ANSYS Fluent (Fluid Dynamics)

- Getting Started
  - ANSYS Fluent Parallel Execution (Batch Mode)
  - ANSYS Fluent Journal Files
  - Translation of ANSYS Fluent UDF's (User-defined Functions)

ANSYS Fluent is a general purpose Computational Fluid Dynamics (CFD) code. ANSYS Fluent is part of the ANSYS software portfolio since 2007 (formerly owned by Fluent, Inc.). As a general purpose CFD code ANSYS Fluent provides a wide variety of physical models for turbulent flows, acoustics, Eulerian and Lagrangian multiphase flow modeling, radiation, combustion and chemical reactions, heat and mass transfer including CHT (conjugate heat transfer in solid domains). Since ANSYS Fluent R19.2 some basic expression language capabilities are provided. But most of the time advanced user customization requires the programming of a user-defined function (UDF) in C language.

Further information about ANSYS Fluent, licensing of the ANSYS software and related terms of software usage at LRZ, the ANSYS mailing list, access to the ANSYS software documentation and LRZ user support can be found on the main ANSYS documentation page.

Getting Started

Before the user tries to start ANSYS Fluent on a LRZ HPC system, the user should follow the steps as outlined in the paragraph "6. SSH User Environment Settings". ANSYS Fluent in the version 19.2 and later is known not to run in distributed parallel mode on LRZ Linux Clusters or SuperMUC-NG without properly generated passphrase-free SSH RSA keys.

Once you are logged into one of the LRZ cluster systems, you can check the availability (i.e. installation) of ANSYS Fluent software by:

```
> module avail fluent
```

Load the preferred ANSYS Fluent version environment module, e.g.:

```
> module load fluent/2020.R1
```

In contrary to ANSYS CFX the ANSYS Fluent software does not consist of a number of separate applications for preprocessing/solver/postprocessing purposes. Moreover ANSYS Fluent is designed as a single-window monolithic application which embeds all of these tasks in a single-window GUI and is even providing meshing capabilities (formerly known as TGRID or Fluent Meshing).

One can use ANSYS Fluent in interactive GUI mode for the only purpose of Serial pre- and/or postprocessing on the Login Nodes (Linux: SSH Option ".Y" or X11-Forwarding; Windows: using PuTTY and XMing for X11-forwarding). This interactive usage is mainly intended for making quick simulation setup changes, which require GUI access. And since ANSYS Fluent is loading the full mesh into the login nodes memory, this approach is only applicable to comparable small cases. It is NOT permitted to run computationally intensive ANSYS Fluent simulation runs or serial/parallel postprocessing sessions with large memory consumption on Login Nodes. Alternatively ANSYS Fluent can be run, e.g. for memory intensive postprocessing and OpenGL acceleration of the GUI on the Remote Visualization Systems.

The so-called ANSYS Fluent launcher is started by:

```
> fluent
```

where you have to specify, whether you intend to work:

- on a 2-dimensional or a 3-dimensional case
- in single or double precision
- in meshing mode
- in serial or parallel mode (on Login Nodes of the LRZ cluster systems only Serial mode is permitted)

It is not permitted to run computationally intensive ANSYS Fluent simulations on front-end Login Nodes or the Remote Visualization Systems in order not to disturb other LRZ users. A similar mode of operation like the ANSYS CFX Solver Manager monitoring mode for a still running parallel task on a cluster system or for an "a posteriori" monitoring analysis for a finished simulation run is unfortunately not existing for ANSYS Fluent. I.e. a graphical visualization of solver monitors has to be realized by the user outside of the ANSYS Fluent environment, e.g. by Python scripting, MS Excel or similar based on the monitor files and captured ANSYS Fluent output.

But it might be of interest, that ANSYS Fluent can be run in parallel mode for the use of the ANSYS Fluent build-in postprocessing. This is permitted on the LRZ Remote Visualization Systems, but the user has to apply necessary precaution and care:

- not to disturb other LRZ users of the visualization system
• to apply appropriate and moderate level of parallelization in the postprocessing session, which is usually substantially less parallel than the computation run
• not to overload the visualization system with ANSYS Fluent processes, i.e. to apply a level of parallelization which uses less Fluent processes than cores in the visualization system (Make use of "cat /proc/cpuinfo" to find out the required information about the systems architecture).

