



Universität Stuttgart
DuMu^x – Course 2018



Runtime Parameters and Grids

Introduction to DuMu^x

Runtime Parameters

Runtime Parameters

- Avoid having to recompile
- Can run many simulations with different sets of parameters via shell script

Practical concern:

- Reading in runtime parameters is slow → Don't do it too often.

Input file contents

- Input file syntax:

```
[MyGroup]
```

```
MyParameter = 2
```

- Examples:

Groups	Parameters		
Component	GasDensity	LiquidDensity	MolarMass
Problem	EnableGravity	Name	
Grid	Cells	LowerLeft	UpperRight

Runtime parameters

- Use runtime parameters in any file in dumux

```
paramname_ = getParam<TYPE>("GROUPNAME.PARAMNAME");
```

or

```
paramname_ = getParamFromGroup<TYPE>("GROUPNAME", "PARAMNAME");
```

- Use runtime parameters in any file in dumux and give default value for cases when

```
paramname_ = getParam<TYPE>("GROUPNAME.PARAMNAME", default);
```

Functions hasParam and hasParamInGroup

- Check: Parameter in input file?

```
if (hasParam("GROUPNAME.PARAMNAME"))
    std::cout << "GROUPNAME.PARAMNAME is read from the input
file." << std::endl;
else
    std::cout << "Using the default value for
GROUPNAME.PARAMNAME." << std::endl;
```

- Another way of writing the upper

```
if (hasParamInGroup("GROUPNAME", "PARAMNAME"))
```

instead of

```
if (hasParam("GROUPNAME.PARAMNAME"))
```

Overview of required parameters for specific case

- Find out what runtime parameters have to be specified:
compile and then run

```
./myexecutable --help
```

Input file contents

- Input file syntax:

```
[MyGroup]
```

```
MyParameter = 2
```

- Examples:

Groups	Parameters		
Component	GasDensity	LiquidDensity	MolarMass
Problem	EnableGravity	Name	
Grid	Cells	LowerLeft	UpperRight

Grids

Grid types

- YASPGrid (structured, n-dimensional, parallel, tensorproduct grid)
 - UGGrid (2D/3D, unstructured, parallel, multi-geometry)
 - ALUGrid (2D/3D, unstructured, parallel, simplex/cube)
 - FOAMGrid (1D-2D, Dynamic, Multi-dimension/Network Problems)
 - ...
-
- Set in problem file:

```
// Set the grid type  
SET_TYPE_PROP(Injection2p2cTypeTag, Grid, Dune::YaspGrid<2>);
```

Create grid in main file

```
// try to create a grid (from the given grid file or the  
input file)  
GridManager<typename GET_PROP_TYPE(TypeTag, Grid)>  
gridManager;  
gridManager.init();
```

Input file – example grid sections

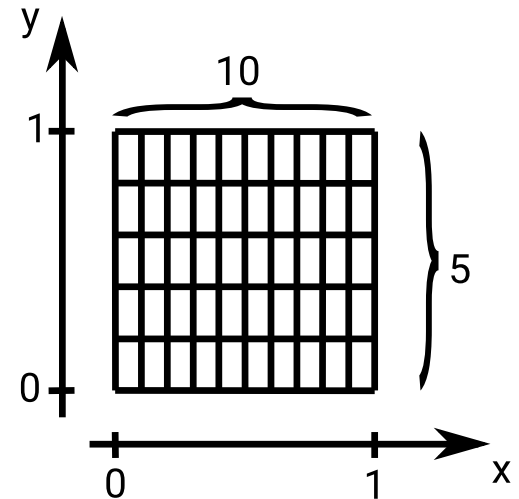
READ IN FROM FILE

```
[Grid]
File = ./grids/heterogeneousSmall.dgf # relative path to the
grid file
```

PROGRAM INSIDE CODE

```
[Grid]
LowerLeft = 0 0
UpperRight = 1 1
cells = 10 5
```

3 entries in 3D



Read grid from file – supported formats

- DGF (Dune grid format)
- Gmsh (gmsh.info)
- Cornerpoint grid format

Exercises

- Exercise about runtime parameters

Go to <https://git.iws.uni-stuttgart.de/dumux-repositories/dumux-course/tree/master/exercises/exercise-runtimeparams> and check out the README

- Exercise about grids

Go to <https://git.iws.uni-stuttgart.de/dumux-repositories/dumux-course/tree/master/exercises/exercise-grids> and check out the README

Thank you!