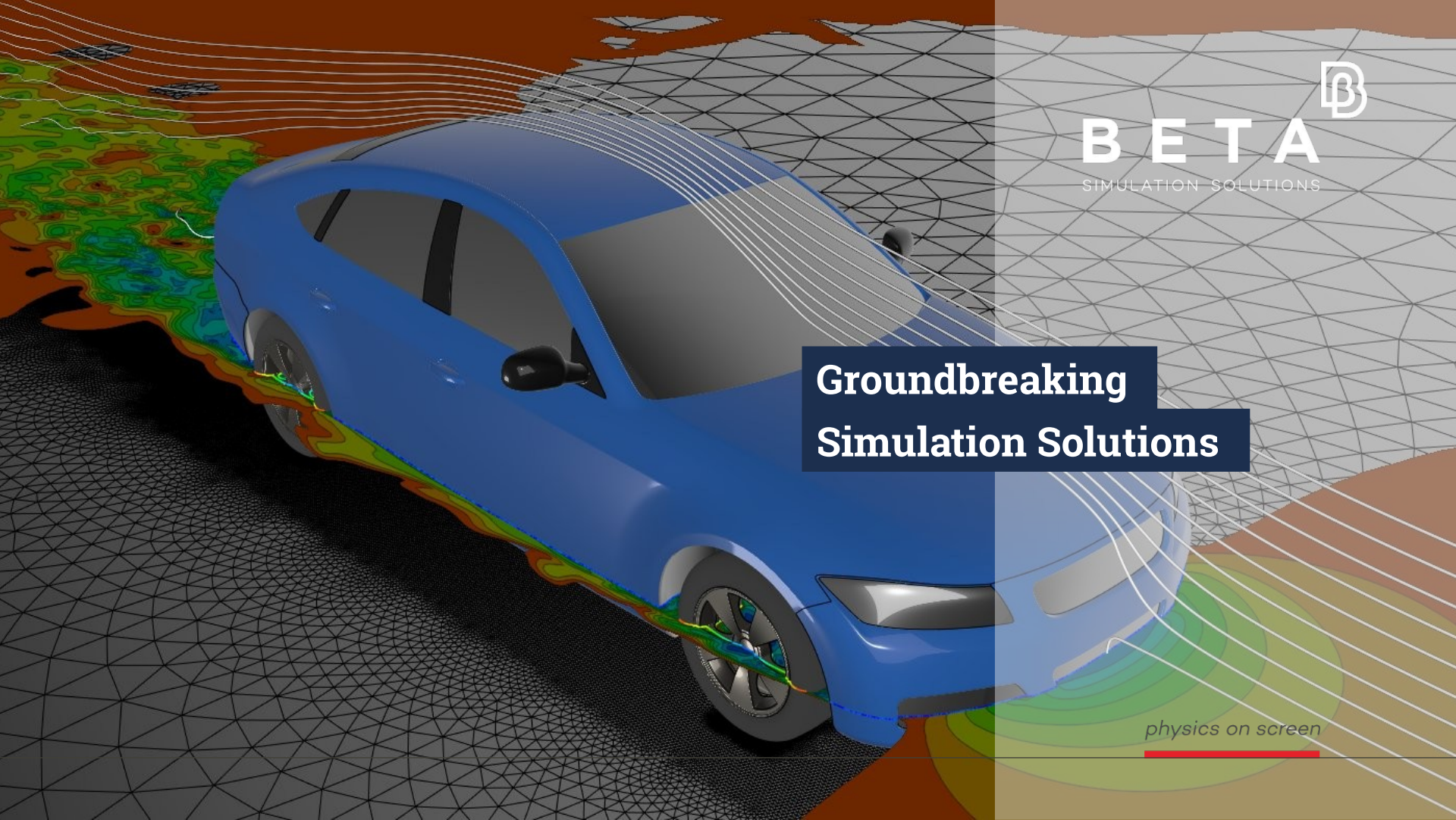

BETA
SIMULATION SOLUTIONS
Groundbreaking
Simulation Solutions
physics on screen

CGNS example files:

Fixes & readability improvements,
that help us ensure conformance.