ANSYS Fluent Parallel Execution (Batch Mode)

All parallel ANSYS Fluent simulations on LRZ Linux Clusters and SuperMUC-NG are submitted as non-interactive batch jobs to the appropriate scheduling systems (SLURM) into the different pre-defined parallel execution queues. Further information about the batch queuing systems and the queue definitions, capabilities and limitations can be found on the documentation pages of the corresponding HPC system (LinuxCluster, SuperMUC-NG).

For job submission to a batch queuing system a corresponding small shell script needs to be provided, which contains:

• Batch queuing system specific commands for the job resource definition
• Module command to load the ANSYS Fluent environment module
• Start command for parallel execution of fluent with all appropriate command line parameters
• Reference to a small ANSYS Fluent journal file (*.jou), which is used to control the execution of ANSYS Fluent with the provided CAS file, since ANSYS Fluent CAS files do not contain a solver control section.

The intended syntax and available command line options for the invocation of the fluent command can be found out by:

> fluent -help

The configuration of the parallel cluster partition (list of node names and corresponding number of cores) is provided to the fluent command from the batch queuing system (SLURM) by the provision of an automatically generated environment variable $FLUHOSTS, based on the information provided by the cluster user in the job resource definition. Furthermore the environment variables $SLURM_NTASKS or $LOADL_TOTAL_TASKS are predefined by the used cluster system and batch queuing system as well and the information is just piped through to the fluent command as description of the cluster partition to be used for the parallel ANSYS Fluent simulation run.

Furthermore we recommend to LRZ cluster users to write for longer simulation runs regular backup files, which can be used as the basis for a job restart in case of machine or job failure. A good practice for a 48 hour ANSYS Fluent simulation (max. time limit) would be to write CAS/DAT files every 6 or 12 hours (to be specified in ANSYS Fluent under: Solution Calculation Activities Autosave Every Iterations). Further information you can find in the ANSYS documentation in the chapter "ANSYS Fluent Users Guide, Part II: Solution Mode; Chapter 40.16.1. Autosave Dialog Box").

Unfortunately ANSYS Fluent is not providing capabilities to stop a simulation run based on an elapsed wall clock time criterion. Furthermore it is unfortunately not possible to stop an ANSYS Fluent batch mode simulation run without loss of information by some whatever input from an external source (only by using "scancel" or "f cancel" commands of the respectively used batch queuing systems). Consequently it is due to the software user to have a good estimate for the overall runtime of the simulation task and to undertake necessary precautions to write regularly the required restart/backup information. Please plan for the setting of the maximum number of iterations/timesteps in your simulation enough time buffer for the writing of output case and data files, which can be a time consuming task depending on your application.

Caution: Changes in the ANSYS Fluent Parallel Start vor all Releases >=R19.2

In the past for ANSYS Releases < R19.2 it was necessary to heavily modify the ANSYS Fluent parallel start-up methods in order to make them run under SLURM and LoadLeveler on the LRZ cluster systems. Consequently for these older ANSYS Releases (e.g. 18.2, 19.0 and 19.1) only the "-mpi=intel" flag has to be specified. No information is required for the cluster interconnect, i.e. the "-pib.infinipath" or "-pib.dapl" flags should not be specified since this information is hard-coded in the corresponding ANSYS Fluent start-up scripts on the corresponding cluster systems.

ANSYS Fluent R19.2 and later versions:

For ANSYS Fluent Releases R19.2 and later a more flexible way of parallel ANSYS Fluent start-up on the different LRZ cluster systems has been found, which requires less intervention and modifications of the ANSYS Fluent start-up scripts and MPI wrappers. Consequently the information about the interconnect network (MPI fabric) of the used LRZ cluster system has to be provided to ANSYS Fluent by the user submitting the appropriate command line flags:

