

# Frequently asked questions

## General

### More details

#### More information

For more information, refer to the [user manual](#).

## Modeling

### Random seed

#### Random seed

- Mersenne Twister from C\+\+ compiler (gcc version > 4.6) or Boost library when available.
- Time → RandomSeed

#### Reproducibility

- In simulation using multithreading (openMP), each thread gets its own random seed. Therefore, to reproduce exact results, one needs to specify the same random seed as well as the same number of threads.

### Spatial scale

#### Lattice scale

- Space → NodeLength

This affects the diffusion coefficient in PDE models.

Dimensionless space is assumed, unless a unit is specified.

## Analysis

### Data output and visualization

#### Logger

*Note: this plugin will be restructured in the near future.*

The Logger periodically writes the values of the symbols specified in the Format to file.

Its Plot function can generate timeplots, or phase plots, depending on the columns selected for the axes. Note: these columns correspond to the column number in the generated data file (first column = 1).

## HistogramLogger

HistogramLogger calculates frequency distributions and writes them to file.

Its Plot function can generate timeplots.

## Parameter sweep / Batch processing

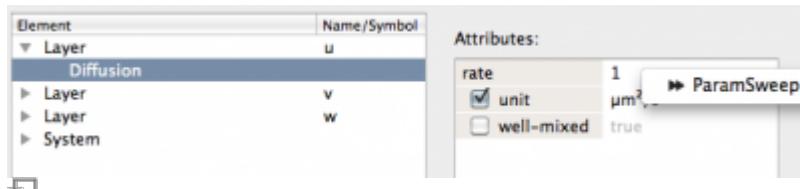
To explore the effect of a parameter on model behavior, you can set up a batch process called ParamSweep.

### 1. Select parameters

In the Attributes window, press right-click on a parameter and choose ParamSweep. A 'fast-forward' symbol appear indicating that this parameter is set for a parameter sweep.

Alternatively, if no Attributes windows is available (e.g. with Expression), right-click in the central window to select for ParamSweep.

Note that model parameters of any type can be explored (double, int, string, enum, etc.).



Right-click in Attributes window to select parameter for ParamSweep

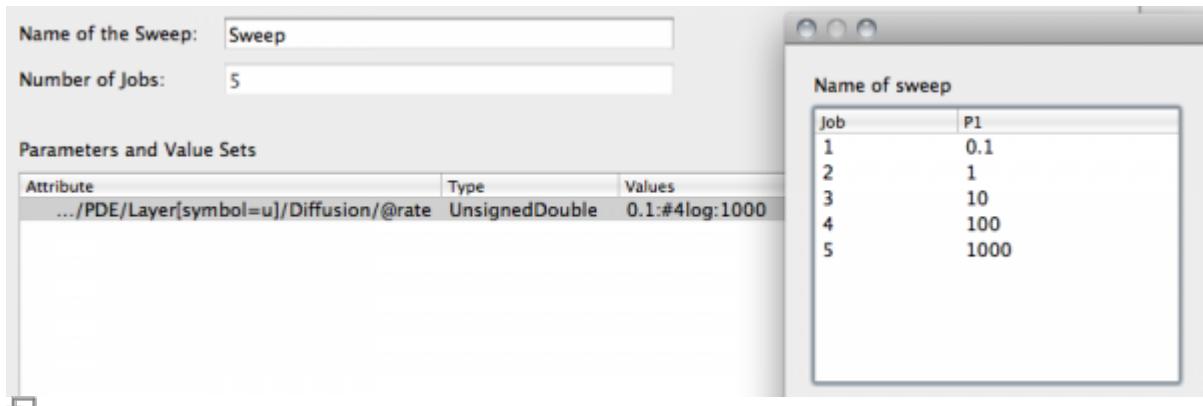
### 2. Set range of values

Once you have chosen a parameter, go to the ParamSweep view (see Documents window).

Here, you can specify a range of values for each parameter.

Syntax	Example	Generated sequence	Description
<b>value1;value2;value3</b>	1.0;1.25;4	1.0;1.25;4	Sequence of values
<b>string1;string2;string3</b>	“square”;“hexagonal”	“square”;“hexagonal”	Sequence of strings
<b>min:increment:max</b>	0:2:10	0;2;4;6;8;10	Range of values with increment
<b>min:#steps:max</b>	0:#2:10	0;5;10	Range of values with number of steps
<b>min:#steps[log]:max</b>	0.1:#2log:10	0.1;1.0;10.0	Range of values with number of steps in logarithmic scale

The Number of Jobs that will be generated is calculated and displayed.



Specify range of values in ParamSweep window and execute batch process by pressing Start.

### 3. Execute ParamSweep

To start a parameter sweep, **press Start in the toolbar** while you have the ParamSweep panel open.

Before it starts the batch process, Morpheus let's you check your jobs and parameters in a pop-up window.

