

Wildkatze Console



1. Create solver.sh to launch the solver.

```
mpirun -np 4 /Path_To_Executable/wildkatze -l /Path_To_License/license.wildkatze.dat "$@"
```

This allows to launch the solver with simple solver.sh on linux terminal.

2. Basic Usage and Commands.

```
Wildkatze> setdir ./
```

Sets up the path to simulation directory. From this location mesh and simulation tree files are loaded.

```
Wildkatze> setmesh meshname
```

setmesh sets the name to the mesh name (**meshname.bmsh** and **meshname.info.bmsh**). Note that extension is not mentioned, solver adds that by itself.

```
Wildkatze> setsim simfile
```

setsim sets the simulation file name. Here again the extension (**.stree**) is not mentioned, solver adds by itself.

```
Wildkatze> readsimtree
```

Loads the simulation file or simulation tree file for which name was set (in this case **simfile.stree**).

```
Wildkatze> init-solution
```

Initializes the solution and prepares for running iterations.

```
Wildkatze> iterate number-of-iterations
```

Runs the simulation for 'number-of-iteration' times. Example iterate 100, shall run 100 iterations.

```
Wildkatze> export-ensight filename
```

Exports results for post processing in Ensight format.

```
Wildkatze> save-restart-bin filename
```

```
Wildkatze> read-restart-bin filename
```

Saves and reads the restart file with filename (added with appropriate extensions).

```
Wildkatze> process-file filename
```

```
Wildkatze> pf filename
```

Processes a journal file given by filename that contains sets of Wildkatze commands.

```
Wildkatze> exit
```

Exits the solver. At the point solver writes some files like residuals, monitors etc.

3. Changing Simulation Options.

```
Wildkatze> options
```

Sets the user Options pointer to Simulation Object Options.

```
Wildkatze> op-list
```

Lists out the Options it is pointing to. In this case it is pointing to Simulation Object's options. In other cases it could be pointing to Model Options or Model Boundary Conditions.

User can change the settings of user Options by following same method. Op-list tells users the current state of the options.

```
Initializing ConjugatedFlowModel
wildkatze> options
    Setting current options to simulation options.
wildkatze> op-list

simulation-type    steady { steady, transient }
iteration          Integer 1
timeIteration      Integer 1
current-inner-iteration Integer 0
max-iterations     Integer 1
max-iteration-per-timestep Integer 4
simulation-time    Real 0
time-step-size0    Real 0.001
time-step-size1    Real 0.001
time-step-size2    Real 0.001
variable-time-step Integer 0
minimum-time-step  Real 5e-07
maximum-time-step  Real 0.0005
courant            Real 0.7
using-expert-driver Integer 0
fractional-after-iteration Integer 10
implicit-step-frequency Integer 10
fractional-solver-correctors Integer 2
transient-type     transient-0 { transient-0, transient-1, transient-2 }

wildkatze>
wildkatze>
```

Wildkatze Console



Wildkatze> `op-double time-step-size0 1.0E-5`

Sets the value of parameter **time-step-size0** equal to **1.0E-5 sec**. **Time-step-size0** is the time step size solver will use to run the simulation. Options of type Real and Integer are set by this method.

Wildkatze> `op-type simulation-type transient`

Sets the simulation type to transient. Op-type is way to select one of the enumerated options.

4. Changing Material Properties.

Wildkatze> `phase-list`

Lists the phase-names present in the simulation.

Wildkatze> `phase-material-options phase-name`

Sets the user Options pointer to material of phase phase-name. One can then change the options by using op-double and op-type commands. (op-list to see the current settings).

```
wildkatze> phase-list
Phase list:
default-phase .
flow-phase .
fluid-phase .
solid-phase .
tube-phase .

wildkatze> phase-material-options fluid-phase
Setting options for fluid-phase
wildkatze> op-list

material-type    Constant-Properties { Constant-Properties }
density-value-constant    Real 1027
viscosity-value-constant   Real 0.0003645
specific-heat-value-constant    Real 4195
thermal-conductivity-value-constant    Real 0.668011
molecular-weight    Real 28.966
wildkatze> 
```

5. Changing Model Options and Boundary Conditions.

Wildkatze> `model-list`

Lists the model-hash-tags for Models present in the simulation.

Wildkatze> `model-phase-options model-hash-tag phase-name`

Sets Model's options for the Phase with phase-name. One can use op-list, op-double and op-type to modify the options.