<table>
<thead>
<tr>
<th>LRZ Linux Cluster</th>
<th>SLURM Queue</th>
<th>Cluster Owner</th>
<th>ANSYS Fluent Command Line Options</th>
<th>ANSYS Fluent Command Line Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoolMUC-2</td>
<td>cm2_tiny, cm2_std, cm2_large</td>
<td>LRZ</td>
<td>-mpi=intel -pib.ofi</td>
<td>-mpi=intel -pib.ofi</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>{ -mpi=ibmmpi -pib.dapl }</td>
<td>{ -mpi=ibmmpi -pib.dapl }</td>
</tr>
<tr>
<td>CoolMUC-3</td>
<td>mpp3</td>
<td>LRZ</td>
<td>-mpi=intel -pib.infinipath</td>
<td>-mpi=intel -pib.ofi</td>
</tr>
<tr>
<td>TUM_Aer</td>
<td>tum_aer_batch</td>
<td>TUM Aerodynamic</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>HTRP</td>
<td>htrp_batch</td>
<td>FRM-II</td>
<td>-mpi=intel -pib.infinipath</td>
<td>-mpi=intel -pib.ofi</td>
</tr>
<tr>
<td>HTTP</td>
<td>httf_batch, httf_skylake</td>
<td>TUM LTF</td>
<td>-mpi=intel -pib.infinipath</td>
<td>-mpi=intel -pib.ofi</td>
</tr>
</tbody>
</table>
### CoolMUC-2 : ANSYS Fluent Job Submission on LRZ Linux Clusters running SLES15 using SLURM

Only the newer versions of ANSYS Fluent ranging from 2019.R1 up to 2020.R1 are compatible with the new CM2 Linux Cluster operating system SLES15 in combination with the INTEL MPI 2018. In the following an example of a job submission batch script for ANSYS Fluent 2020.R1 on CoolMUC-2 (SLURM queue = cm2_tiny) in the batch queuing system SLURM is provided:

```bash
#!/bin/bash
#SBATCH -o ./myjob.%j.%N.out
#SBATCH -D ./
#SBATCH -J fluent_cm2_tiny
#SBATCH --clusters=cm2
#SBATCH --partition=cm2_tiny
#SBATCH --qos=cm2_tiny
#SBATCH --get-user-env
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=28
#SBATCH --mail-type=NONE
#SBATCH --mail-user=Max.Mustermann@lrz.de
#SBATCH --export=NONE
#SBATCH --time=0:10:00
----------------------------------------------------
module load slurm_setup
module avail fluent
module load fluent/2020.R1
module list
# cat /proc/cpuinfo
echo "================================================================================ Fluent Start ================================================================================="
# For later check of the correctness of the supplied ANSYS Fluent command it is echoed to stdout,
# so that it can be reviewed afterwards in the job file:
echo fluent 3ddp -pcheck -mpi=intel -pib.ofi -t$SLURM_NTASKS -ssh -cnf=$FLUHOSTS -gu -driver null -i
Static_Mixer_run_Fluent.jou
fluent 3ddp -pcheck -mpi=intel -pib.ofi -t$SLURM_NTASKS -ssh -cnf=$FLUHOSTS -gu -driver null -i
Static_Mixer_run_Fluent.jou > Static_Mixer_run_Fluent.out
# Please do not forget to insert here your own JOU and OUT file with its correct name!
echo "================================================================================ Fluent End ================================================================================="
```

For earlier versions of ANSYS Fluent 19.1 and 19.2 the only found possibility to run this software on the renewed CM2 Linux Cluster is so far the use of IBM MPI (Platform MPI) with infiniband interfaces. In the following an example of a job submission batch script for ANSYS Fluent 19.2 on CoolMUC-2 (SLURM queue = cm2_tiny) in the batch queuing system SLURM is provided. Since these older versions of ANSYS Fluent 19.1 and 19.2 will be retired rather soon, please seriously and urgently consider migration to a contemporary version of ANSYS Fluent, like e.g. version 2020.R1. In case of encountered issues in the ANSYS Fluent software under Intel MPI this launch of ANSYS Fluent using IBM MPI can eventually provide an alternative. The ANSYS Fluent performance using IBM MPI can slightly, but not substantially be affected.
Assumed that the above SLURM script example has been saved under the filename "fluent_cm2_tiny.sh", the SLURM batch job has to be submitted by issuing the following command on one of the Linux Cluster login nodes:

```
sbatch fluent_cm2_tiny.sh
```

**Warning:** Do NOT use additionally `mpirun`, `mpiexec` or any `srun` command to start the parallel processes. This is done by a MPI wrapper by the fluent startup script in the background.