Also note that batch processes can benefit from **parallel computing** by allowing multiple concurrent jobs, see **Settings → Local → Concurrent Jobs**.

Note that the parameter sweep is always run in Local mode (overrides Interactive mode).

### Multi-dimensional ParamSweeps

Multi-dimensional parameter sweep are generated by selecting multiple parameters for ParamSweep.

To couple multiple parameters such that they are incremented simultaneously, drag one over the other to 'pair' them.

### How to post-process results of a parameter sweep?

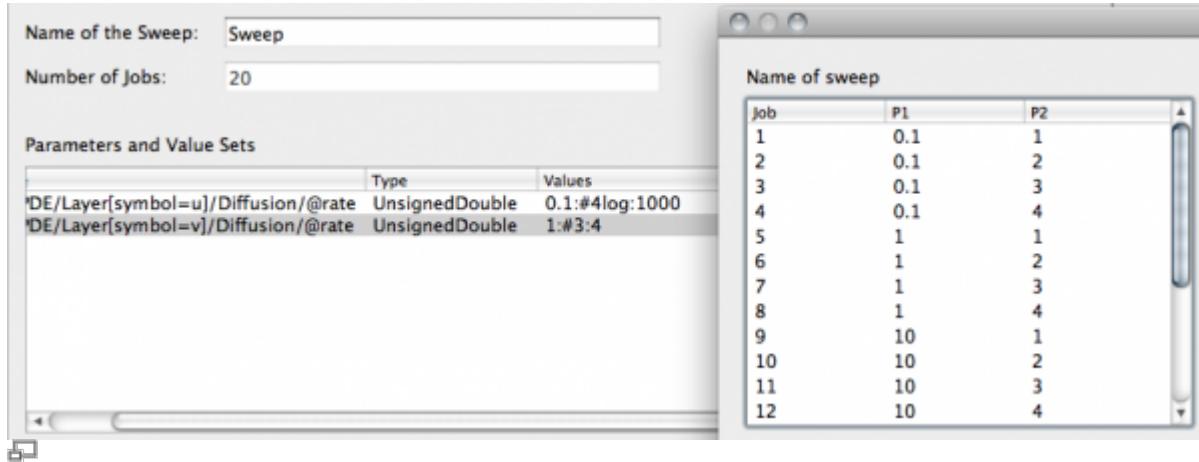
Morpheus does not include tools for post-processing of data. Thus, analyzing results should be done using your own favorite tools.

Yet, Morpheus does supply a few features that enable you to analyze results of parameter sweeps:

- Sweep summary: file containing data on parameters and the folder in which the results are saved.
- Python scripts:
  - `morphMakeTable.py`: generate  $\$Latex\$$  table from images from a parameter sweep (using sweep summary)
  - `morphSweepData.py`: gather data from log files from a parameter sweep (using sweep summary)

### ImageTable

This option can be used to create a  $\$Latex\$$  table of images from a parameter sweep. This is a simplified interface to the python script `morphMakeTable.py`.

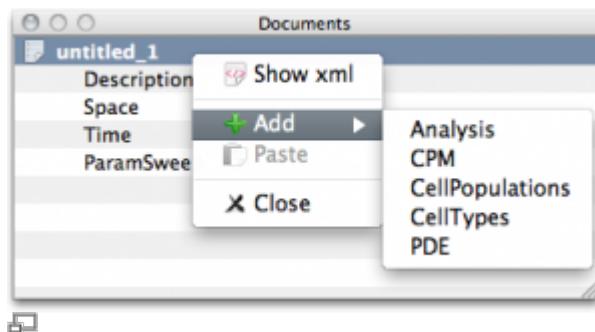


## Editing

### Add or remove model items

#### Add/Remove in Document view

To add or remove optional model components such as PDE, CellTypes, or CellPopulation, right-click in the Documents View and choose Add or Remove.



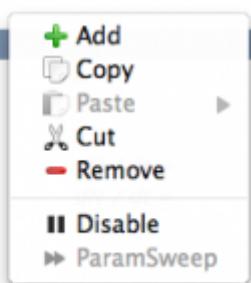
Right-click in the Documents view

#### Add/Remove in Editor view

To add or remove optional plugins, right-click **on the parent node** and choose Add or Remove. The parent node is the place on which the new plugin should appear.

In case of Remove, you will be prompted to confirm the removal.

Element	Name/Symbol
CellType	cells
Property	D =
Property	N =
Property	Dn =
NeighborsReporter	
System	
Constant	
Constant	
Constant	
DiffEqn	
DiffEqn	
CellType	



Right-click on parent node in editor view

## Save to Clipboard

To remove but save a copy on the Clipboard, choose Cut. You will not be prompted for confirmation.