Wildkatze> `model-phase-bnd-options model-name phase-name region-name boundary-name`

Sets Model's options for the Phase with phase-name for boundary that present in region-name region. One can use op-list, op-double and op-type to modify the options.

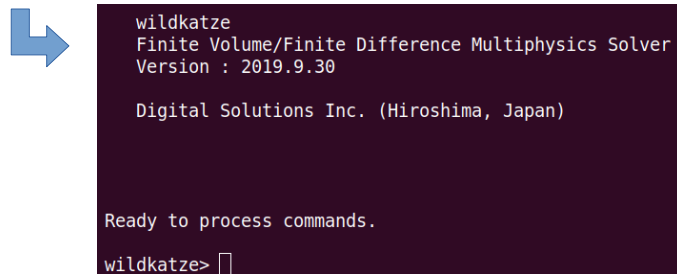
Setting Up Simulation from Console



1. Create solver.sh to launch the solver.

```
mpirun -np 4 /Path_To_Executable/wildkatze -l /Path_To_License/license.wildkatze.dat "$@"
```

After this solver could be launched by this script.



One can then pass the script arguments to the solver itself by the above modification. That means

solver.sh arg1 arg2 arg2 ...

Are passed to solver and above script is same as

wildkatze arg1 arg2 arg2 ...

2. Processing batch file batch.txt

Given the batch file batch.txt that contains various wildkatze commands to be executed, we can process it by:

solver.sh -f batch.txt

4. single phase flow set up

Given mesh in **mesh.info.bmsh** and **mesh.bmsh** files, we can set the basic single phase flow

solver.sh -lc flow mesh -lc pf o.txt

This sets up single phase (**flow-phase**) flow with phase properties being taken from Air. The simulation is set up in **output.stree**.

Note that the above command also produces batch file **o.txt** that user can rename and specialize for advance set up. In this case user can just process this batch file by **solver.sh -f batch.txt** as mentioned above.

7. Specializing the model set up

flow/vof/kpoly mesh, generates basic set up, we can specialize by providing more options to these commands.

1. Specifying the **velocity at velocity inlets**.
Example:

solver.sh -lc flow mesh -vel 5.0 -lc pf o.txt

Also specifies all the inlets as 5.0m/sec.

2. Exporting **forces on walls**.

solver.sh -lc flow mesh -vel 5.0 -forces -lc pf o.txt

This sets up forces exports on the wall boundaries.

9. Unsteady Model

Be default the solver produces set up for steady calculation, one can set up unsteady simulation by two options:

1. **-dt timestepsize**

Example flow mesh -dt 0.001 will create set up with unsteady model with constant time step of 0.001 seconds.

2. **-vdt timestepsize**

This creates unsteady simulation with variable timestep and maximum stepsize set to specified timestep. Example flow mesh -vdt 0.001 will set up with max timestep of 0.001.

10. Example

solver.sh -lc vof mesh -vel 7.5 -forces -wd -gr -en -kw -dt 0.0001 -lc pf o.txt -lc iterate 100 -lc export-ensight result

This command sets up two phase VOF model, sets to transient mode, sets up velocity inlet to 7.5m/sec, exports forces on walls, includes gravity, wall distance, K Omega and energy models. It then creates output.stree and runs 100 times steps and exports results for post processing with file named result.case

All in one command.

3. Multiple wildkatze commands from console

If the number of commands are not too many it is easier to just pass them to program via console command. This could be achieved by:

solver.sh -lc command1 -lc command2 -lc command3

This is same as **solver.sh -f batch.txt**, Where batch.txt consists of lines command1, command2 and command3.

5. Two phase VOF

Given mesh in **mesh.info.bmsh** and **mesh.bmsh** files, we can set the basic two phase VOF model

solver.sh -lc vof mesh -lc pf o.txt

This sets up two phase (**prim-phase** and **second-phase**) VOF model with interface tracking is calculated for **prim-phase**.

6. Polyurethane Model

Two phase Polyurethane model is similarly set up as

solver.sh -lc kpoly mesh -lc pf o.txt

8. Adding Other Models

One can add other models to the basic set up by adding argument to the flow/vof/kpoly command. For example to set up flow with wall distance model, k-epsilon and energy.

solver.sh -lc flow mesh -wd -ke -en -lc pf o.txt

Here flag **-wd** adds **WallDistanceModel**, **-ke** adds **KepsilonTurbulenceModel** and **-en** adds **EnergyModel**.