**CoolMUC-3 / IvyMUC : ANSYS Fluent Job Submission on LRZ Linux Clusters running SLES12 using SLURM**

In the following an example of a job submission batch script for ANSYS Fluent on CoolMUC-3 (SLURM queue = mpp3) in the batch queueing system SLURM is provided:

```
#!/bin/bash
#SBATCH -o ./myjob.%j.%N.out
#SBATCH -D ./
#SBATCH -J fluent_mpp3_slurm
#SBATCH --clusters=mpp3
#SBATCH --get-user-env
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=64
#SBATCH --mail-type=NONE
#SBATCH --mail-user=Max.Mustermann@lrz.de
#SBATCH --export=NONE
#SBATCH --time=0:10:00
#----------------------------------------------------
module load slurm_setup
module avail fluent
module load fluent/2020.R1
module list
# cat /proc/cpuinfo
echo "================================================================================ Fluent Start =========================================================================="
# For later check of the correctness of the supplied ANSYS Fluent command it is echoed to stdout,
# so that it can be reviewed afterwards in the job file:
echo fluent 3ddp -mpi=intel -pib.ofi -t$SLURM_NTASKS -ssh -cnf=$FLUHOSTS -gu -driver null -i Static_Mixer_run_Fluent.jou
fluent 3ddp -mpi=intel -pib.ofi -t$SLURM_NTASKS -ssh -cnf=$FLUHOSTS -gu -driver null -i Static_Mixer_run_Fluent.jou > Static_Mixer_run_Fluent.out
# Please do not forget to insert here your own JOU and OUT file with its correct name!
echo "================================================================================ Fluent End =========================================================================="
```
Assumed that the above SLURM script has been saved under the filename "fluent_mpp3_slurm.sh", the SLURM batch job has to be submitted by issuing the following command on one of the Linux Cluster login nodes:

```
sbatch fluent_mpp3_slurm.sh
```

**Warning:** Do NOT use additionally `mpirun`, `mpiexec` or any `srun` command to start the parallel processes. This is done by a MPI wrapper by the `fluent` startup script in the background. Also, do not try to change the default Intel MPI to any other MPI version to run ANSYS Fluent in parallel. On the LRZ cluster systems only the usage of Intel MPI is supported and known to work properly with ANSYS Fluent.

**SuperMUC-NG : ANSYS Fluent Job Submission on SNG running SLES12 using SLURM**

In the following an example of a job submission batch script for ANSYS Fluent on SuperMUC-NG (Login node: skx.supermuc.lrz.de, SLURM partition = test) in the batch queuing system SLURM is provided. Please note that supported ANSYS versions on SuperMUC-NG are ANSYS 19.2 or later. At this time **ANSYS 2020.R1** is the default version.

```bash
#!/bin/bash
#SBATCH -o ./myjob.%j.%N.out
#SBATCH -D ./
#SBATCH -J fluent_test
#SBATCH --partition=test
#SBATCH --get-user-env
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=48
#SBATCH --mail-type=END
#SBATCH --mail-user=Max.Mustermann@lrz.de
#SBATCH --no-requeue
#SBATCH --export=NONE
#SBATCH --time=0:05:00
#SBATCH --account=<Your_own_project>

module load slurm_setup
module av fluent
module load fluent/2020.R1
module list

# cat /proc/cpuinfo
echo "Fluent Start
-----------------------------------------------"
# For later check of the correctness of the supplied ANSYS Fluent command it is echoed to stdout, # so that it can be reviewed afterwards in the job file:
echo fluent 3ddp -mpi=intel -pib.ofi -t$SLURM_NTASKS -ssh -cnf=$FLUHOSTS -g -i Static_Mixer_run_Fluent.jou
echo fluent 3ddp -mpi=intel -pib.ofi -t$SLURM_NTASKS -ssh -cnf=$FLUHOSTS -g -i Static_Mixer_run_Fluent.jou > Static_Mixer_run_Fluent.out

# Please do not forget to insert here your own JOU and OUT file with their correct names!
echo "Fluent End
-----------------------------------------------"
```

Assumed that the above SLURM script has been saved under the filename "fluent_sng_slurm.sh", the SLURM batch job has to be submitted by issuing the following command on one of the SuperMUC-NG login nodes:

```
sbatch fluent_sng_slurm.sh
```
ANSYS Fluent Journal Files

In contrary to ANSYS CFX the information provided in an ANSYS Fluent CAS file is either not sufficient or not properly used by ANSYS Fluent in order to run a simulation on a parallel cluster system by just submitting the CAS file. For a proper start and solver control of a parallel ANSYS Fluent simulation run in batch mode it is required to provide an at least minimal so-called journal file (file extension *.jou), as you can see in the above SLURM script examples by the command line option "-i Static_Mixer_run_Fluent.jou".