## Copy/paste items

### Copy

To **copy** an item, right-click in the editor view and choose **Copy**. The XML snippet will be shown in the Clipboard.

### Paste

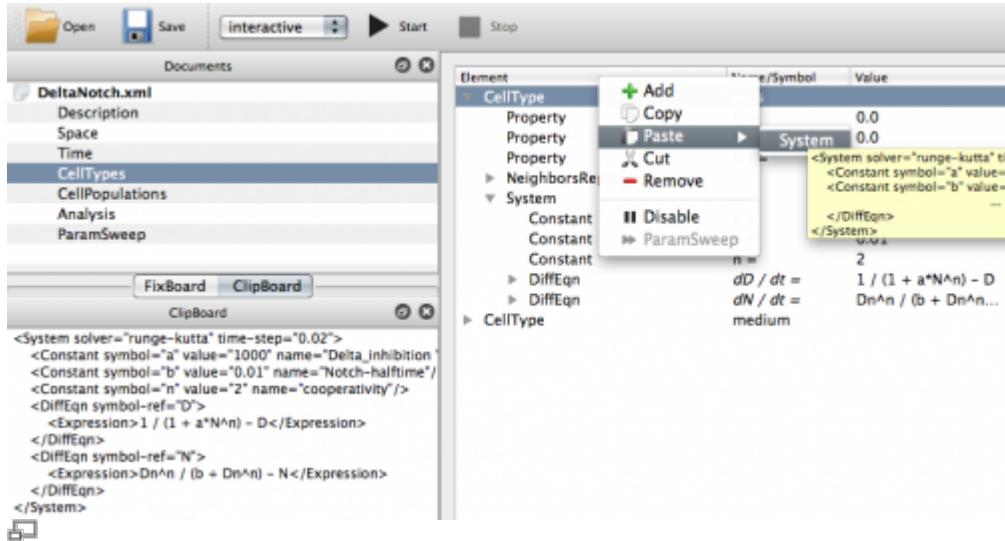
To **paste** an item that appears on the Clipboard, right-click in the editor view **on the parent item** and choose **Paste**. Choose an item from the **drop-down list**.

Note that you can copy/paste between different models.

An extract of the XML will be shown as a **tooltip**. This enables you to distinguish items with identical names.

### Clipboard

Snippets of XML that were copied or cut are shown here. The Clipboard is shared over all models in the Documents view.



## Copy/Paste between models

During model construction, it is often useful to copy/paste items between different models that appear in the Documents view. For instance, to integrate single-scale models into a multi-scale model.

This is possible, because the Clipboard is shared over all models in the Documents view.

## Disable items

### Disable

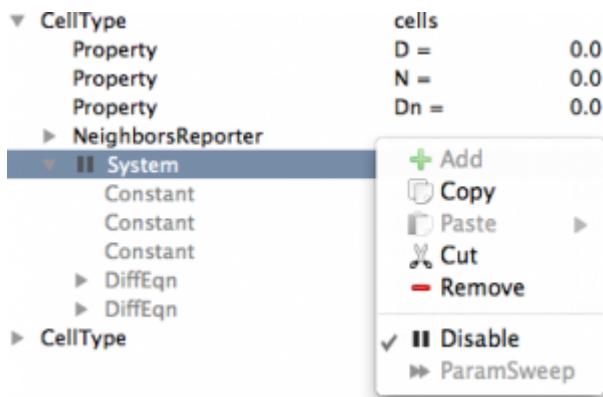
During the model construction or testing process, it is often desirable to temporarily **switch off** an item of the model without removing it from the model.

To **switch off** an item, right-click in the editor view and choose **Disable**. A 'pause' symbol appears before the node, and the item as well as its child nodes are grayed out.

Note that the item is still saved to XML, but is commented out.

### Re-enable

To re-enable, do the same: right-click in the editor view and choose **Disable**.



Switch off a model item by right-click and choose **Disable**.

## Edit XML model without GUI

### Text editor

The XML files can be opened in any text editor that allow you to manually edit the model.

### Command line tools

More advanced editing of the XML files requires command line tools like [xmlstarlet](#).

For example, to remove all Property items from the CellType called ct1, do:

```
xmlstarlet ed -d "/MorpheusModel/CellTypes/CellType[@name='ct1']/Property"
model_old.xml > model_new.xml
```

## GUI

### FixBoard

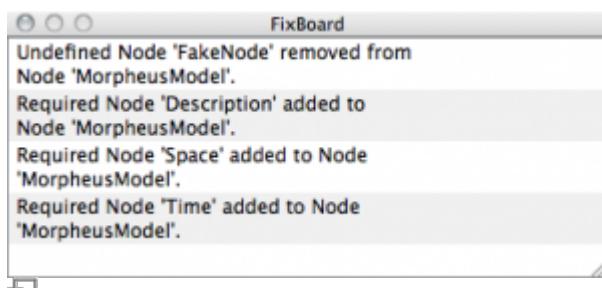
#### FixBoard

When you open a outdated or broken XML model, Morpheus will try to automagically correct and update the model. For example, Morpheus creates required elements that are missing and removes obsolete elements.

