AquaEdit
AquaCross
AquaView
AquaTool
» Other

User manual

Document ID: TR-20000-583-5
Date: 01\03\2024

AquaSim version: 2.19

Aquastructures AS
Kjøpmannsgata 21
7013 Trondheim
Norway

Contents

1 Introduction

AquaSim is an analysis tool developed by Aquastructures AS. It uses the Finite Element Method (FEM) for calculation and simulation of structural response. The software is well suited for slender, lightweight- and large volume structures, flexible configurations and coupled systems exposed to environmental loads such as:

- waves
- currents
- wind
- impulse loads
- operational conditions
- resonance

This manual describes the functionality of the following post-processing tools:

- AVSpack
- PostProcFilter
- PostProcReduce
- Surface Extractor

These are programs suitable for processing AquaSim result files such as .AVS, .AVZ or .xml. They are run through the Command Prompt in Windows. The programs are found under the AquaSim installation folder on your computer.

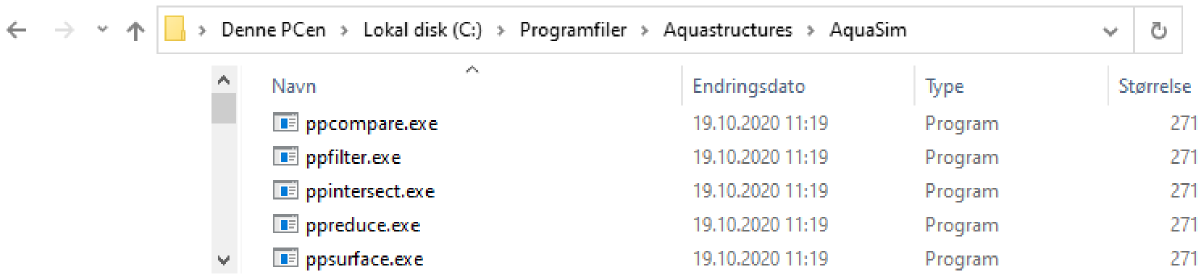

Figure 1 Location of the post-processing tools on your computer

2 Range

Throughout this document you will see references to RANGE or RANGES. A range is parsed as one or more numbers (usually for ID range of time steps) and can be inputted as:

1. A single number: 1 or 5, etc.
2. A list of numbers: 1,5,100, etc.
3. A range om numbers: 1-5 (all numbers starting from 1 through to 5, same as using a list of 1,2,3,4,5)
4. A combination of the above: 1,5-10,100,101 (expanded this would be: 1,5,6,7,8,9,10,100,101)

Using RANGE it is important not to have spaces between values.

When using post processing tools that understand node and element names (ppreduce) RANGE can include the names: -elements 1,"E 2",3-5,AnotherElement.

This would be node 1, 2 (named E2), 3, 4, 5, and (in this example) node 10, having been named in AquaEdit.

3 AVSPack (avspack.exe)

Combines AVS and XML files into (compressed) AVZ. Also includes extra files like terrain and shapes where applicable.

3.1 Example usage

```
avspack.exe –join file.avs file.xml
```

Packs .AVS and .XML file into an AVZ file with the same name as the given AVS file (albeit with .AVZ as the file extension).

Note: *normally, this command is invoked by the .bat-file exported by AquaEdit.*

Note: *.AVZ-files are ZIP compressed files with (primary) two files inside – model.avs and model.xml.*

4 PostProcFilter (ppfilter.exe)

This tool allows .AVS or .AVZ file to be split into multiple files, containing limited number of steps and/or stresses.

```
PostProcFilter
==============
Usage: ppfilter INFILE OUTFILE [OPTIONS] {stress} ...
Options:
  -steps RANGE        outputs the specified timesteps (use from-to step as RANGE,
eg: 10-15 will output timesteps 11 through 16.
  -split TIMESTEPS    Split the INFILE into a number of files each TIMESTEPS
timesteps long.
                      OUTFILE will be OUTFILE_STARTTIMESTEP.AVS.
  -nth NUMBER          Output every nth timestep
```

Note: *stresses are optional but can be specified to limit to only the given stresses.*

4.1 Example usage

Example 1, extracting specific steps

```
ppfilter.exe in.avs out.avs –steps 0-10
```

Outputs a file containing the 10 first timesteps.