Such a basic journal file contains a number of so-called TUI commands to ANSYS Fluent (TUI = Text User Interface). Details on the TUI command language of ANSYS Fluent can be found in the ANSYS Fluent documentation, Part II: Solution Mode; Chapter 2: Text User Interface (TUI).

As a small example of such a basic ANSYS Fluent journal file the "Static_Mixer_run_Fluent.jou" from the above SLURM batch queuing scripts is provided here. This small journal file does the following:

- read CAS file for the Static_Mixer.cas simulation
- write an ANSYS Fluent settings file
- do hybrid initialization of the case
- print time stamps of wall clock time at the start and end of solver iterations, e.g. for performance analysis
- do 100 steady-state iterations of the ANSYS Fluent solver, pseudo-transient solution method
- write a pair of CAS/DAT results files
- write reports of parallel time usage, system status and the ANSYS Fluent simulation summary

Please note, that for carrying out the same simulation as a transient simulation the journal file would require modifications for the timestep integration. Same applies, if it is e.g. intended to initialize a run from a previously computed interpolation file or to continue a run from a previously obtained results file pair of CAS/DAT files.

ANSYS Fluent Check-pointing and Emergency Exit in Batch Mode

Normally a user does not has direct SSH access to the compute nodes of a Linux cluster where ANSYS Fluent is executed under the control of a batch queueing system like e.g. SLURM. Therefore the following approach is provided to control ANSYS Fluent to a certain extend from the Linux cluster login node, while it is executed on a certain Linux cluster partition. So it might be desirable to:

- Check-pointing the simulation:
  The user might want to make ANSYS Fluent to write a pair of CAS/DAT files of the current intermediate simulation result on the next possible convenience, i.e. by reaching the next steady-state iteration or by finalizing the currently computed timestep.

- Immediate stop or emergency exit from the current simulation run:
  The user might want to make ANSYS Fluent stop the current simulation on the next possible convenience and by writing the last computed CFD result to a pair of CAS/DAT files for postprocessing or simulation restart. This could e.g. be desirable, if for some reason the time in the queue is about to expire, but in accordance with the specified journal file ANSYS Fluent would normally not come to a normal end of execution with results file writing. Without the following possibility ANSYS Fluent would just be canceled by the SLURM manager with total loss of the last computed result (i.e. no CAS/DAT file being written). With the following possibility this can be circumvented.

As in the example journal file above, the following lines need to be included, in order to make ANSYS Fluent check-pointing and emergency exit functionality available:
; Journal file snippet enabling check-pointing and emergency exit functionality
;
; Scheme commands to specify the check pointing and emergency exit commands:
(set! checkpoint/check-filename "./check-fluent")
(set! checkpoint/exit-filename "./exit-fluent")
;
;....

With the above two scheme language definitions two filenames are getting declared, which afterwards during ANSYS Fluent runtime can be used to initiate either checkpointing (i.e. writing gzip'ed CAS/DAT files of the intermediate result) or an emergency exit from the current simulation. Once the above declarations had been included in the controlling journal file prior to SLURM submission, than from a Linux terminal window on the Linux cluster login node and being positioned in the ANSYS Fluent working directory checkpointing/emergency exit can be initiated with the following Linux commands:

cd ~/<working-dir>
# Linux command for initiating the ANSYS Fluent check-pointing:
touch ./check-fluent
#
# Linux command to initiate an emergency exit of ANSYS Fluent from the currently running simulation:
touch ./exit-fluent

If the gzip'ed pair of CAS/DAT files have been written, ANSYS Fluent automatically removes the created empty files from the filesystem, so that the user does not has to care about them. As an additional convenience, ANSYS Fluent creates in the case of an emergency exit a new journal file, which starts with reading in the latest created CAS/DAT files and contains the remaining part of the users journal file which up to this point has not been executed. This newly generated journal file can be used as a template for re-submission to SLURM and continuation of the simulation run.

ANSYS Fluent Journal Commands to flush large Linux I/O Buffer Caches

Under certain circumstances you might encounter in your output files (transcript) from ANSYS Fluent simulations warning messages of the following kind:

> WARNING: Rank:0 Machine mpp3-xxxxxxx has 66 % of RAM filled with file buffer caches. This can cause potential performance issues. Please use -cflush flag to flush the cache. (In case of any trouble with that, try the TUI/Scheme command '(flush-cache)').