The changes that Morpheus made are shown on the Fixboard. **You are strongly advised to review those changes.**

As an example, the following message will appear in the FixBoard when we invalidate the [minimal model](#) by removing the Description, Space and Time elements and replace them with an fictitious FakeNode node:

```
<?xml version='1.0' encoding='UTF-8'?>
<MorpheusModel version="1">
  <FakeNode/>
</MorpheusModel>
```



The FixBoard shows the automatic changes make in a broken or outdated model for review.

Note that this is a feature of graphical user interface. The command line interface `morpheus` cannot correct outdated or broken models supplied as command line arguments.

## JobQueue (job archive)

### JobQueue

This panel has multiple functions:

- Overview of pending, running and terminated jobs
- Archive of simulation models
- Sorting simulation jobs and sweeps
- Access to simulation result browser
- Stop and remove jobs and results

### Result browser

#### Morpheus results browser

Simulation results can be shown by clicking the job in the JobQueue. This opens a file browser within Morpheus showing the Output Folder and the Output Text.

#### File browser and terminal

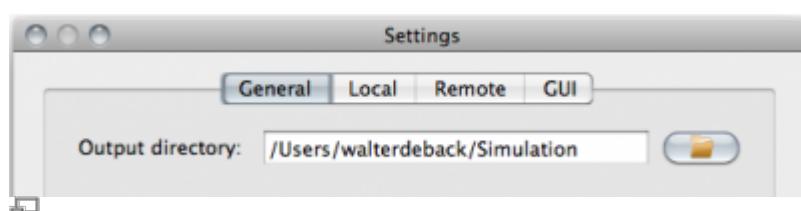
The buttons above the result browser can be used to open the Output Folder your system-wide file browser or open in a Terminal.



Use the buttons to open Output Folder in your file browser or in terminal.

#### Change results folder

To change the folder where results are stored, check the Settings.



The folder where results are stored can be changed in File→Settings.

#### Reset JobQueue archive

The information in the JobQueue is stored in using a SQLite database. This database is named `morpheus.db.sqlite` and stored in the OS-specific folder for Application Data.

Remove or rename the file `morpheus.db.sqlite` to reset the job archive. Note that this does not remove the simulation results themselves.

## Checkpointing (simulation snapshots)

### Checkpointing

Storing snapshots of the simulation state during execution is called **checkpointing**.

This allows the user to e.g.:

- Restore a simulation
- Continue a simulation under different conditions
- Use the end-state of a simulation as initial condition of a new simulation

### Enable checkpointing

Checkpointing is switched off by default.

To enable checkpointing, add the item `Time → SaveInterval`.

To set the interval, set the `Time → SaveInterval → value`.

Special cases:

- `value = -1`: Disable checkpointing
- `value = 0`: Store state only at start and end
- `value > 0`: Store state at specified time interval

### File format

Snapshots are saved in a compressed XML format with the filename `[Title][Time].xml.gz`.

### Open saved models

Snapshot files `[Title][Time].xml.gz` can be opened by double-clicking on them in the GUI.

From the command line, first unzip the file and then run `morpheus [Title][Time].xml`

### Limitations

The snapshots store the complete simulation state. However, the values of **PDE Layers are not stored**.

This is excluded because of (1) the large storage requirements and (2) the required time to write to text file. (In the future, separate binary files will be written for PDE Layeres which can be stored and written more efficiently, but are more difficult to do in a platform independemt fashion.)

### Restore parameter sweeps

Parameter sweeps can also be restored. This restores the parameters and value ranges specified for a particular model.

This feature requires an opened model that has the same parameters.

# Settings

## Multithreading / Concurrent jobs

1. Number of threads: Used to set number of openMP threads for (in-job parallelization)
2. Concurrent jobs: number of jobs executed in parallel (between-job parallelization)

## Gnuplot path

By default, Morpheus uses the first GnuPlot executable it can find in the PATH environmental variable. Optionally, you can overrule this path.

### Graphical User Interface

To use a specific GnuPlot version, specify you gnuplot executable under **File** → **Settings** → **Local** → **Gnuplot executable**.

### Command Line Interface

To specify a gnuplot executable from the command line, you can use the **gnuplot-path** option as command line argument:

```
morpheus -gnuplot-path [File]
```

From:  
<https://imc.zih.tu-dresden.de/wiki/morpheus/> - **Morpheus**

Permanent link:  
<https://imc.zih.tu-dresden.de/wiki/morpheus/doku.php?id=documentation:faq&rev=1629202254>

Last update: **14:10 17.08.2021**