The flag for other models are as follows:

-gr	GravityModel
-cmp	CompressibleFlowModel
-bsq	BoussinesqModel
-kw	KomegaTurbulenceModel
-sa	SATurbulenceModel
-sles	LesSubgridModel

Conjugate Heat Transfer



11. Conjugate Heat Transfer

Simplest way to set up CHT simulation is to use:

```
solver.sh -lc chtflow meshfile -lc pf o.txt
```

This command sets up two phases one for fluid and one for solids, rest of the options for **chtflow** are same as of **flow**. That means

```
solver.sh -lc chtflow mesh -kw -lc pf o.txt
```

Sets up CHT with k-omega turbulence model, for example.

One can use configuration file to enhance the set for CHT flow by using a configuration file. Configuration file could be specified to chflow command as:

```
solver -cl chtflow mesh -conf  
configurationFileName -lc pf o.txt
```

12.1. Configuration File

As we saw so far that we can fine tune the set up by specifying the options on command prompt. This becomes tedious as more and more options to the command needs to be added. To avoid this situation configuration file is introduced.

Configuration file is used to specialize or enhance the command prompt set up.

Creating basic configuration file

```
solver -lc info2conf meshfile
```

This creates user a basic setup for a CHT simulation where flow and solid regions are grouped into two region-sets.

This could be made more clear by inspecting one of the configuration file produced by the solver.

Example:

Given a **test.info.bmsh** file with

1. two fluid regions: **water1, water2**
2. four solid regions: **endcapMinusX, endcapPlusX, shell, tubesheet**

Now the command:

```
solver -lc info2conf test
```

Produces a configuration file:

The configuration file as seen from the example is set of commands or options passed to the solver. These options solver checks in order to produce a default set up file for simulation user wishes to perform.

The configuration file command/options has form:

```
( type-id command/option argument1 argument2 .... )
```

In current version type-id could be either 0 or 1. Type-id of 0 specifies a comment and 1 specifies a command or option that solver shall take care of while creating a set up or simulation tree (.stree) file.

We can inspect the configuration file produced:

```
( 1 region-set fluid-regions 2 water1 water2 )  
( 1 region-set solid-regions 4 endcapMinusX endcapPlusX shell tubesheet )
```

region-set Name Number-of-Region RegionNames creates a region-set. Here two region sets are created, one for flow and one for solids.

User can add more region sets here by adding more options. For example

```
( 1 region-set tubes 1 tubesheet )
```

Adds one more **region-set** with name **tubes** for one region **tubesheet**.

```
( 1 phase flow-phase )  
( 1 phase fluid-phase )  
( 1 phase solid-phase )
```

Adds three phases **flow-phase**, **fluid-phase** and **solid-phase**. In this simulation flow-phase properties are dependent on fluid-phase and solid-phase. Fluid-phase properties are tied to region-set fluid-regions and solid-phase properties are tied to solid-regions.

To learn more please check **Phases To Region Sets (phase2regionsets)** to see how Phase properties can vary based on types of regions.

```
( 1 phase-set flow-set 1 flow-phase )
```

This defines a phase-set with flow-phase. The ConjugatedFlowModel then needs to know which regions of the fluids and solids be getting properties from which phases. This is done as follows:

Phases To Region Sets

The flow-phase in this case needs to be informed of a mapping that tells which region set takes it's properties from which Phase. This Phase may in turn gets it's properties from other dependencies (Flow region might be multiphase for example). This is defined in configuration file as:

```
( 1 phase2regionsets flow-phase  
2  
fluid-phase fluid-regions  
solid-phase solid-regions  
)
```

Continued to next page...

```
( 0 Wildkatze Configuration )  
  
( 0 region sets )  
( 1 region-set fluid-regions 2 water1 water2 )  
( 1 region-set solid-regions 4 endcapMinusX endcapPlusX shell tubesheet )  
  
( 0 phases )  
( 1 phase flow-phase )  
( 1 phase fluid-phase )  
( 1 phase solid-phase )  
( 1 phase-set flow-set 1 flow-phase )  
( 1 phase2regionsets flow-phase  
2  
fluid-phase fluid-regions  
solid-phase solid-regions  
)  
( 1 phase-material fluid-phase Air 1.225 1.785E-5 1006.43 0.0242 28.966 )  
( 1 phase-material solid-phase Air 2700.0 1.0E-20 896.0 250.0 1.0 )
```


Conjugate Heat Transfer



Phases To Region Sets

Phase2regionsets is configuration command to relate **flow-phase** to region to phase connection. Now in above given example **flow-phase** takes properties from 2 connections. **Fluid-phase** specifies properties for **fluid-regions** and **solid-phase** provides properties for **solid-regions**.