Example 2, extracting specific stresses

```
ppfilter.exe in.avs out.avs Von_Mises_stress_MPa Net.Force_in_horizontal_twines
```

Outputs a file containing only Von_Mises_stress_MPa and Net.Force_in_horizontal_twines.

Note: *ppfilter only outputs .AVS. To combine the new .AVS- and the original .XML-file please refer to avspack.exe.*

Example 3, automatically splitting one file into several

```
ppfilter.exe in.avs out.avs -split 50
```

Using the -split command ppfilter will automatically split a file into chunks of timesteps, in the example 50.

Example 4, reducing number of timesteps in file

```
ppfilter.exe in.avs out.avs -nth 5
```

Using -nth ppfilter can reduce the number of timesteps in a file by only outputting every -nth timestep. A file with 500 steps will be reduced to 100 steps using -nth 5.

5 PostProcReduce (ppreduce.exe)

Outputs pure text (optionally CSV format) files for ease of processing.

```
PostProcReduce
================
Usage: ppreduce INFILE OUTFILE [OPTIONS] {stress} ...
Options:
  -mode TYPE          will set the default method for choosing element output. The
default is average, other values are MAX, MIN, ABSMAX, ABSMIN
  -useSeconds         will convert timesteps to real time (NB: not available for
all files)
  -tab                uses tab between sections instead of spaces
  -elements ID        will list only element ID (ID can also be multiple IDs e.g
1,5,8 or a range of IDs eg -elements 1-5,10-50. If omitted all elements will be
reported.
  -nodes ID        will list only node ID (ID can also be multiple IDs e.g 1,5,8
or a range of IDs eg -nodes 1-5,10-50. Must not contain spaces). Nodes can
additionally report x, y, and z positions.
  -steps RANGE        will list only the specified step range (See -elements for
how to specify range). If omitted all steps will be reported.
  -component          Values will instead be reported per component. -elements are
now component ids instead of elements (DEPRECATED).
  -components RANGE   will list only components with the given ID (ID can also be
multiple IDs e.g 1,5,8 or a range of IDs eg -components 1-5,10-50.
  -collate            Collates all chosen timesteps into a single line.
  -CSV                The output will be made as a comma separated list
  -XML filename       Optional XML file to load along with AVS files
  -useTagNames        If the INFILE is an AVZ file or -xml has been used to
specify an XML file to read information from this will display named nodes and
elements in the output. If this is not specified nodes and elements will always
show their ID
  -format FORMAT      The format of the decimal outputs, please refer to the
manual for further information. (use -format 0.000E00 for scientific notation.)
  -singleelement      Split the output for every element into its own file.
```

5.1 Options

The following options are a part of PostProcReduce.

-mode

When working with elements only one stress value is outputted, even though an element consists of readings from both nodes it is connected to (four for membranes). Using the –mode option the user can select how the readings should be interpreted. By default, the average of the two (or four) nodes are used.

-useSeconds

AquaSim, by default, works in (and outputs) time steps. The length (in seconds) of a time step is defined by several parameters including environmental wave period, current, and the number of time steps per wave.

By using –useSeconds ppreduce will output tables using seconds instead of time steps.

-tab

By default, the outputted text file is delimited by space characters. By using –tab a single tabulator character will be used between sections

-elements

To output stress data on elements use –elements RANGE. Where RANGE is one or more element IDs.

-nodes

Instead of elements data can be written out as nodes instead. Use –nodes RANGE. Nodes expect "stress types" of x, y, z position instead of actual stresses. See Example 5.2.

-steps

The outputted data can be limited using –steps RANGE.