This warning messages point you to the fact, that on at least one computational node of your assigned Linux cluster partition it was detected on ANSYS Fluent startup that larger amounts of the node memory are occupied by large Linux I/O buffer caches. So what does that mean and how you should react?

1. In principle this should not be the case, since the SLURM prolog of the batch queueing system should provide to you an almost clean set of computational nodes with purged Linux I/O buffer caches. But it can occur, that this SLURM prolog fails in entirely flushing the caches. If that is observed by you on a more regular basis, please file a LRZ service request and provide the ANSYS Fluent output file (transcript) to the LRZ support staff.
2. The warning message from ANSYS Fluent is not a very serious concern and your intended simulation run should continue without further issues. But it can become a concern from a simulation performance point of view. Memory occupation by larger remaining Linux I/O buffer caches can (and most likely will) lead to an unbalanced distribution of ANSYS Fluent's memory allocations with respect to the launched ANSYS Fluent tasks on the dual socket compute node, i.e. a number of ANSYS Fluent tasks need to permanently access their corresponding data not from the processors own local memory but from the memory of the other neighbouring processor on the dual-socket compute node, thereby encountering less efficient memory access. This potentially can lead to ANSYS Fluent performance degradation in the order of up to 10-12% based on the measurable increase in simulation run time. For a 48 hours simulation run 10-12% increase in simulation time are about 5 hours plus.
3. Consequently, if you permanently observe these warning messages, you potentially would like to get the following command lines included in your ANSYS Fluent journal file in order to get the large Linux I/O buffers flushed by yourself prior to the startup of ANSYS Fluent for your intended simulation. Please do not include these lines by default, because they delay the application startup a bit depending on the amount of available system memory on the nodes. Start-up delay can be up to 1-3 minutes in comparison with normal startup procedure of ANSYS Fluent.
4. Do not try to use the above mentioned recommendation to include the option "-cflush" on the ANSYS Fluent command line in your SLURM script, since this does not work for the Linux cluster diskless compute nodes. Instead use the following lines in your simulation journal file:

; Following commands reduce the pending Linux I/O buffers to a minimum.
; Large I/O buffers can potentially have a performance impact on ANSYS Fluent
; due to non-local memory allocation in dual-processor systems of the compute nodes.
(rpsetvar 'cache-flush/target/reduce-by-mb 12288)
(flush-cache)
;
... followed by the remaining code lines from your normal ANSYS Fluent journal file...
;
/file/set-batch-options no yes yes no
; Option to disable HDF5-based CFF file format (legacy CAS/DAT files):
/file/cff-files? no
/file/read-case ".//Static_Mixer.cas"
;
...
The "magic number" 12288 (Mb) in the 1st command line of the above ANSYS Fluent journal file snippet is the specification of the amount of system memory, which cannot be purged due to resident Linux OS files in the memory of the diskless compute nodes. This is mainly to be addressed to buffers of the GPFS parallel file system and the OS system image. Since the size of the preloaded OS system image can be subject to slight changes over time, this is rather an experimental value than just a fixed number. If you observe, that the flushing process is not carried out successfully (by error messages in the ANSYS Fluent transcript), try to increase this number. The given number essentially means, that on a compute node with 64 Gb ANSYS Fluent tries to clean-up/flush about 64Gb - 12.288Gb = 51.772Gb of memory.

Creation of Graphics Output from ANSYS Fluent in Batch Mode

Normally in batch mode ANSYS Fluent is only producing a transcript of the simulation run, i.e. a tabulated list of computed residuals and monitor data together with other text-based information about the current simulation run, and finally the specified output files (CAS/DAT files, backup files, monitor files and reports). But at least it would be desirable, to have from a batch simulation run a PNG or JPEG graphics file with the graphical representation of the convergence history of ANSYS Fluent available. This can be realized by doing the following.

First of all ANSYS Fluent needs to be started in the SLURM script with the correct command line options and by including a so-called NULL graphics driver:

```
fluent 3ddp -mpi=intel -t$LOADL_TOTAL_TASKS -cnf=$FLUHOSTS -gu -driver null -i Static_Mixer_run_Fluent.jou > Static_Mixer_run_Fluent.out
```

Here the options "-gu -driver null" allow the generation and export of graphics files from an ANSYS Fluent simulation run in batch mode.