Main Problems:

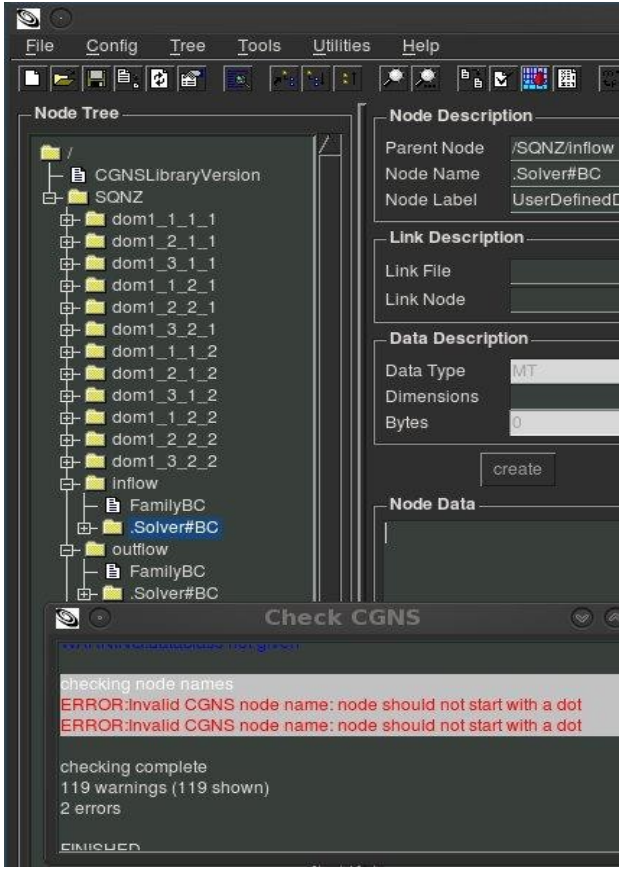
1) Non-conformance
2) Misleading idioms
3) Inadvisable idioms
4) Missing info

Consider also:

- Is *cgnscheck* up to date? It should be our first line of validation!
- Can *cgnsview* perform the most basic file modifications correctly?
- Do we need a mininal “conformance grade” so that vendors can verify their software?

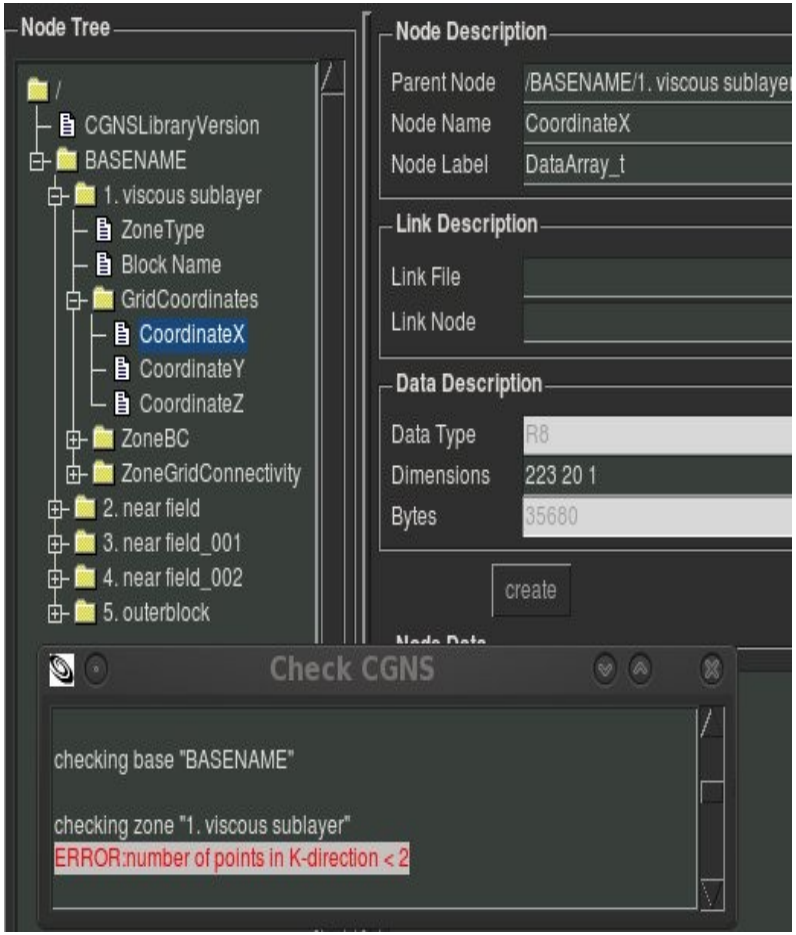
CGNS example files

1) Non-Conformance



Structured files errors:

- Have bad **node** names that start with ".", such as .SolverBC (*sqnz_s.adf.cgns, sqnz_s.hdf.cgns*)