Note: *-steps use 0 as the initial step, so the range 1-10 would be the first ten timesteps EXCLUDING the initial step. To include the initial timestep, start from 0. For clarification, timestep 0 is the original geometry as drawn in AquaEdit before any simulation has been run.*

-component

Using –component will write data from a whole component instead. The IDs given as –elements RANGE will be used as component numbers instead.

Note: *Future expected change to –component RANGE instead of using –elements for component IDs. Still available for 2021 release of AquaSim.*

-components RANGE

Using -components RANGE will return information about entire components, instead of just individual elements. For every element in a component, the stress will be found by using the output type specified by -mode to reduce element values into a single value for the component. In most cases either -type MIN or -type MAX is the most sensible modes.

Note: *this deprecates -component.*

-collate

Turns timestep information on components into a single line, using the collection method specified by -mode.

Example, without -collate:

```
Time [-]   EID [-]  Comp [-]  Von_Mises_stress [MPa]
       1         0         1                    0.00
       2         0         1                    0.80
       3         0         1                    0.60
```

Example, with -collate

```
Time [-]   EID [-]  Comp [-]  Von_Mises_stress [MPa]
       3         0         1                    0.80
```

***Note:** "Time" will return the last timestep that was specified in -steps RANGE. By using -type MAX the maximum found stress in the RANGE will be reported.*

-CSV

Using this will output comma separated values instead of tab.

-XML

Allows for specifying an XML file to load for additional supporting functionality. Currently this includes the ability to specify and view node and element names.

If the input file is an .AVZ-file, the .XML-file will be autodetected from inside the AVZ archive and automatically loaded.

-useTagNames

If an .XML-file has been specified, or ppreduce has found the .XML-file in an AVZ archive, node and element numbers can be displayed in the output file.

Example, print node 1 by ID, and node 2 by the given name ("E 2):

```
Ppreduce minimal01.avz minimal01.avz.txt -elements 1,"E 2"
Local_section_forces.Axial_force_[N]
```

Example, output from -useTagNames

```
Time [-]   EID [-]  Comp [-]  Axial_force [N][MAX]
       0         1         1                 0.000
       0       E 2         1                 0.000
       1         1         1                 0.000
       1       E 2         1                 0.000
```

-format

By default, numbers are outputted in decimal format. Using -format 0.000E00 will instead output numbers in the scientific format.

-singleelement

Each specified element will output its own file. Filenames will be named as “out”+number of element +”.txt”

5.2 Example usage

Example 1

Output the Von Mises stress and Axial Force on element 10. Note the default output Time (in timesteps) Element ID, Component ID, and each of the given stresses.

```
ppreduce.exe in.avs out.txt -elements 10 Von_Mises_stress_[MPa]
Local_section_forces.Axial_force_N
```

Output from example 1

```
Time [-]   EID [-]  Comp [-]  Von_Mises_stress [MPa]  Axial_force [N]
       1        10         1                    0.00             0.00
       2        10         1                    0.80        -12173.00
       3        10         1                    0.92        -10994.00
       4        10         1                    2.02        -18653.00
```

Example 2

Output the positions of nodes. Note the output Time (in steps), the Node (or Vertex) ID, and the positions (in meters).

```
ppreduce in.avs out.txt -nodes 20-21 x y z
```

Output from example 2

```
Time [-]   VID [-]         X         Y         Z
       1        20    -40.00    350.00   -100.00
       1        21    -40.00    360.00   -100.00
       2        20    -40.01    350.49   -104.07
       2        21    -40.01    360.47   -104.65
       3        20    -40.01    350.49   -104.07
       3        21    -40.01    360.47   -104.65
```

Example 3

Output the maximum Von_Mises_stress_MPa found in the components 1 and 2.

```
ppreduce.exe in.avs out.txt -mode max -elements 1,2 -component
Von_Mises_stress_[MPa]
```

Output from example 3

```
Time [-]  Comp [-]     - [-]  Von_Mises_stress [MPa]
       1         1         0                    0.00
       1         2         0                    0.00
       2         1         0                    1.26
       2         2         0                    1.07
       3         1         0                    1.41
```

Example 4

Same example as above, but this time with –CSV. This ease import into excel and other tools.