In the situation user wishes to specifies different properties for other solids, he/she shall add more connections to **phase2regionsets**. For example we created region set called **tubesheet** earlier, if we had a phase called **phase-tubesheet**, then we can inform the solver of three different phase zones in flow-phase as:

(see next column)

```
( 1 phase2regionsets    flow-phase
    3
    fluid-phase          fluid-regions
    solid-phase          solid-regions
    phase-tubesheet     tubesheet
)
```

Now only thing remaining is to provide material properties for the phases (fluid-phase, solid-phase and phase-tubesheet etc).

This is done by phase-material command in configuration file. The default configuration example provides us:

```
( 1 phase-material fluid-phase Air    1.225  1.785E-5  1006.43  0.0242  28.966 )
( 1 phase-material solid-phase Air   2700.0  1.0E-20   896.0   250.0   1.0 )
```

The command phase-material specifies material for fluid and solid-phase. The keyword Air is telling the solver to base the material properties on Air from data base. Once this is done, user has to provide a set of values to be used by the simulation. The order is given as:

Density viscosity specific-heat thermal-conductivity and molecular-weight.

For solid regions some of the properties make no sense but they still need to be provided in the order mentioned. The values user provide would be ignored by the solver when not used. For example 1.0E-20 value that is provided for viscosity is ignored for solids.

12.2. Configuration File Other Options

Defining Grid-Grid Interfaces

```
( 1 file2ggi intersections_out.txt )
```

This adds grid-grid interfaces whose definition is provided by file intersections_out.txt.

intersections_out.txt is the file created by solver by using command **auto-interfaces** on Wildkatze console.

Options/Settings for Model

```
( 1 model-setting-real ConjugatedFlowModel
flow-phase energy-urf 0.999 )
```

This sets up **energy-urf** for **ConjugatedFlowModel** to 0.999 for **flow-phase**. User can override default options this way.

Boundary Condition for Model

```
( 1 model-setting-real-bc ConjugatedFlowModel
flow-phase water1 inlet7 temperature 500 )
```

This sets option for region (water1 here) and it's boundary (inlet7 here). In this case it sets **temperature** to 500K.

Example Configuration File With 2 Sets of Solids

```
( 0 Wildkatze Configuration )

( 0 region sets )
( 1 region-set fluid-regions 2 water1 water2 )
( 1 region-set solid-regions 3 endcapMinusX endcapPlusX shell )
( 1 region-set solid-tubes 1 tubesheet )

( 0 phases )
( 1 phase flow-phase )
( 1 phase fluid-phase )
( 1 phase solid-phase )
( 1 phase tube-phase )

( 1 phase-set flow-set 1 flow-phase )

( 1 phase2regionsets    flow-phase
    3
    fluid-phase fluid-regions
    solid-phase solid-regions
    tube-phase solid-tubes
)

( 1 phase-material    fluid-phase Air    1027  0.0003645  4195  0.668011  28.966 )
( 1 phase-material    solid-phase Air    7850  1E-20    486  52        1.0 )
( 1 phase-material    tube-phase Air    7832  1.0E-20   434  63.9     1.0 )

( 1 file2ggi intersections_out.txt )

( 1 model-setting-real-bc ConjugatedFlowModel flow-phase water1 inlet7 temperature 500 )
( 1 model-setting-real-bc ConjugatedFlowModel flow-phase water1 inlet7 velocity 1.0 )
( 1 model-setting-real-bc ConjugatedFlowModel flow-phase water2 inlet10 velocity 0.5 )

( 1 model-setting-real ConjugatedFlowModel flow-phase maximum-temperature 600 )
( 1 model-setting-real ConjugatedFlowModel flow-phase energy-urf 0.999 )
( 1 model-setting-real ConjugatedFlowModel flow-phase energy-convection-urf 0.95 )
( 1 model-setting-real ConjugatedFlowModel flow-phase energy-explicit-urf 0.99 )
( 1 model-setting-real ConjugatedFlowModel flow-phase energy-solid-explicit-urf 0.8 )
```