIDA will save plots to DAresults/ folder.
4. After that, type anything to exit MIDA.

### Linux – Command line version

1. Install Python 3.x, and related packages (i.e., numpy, pandas, and matplotlib)
2. Similar to Windows/Mac, make sure model is executable and is able to run correctly before using MIDA
3. Prepare a namelist.txt and printDA.csv and place them under the same folder of main.py. Please refer to Windows-Command section to get more information of printDA.csv.
4. If ‘python’ command triggers python 3.x, type ‘python main.py’ to start MIDA. The workflow is same as the above command-versions in Windows/Mac.