Output from example 4

```
Time [-],Comp [-],- [-],Von_Mises_stress [MPa]
1,1,0,0.00
1,2,0,0.00
2,1,0,1.26
2,2,0,1.07
3,1,0,1.41
3,2,0,3.75
4,1,0,2.82
```

Example 5 (advanced use)

Windows Batch file commands can be used to further simplify and efficiently gather data from multiple files.

```
FOR %%I in (*.AVZ) DO ppreduce %%I %%I.csv -mode max -components 6,7 -steps 1-55 -
collate -CSV Net.Axial_force_in_horizontal_twines_[N]
```

In this example we use windows batch file functionality to process many files at once. The FOR syntax takes a "globbing" pattern and the variable to store file names in.

In this case, the value we will use to store is %%I, and the globbing pattern is (*.AVZ) meaning all .AVZ-files in the folder. This can be further limited. See globbing examples.

After the variable name and the globbing pattern we find 'DO'. This is to signify that what comes after this will be repeated for each and every file found. In this example we use %%I as the input name and %%I.csv as the output name. So run01.AVZ will become run01.AVZ.csv for the output part of ppreduce.

Example 6 (globbing)

Filename patterns can be constructed with a variety of characters and wildcards (? and *).

The easiest globbing pattern is all filenames ending with a specific extension, like *.AVZ. We can expand on this by adding characters, like analysis*.AVZ. This will find all files that start with analysis and end with .AVZ. It is always a good idea to use the file extension, so non-AVZ (or AVS if you so choose) will not be attempted processed.

For more specific applications, '?' can be used in place of a single character (* will match any number of characters). E.g analysis1?.AVZ will match analysis10.avz through analysis19.avz, and also any other characters, like analysis1a.avz.

See more advanced globbing and batch scripting functionality at the end of this document.

6 Surface extractor (ppsurface.exe)

This chapter presents the functionality of Surface extractor.

```
Usage: ppsurface IN OUT [options]
  IN  Input filename
  OUT  Output filename
  Options:
    -delta
       Delta between each printed timestep
       Default: 1.0
    -grid
       The grid to extract, parameters: width length divx divy
    -help
       Prints this help text
       Default: false
    -hydro
       Prints the hydrodynamic pressure at the extracted locations
       Default: false
    -node
       The nodes to print
    -position
       The position to extract surface data for
    -seconds
       Print seconds instead of timesteps
       Default: false
    -start
       The time(step) to start at
       Default: 0.0
    -stop
       the time(step) to stop at (0 = end-of-file)
       Default: 0.0
```

-node

Giving a single node is compatible with the –grid option. The grid will follow the –node. The z output will be the z position of the node.

6.1 Output from Surface extractor

All output from ppsurface is in the following format:

Timestep (or seconds, if using –seconds), the x position measures, the y position measure, the z position of possible node (if using –position or –grid this will always be 0), overflow is Wave-Z, i.e. the total height of the wave above a given node. And Wave which is wave displacement from surface (z=0).

```
Timestep     X           Y           Z           Overflow    Wave
6.000000     3.000000    -3.000000   0.000000    0.000000    0.000000
6.000000     3.000000    0.000000    0.000000    0.000000    0.000000
6.000000     3.000000    3.000000    0.000000    0.000000    0.000000
7.000000     -3.000000   -3.000000   0.000000    0.017136    0.017136
7.000000     -3.000000   0.000000    0.000000    0.017136    0.017136
7.000000     -3.000000   3.000000    0.000000    0.017136    0.017136
7.000000     0.000000    -3.000000   0.000000    0.048031    0.048031
```

6.2 Example usage

Example 1

Position and wave height for node 10 through 15.

```
ppsurface in.avs out.txt -node 10-15
```

Example 2

Wave height at a given position.

```
ppsurface in.avs out.txt -position 0 100
```

Example 3

A grid of 200x200m, with 0.5m resolution.

```
ppsurface in.avs out.txt -grid 100 100 200 200
```

Example 4

Same grid as above, but with the center of the grid displaced.

```
ppsurface in.avs out.txt -grid 100 100 200 200 -position 100 100
```

Example 5

Same as example 1, but outputted using seconds and a time resolution of 0.25s.

```
ppsurface in.avs out.txt -node 10-15 -seconds -delta 0.25
```

7 Globbing and advanced automation

Globbing is an expression of looping over a set of files.