The user than has to organize this graphics file export by appropriate journal commands or by defined calculation activities. For example, the following journal commands can be used to write the convergence history of the ANSYS Fluent simulations as colored and black&white residual diagrams in two separate PNG files:

```
;.....
; write a PNG file of the residual history
;
/display/monitors/residual/plot? yes
/display/set-window 1
/display/set/picture/driver png
/display/save-picture residual_history.png
/display/set/picture/color-mode/mono-chrome
/display/set-window 1
/display/save-picture residual_history_bw.png
;
;.....
```

By similar journal file statements e.g. diagrams of monitor data or postprocessing graphics like contour plots, streamlines, isosurfaces, etc. being previously defined by the postprocessing functionality inside ANSYS Fluent can be exported from an ANSYS Fluent simulation in batch mode as well.

Translation of ANSYS Fluent UDF's (User-defined Functions)

For customization purposes ANSYS Fluent provides essentially the following two possibilities:

- Starting with the ANSYS Fluent 2019.R1 release (early beta in R19.2), the CFD solver is now supporting algebraic expressions, like ANSYS CFX and AIM Fluids users are used to already for a long time. Consequently, for many purposes, where users had to write a piece of a C subroutine in the past, the goal can now be achieved by just inserting a named algebraic expression in ANSYS Fluent's GUI, e.g. for specifying a turbulent velocity profile at an inlet cross-section of your geometry in accordance with the 1/7th power law.
- Nevertheless in many cases User-defined Functions (UDF's) are still rather common for ANSYS Fluent to customize build-in fluids solver capabilities and model functionalities. UDF's are user programmed subroutines, written in C language, which are hooked-up in the CFD setup to fulfill their purpose.

Prior to be able to run ANSYS Fluent for a CAS file depending upon a UDF, the UDF written in C language needs to be compiled. Best a UDF is compiled on a system with the same operating system (OS) and same processor architecture as the target compute cluster. In the case of LRZ Linux Clusters we recommend to compile UDF's either on the cluster login node or on the Remote Visualization Systems (RVS). It is not possible to compile a UDF in batch mode on a compute node of the Linux clusters or SuperMUC-NG, since compute nodes are diskless nodes with an as minimum boot image as possible, which is usually lacking a C programming environment (compiler, linker, libraries).

Furthermore, for parallel computation of the CAS file ANSYS Fluent needs to compile two versions of a UDF - the node and the host library subtrees from the provided UDF. But in order to accomplish this UDF compilation, the ANSYS Fluent solver does not necessarily to be run in parallel mode. Since approx. ANSYS Fluent 18.x (?) the solver does not know the "serial mode" anymore. Instead of what is called now the "serial mode" of ANSYS Fluent is in fact the general parallel mode with just one parallel task. You can see this by examining the output of the ANSYS Fluent solver if run in "serial mode" and you will observe, that the solver is started in parallel with the "-t1" option specified, i.e. with one parallel task and the remaining usual infrastructure of a parallel run. Consequently, if ANSYS Fluent is started on e.g. the Linux Cluster login node, the CAS file with the UDF is loaded and the UDF is compiled, than both versions of the library subtree (node and host libraries) are automatically created (ANSYS Fluent GUI: Ribbon User-defined Functions Compiled Build). This would be the recommended way of compiling a UDF prior to submitting the CAS file for computation to SLURM (...and btw. an attempt to start ANSYS Fluent in real parallel mode with more than 1 parallel task on the Linux Cluster login node would be not successful, since this is
prevented by the configuration of the login nodes; parallel execution of programs is permitted on login nodes). For comparable large cases, where the memory of Linux Cluster login nodes could become a potential bottleneck for opening the CAS file in ANSYS Fluent, either of the two following ways are recommended:

1. Open a RVS session on one of the Remote Visualization Systems, which are equipped with a larger amount of memory and where it is permitted to run ANSYS Fluent in local parallel mode. If the CAS file is "small" enough to be opened on RVS, this method is recommended. The UDF libraries being compiled on RVS can be re-used for parallel run on the Linux Clusters.

2. Create a smaller CAS file, e.g. from a substantially coarsened mesh, and with the same identical setup and UDF. Eventually you have such a coarse mesh already from a previously carried out mesh independency study or other tests. Hook-up your UDF with that smaller case file and open it in ANSYS Fluent in serial mode on the Linux Cluster login node for compilation. The CAS file now should be small enough to carry out this procedure without running into node memory issues. The resulting UDF libraries now again can be re-used with the productive large mesh CAS file.

In case of trouble or need for help, please contact the LRZ Service Desk.