7.1 Simple example of globbing

Loop over all .AVZ-files and run ppfilter. The output file will be “(inputfile)_new.avs”. Each file will be processed on after the other, for all .AVS-files in the current folder.

```
FOR %%a in (*.AVZ) DO (
	ppfilter "%%a" "%%a_new.avs " -nth 10 -total 4000
)
```

Output commands, from simple example

```
ppfilter "run01.AVZ" "run01.AVZ_new.avs" -nth 10 -total 4000
ppfilter "run03.AVZ" "run03.AVZ_new.avs" -nth 10 -total 4000
```

7.2 Complex example of globbing

This example loops over all .AVZ- and .AVS-files. The output filename will be (input)_nth.avs. Because of the START in front of the ppfilter line, every command will be ran at the same time. One additional command window will be opened per run. Warning, if there are many files this can cause a lot of windows to open. This example also employs some measures to make filenames more convenient than in the above example.

```
@echo off
SETLOCAL EnableDelayedExpansion
FOR %%a in (*.AVZ;*.AVS) DO (
	set inFile=%%a%
	set outFile=!inFile:~0,-4!_nth.avs

	START ppfilter "!inFile!" "!outFile!" -nth 10 -total 4000
)
```

Output commands, from complex example

```
START ppfilter "run01.AVZ" "run01_nth.avs" -nth 10 -total 4000
START ppfilter "run03.AVZ" "run03_nth.avs" -nth 10 -total 4000
```

Tips and hints

The command can be «dry» run or debugged by placing ECHO in front of it. This cause the command to be printed to screen, but not executed.

```
ECHO START ppfilter "!inFile!" "!outFile!" -nth 10 -total 4000
```

8 Stresses

This chapter presents the entry-format of stresses in AquaSim.

Note: *this section is still under development.*

8.1 General

```
Acceleration.Acc_X_[m/s2]
Acceleration.Acc_Y_[m/s2]
Acceleration.Acc_Z_[m/s2]
Acceleration.Spin_X_[DEG/s2]
Acceleration.Spin_Y_[DEG/s2]
Acceleration.Spin_Z_[DEG/s2]
```

```
Global_section_forces.Bending_about_X_[Nm]
Global_section_forces.Bending_about_Y_[Nm]
Global_section_forces.Bending_about_Z_[Nm]
Global_section_forces.Force_X_[N]
Global_section_forces.Force_Y_[N]
Global_section_forces.Force_Z_[N]
```

```
Local_section_forces.Axial_force_[N]
Local_section_forces.Shear_force_Y_[N]
Local_section_forces.Shear_force_Z_[N]
Local_section_forces.Torsion_moment_[Nm]
Local_section_forces.Bending_about_Y_[Nm]
Local_section_forces.Bending_about_Z_[Nm]
```

```
Net.Axial_force_in_horizontal_twines_[N]
Net.Axial_force_in_vertical_twines_[N]
```

```
Slamming.CD_Coeff
Slamming._Submerged_volume_[m3]
```

```
Pilhoyde
Pilhoyde2
```

```
Rotation.Rotation_about_X_[DEG]
Rotation.Rotation_about_Y_[DEG]
Rotation.Rotation_about_Z_[DEG]
```

```
Stress_component.Left_web_ [MPa]
Stress_component.Right_web_[MPa]
Stress_component.Lower_flange_[MPa]
Stress_component.Upper_flange_[MPa]
```

```
Velocity.Velocity_X_[m/s]
Velocity.Velocity_Y_[m/s]
Velocity.Velocity_Z_[m/s]
Velocity.Spin_about_X_[DEG/s]
Velocity.Spin_about_Y_[DEG/s]
Velocity.Spin_about_Z_[DEG/s]
```

```
Von_Mises_stress_[MPa]
Von_Mises_stress_2_[MPa]
```

```
Displacement.X_[m]
Displacement.Y_[m]
Displacement.Z_[m]
Displacement.Total_[m]
```

```
Convergence_norm
```

8.2 Available in PFAT

```
Acceleration.Max_X_[m/s2]
Acceleration.Max_Y_[m/s2]
Acceleration.Max_Z_[m/s2]
Acceleration.Min_X_[m/s2]
Acceleration.Min_Y_[m/s2]
Acceleration.Min_Z_[m/s2]
```

```
Acceleration.Max_rX_[rad/s2]
Acceleration.Max_rY_[rad/s2]
Acceleration.Max_rZ_[rad/s2]
Acceleration.Min_rX_[rad/s2]
Acceleration.Min_rY_[rad/s2]
Acceleration.Min_rZ_[rad/s2]
```

```
Global_section_forces.Max_Moment_about_X_[Nm]
Global_section_forces.Max_Moment_about_X_[Nm]
Global_section_forces.Max_Moment_about_X_[Nm]
Global_section_forces.Max_force_X_[N]
Global_section_forces.Max_force_Y_[N]
Global_section_forces.Max_force_Z_[N]
```

```
Input_data.Component_number
Input_data.Component_type
Input_data.Cross_section_area_[mm2]
Input_data.Current_reduction_[%]
Input_data.E_modulus_[MPa]
Input_data.Element_no
```

```
Input_data.Git_[MPa]
Input_data.I_t_[cm4]
Input_data.I_y_[cm4]
Input_data.I_z_[cm4]
Input_data.Load_type
Input_data.Mass_2D
Input_data.Mass_per_m_[kg/m]
Input_data.Net_Cross_section_area_twine_[mm2]
Input_data.Net_Fouling_Factor
Input_data.Net_Horisontal_halvmesh_length_[mm]
Input_data.Net_Vertical_halvmesh_length_[mm]
Input_data.Net_Solidity_2d_[%]
Input_data.Net_Solidity_no_knot_[%]
Input_data.Net_Weight_in_water_[kg/m3]
Input_data.No_Current_reduction
Input_data.Node_numbers
Input_data.Relative_weight_[N/m]
Input_data.Waterplane_width_[mm]
Input_data.Weight_per_m2
Input_data.Twine_density_[kg/m3]
```

```
Local_section_forces.Max_axial_compression_force_[N]
Local_section_forces.Max_axial_tension_force_[N]
Local_section_forces.Max_axial_force_[N]
Local_section_forces.Max_moment_about_Y_[Nm]
Local_section_forces.Max_moment_about_Z_[Nm]
Local_section_forces.Max_shear_force_Y_[N]
Local_section_forces.Max_shear_force_Z_[N]
Local_section_forces.Max_torsion_moment_[Nm]
```

```
Max_Displacement.Displacement_X_[mm]
Max_Displacement.Displacement_Y_[mm]
Max_Displacement.Displacement_Z_[mm]
```

```
Max_Displacement.Positive_X_[mm]
Max_Displacement.Positive_Y_[mm]
Max_Displacement.Positive_Z_[mm]
Max_Displacement.Negative_X_[mm]
Max_Displacement.Negative_Y_[mm]
Max_Displacement.Negative_Z_[mm]
```

```
Max_Rotation.Rotation_about_X_[DEG]
Max_Rotation.Rotation_about_Y_[DEG]
Max_Rotation.Rotation_about_Z_[DEG]
```

```
Max_Rotation.Positive_about_X_[DEG]
Max_Rotation.Positive_about_Y_[DEG]
Max_Rotation.Positive_about_Z_[DEG]
Max_Rotation.Negative_about_X_[DEG]
Max_Rotation.Negative_about_Y_[DEG]
```

```
Max_Rotation.Negative_about_Z_[DEG]
```

```
Max_hinge_forces.Axial_force_1_[N]
Max_hinge_forces.Axial_force_2_[N]
Max_hinge_forces.Resultant_Von_Mises_1_[N]
Max_hinge_forces.Resultant_Von_Mises_2_[N]
Max_hinge_forces.Shear_force_Y_[N]
Max_hinge_forces.Shear_force_Z_[N]
```

```
Max_sea_pressure_[Pa]
```

```
Net.Max_force_in_horizontal_twines_[N]
Net.Max_force_in_vertical_twines_[N]
```

```
Nominal_stress_range.Left_web_[MPa]
Nominal_stress_range.Right_web_[MPa]
Nominal_stress_range.Lower_flange_[MPa]
Nominal_stress_range.Upper_flange_[MPa]
```

```
Stresscomponent_compress.Left_web_[MPa]
Stresscomponent_compress.Right_web_[MPa]
Stresscomponent_compress.Lower_flange_[MPa]
Stresscomponent_compress.Upper_flange_[MPa]
```

```
Stresscomponent_max.Left_web_[MPa]
Stresscomponent_max.Right_web_[MPa]
Stresscomponent_max.Lower_flange_[MPa]
Stresscomponent_max.Upper_flange_[MPa]
```

```
Total_pressure.Beams_Truss_[Pa]
```

```
Velocity.Max_X_[m/s]
Velocity.Max_Y_[m/s]
Velocity.Max_Z_[m/s]
Velocity.Min_X_[m/s]
Velocity.Min_Y_[m/s]
Velocity.Min_Z_[m/s]
```

```
Velocity.Max_rX_[rad/s]
Velocity.Max_rY_[rad/s]
Velocity.Max_rZ_[rad/s]
Velocity.Min_rX_[rad/s]
Velocity.Min_rY_[rad/s]
Velocity.Min_rZ_[rad/s]
```

```
Von_Mises_stress_max_[MPa]
```

8.3 Not available in file, only in AquaView

```
Position.X_[m]
Position.Y_[m]
Position.Z_[m]
```

```
Rigid_body_rotation.Around_Axis_X_[DEG]
Rigid_body_rotation.Around_Axis_Y_[DEG]
Rigid_body_rotation.Around_Axis_Z_[DEG]
```

```
Distance_between_nodes.X_axis_[m]
Distance_between_nodes.Y_axis_[m]
Distance_between_nodes.Z_axis_[m]
Distance_between_nodes.All_axis_[m]
Distance_to_terrain_[m]
```

```
Max_distance_between_components.X_axis_[m]
Max_distance_between_components.Y_axis_[m]
Max_distance_between_components.Z_axis_[m]
Max_distance_between_components.All_axis_[m]
```

```
Min_distance_between_components.X_axis_[m]
Min_distance_between_components.Y_axis_[m]
Min_distance_between_components.Z_axis_[m]
Min_distance_between_components.All_axis_[m]
```

```
Max_distance_between_elements.X_axis_[m]
Max_distance_between_elements.Y_axis_[m]
Max_distance_between_elements.Z_axis_[m]
Max_distance_between_elements.All_axis_[m]
```

```
Min_distance_between_elements.X_axis_[m]
Min_distance_between_elements.Y_axis_[m]
Min_distance_between_elements.Z_axis_[m]
Min_distance_between_elements.All_axis_[m]
```

```
Volume_old_[m^3]
Area_[m^2]
Volume_[m^3]
Volume_debug_[x]
```

```
Normal load_[%]
Accident_[%]
```

8.4 AquaCross

```
Sigma.Axial_force_[MPa];
Sigma.Moment_Y_[MPa];
Sigma.Moment_Z_[MPa];
Tau.Shear_y_[MPa];
Tau.Shear_Z_[MPa];
Tau.Torsion_[MPa];
Tau.Torsion2_[MPa];
Sigma.Resulting_[MPa];
Tau.Resulting_[MPa];
```