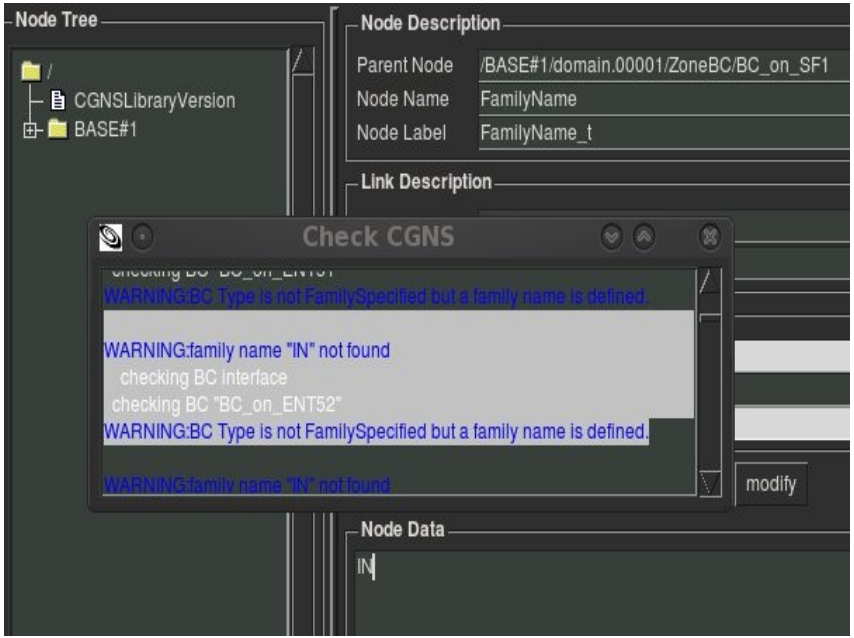
CGNS example files

1) Non-Conformance

Structured files errors:

- Wrong 2D definition, because they use “3” on CellDimension (*oversetnasa1.cgns, oversetnasa2.cgns*)

CGNS example files



1) Non-Conformance

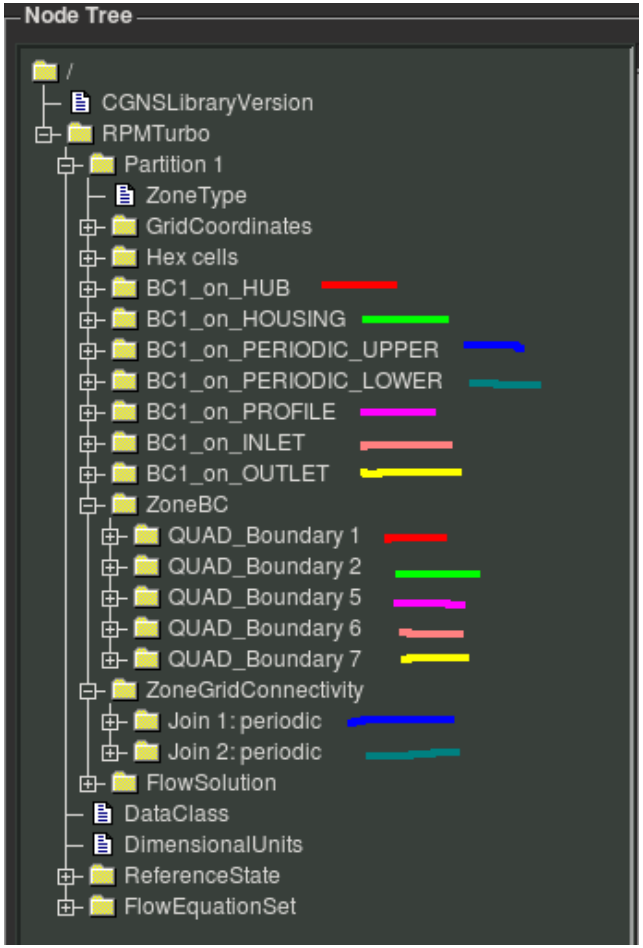
Structured files errors:

- Family names exist but families don't (*wbnp_hex_coarse.cgns*)

CGNS example files

2) Misleading idioms

- We mostly dealt with the implied notion that an element section (*Elements_t*) has a 1-to-1 correspondence with BCs (*BC_t*). Lets call it ***BC-correspondence*** issue.
- BC-correspondence is troublesome in many ways:
  - Software vendors use *BC_t*/*Elements_t* interchangeably, maybe don't even read *BC_t*
  - Facilitates usage of *PointRange* on *BC_t*, so vendors often ignore or treat as error the *PointList* ! Thus, users who prefer MIXED sections + *PointList* BCs, can't exchange files with those who prefer Separated sections + *PointRanges*!
  - Multiple sections can not be used in order to reduce memory footprint (during IO operations), especially for big meshes
- BC-correspondence is propagated by the example files, and hurts interoperability.

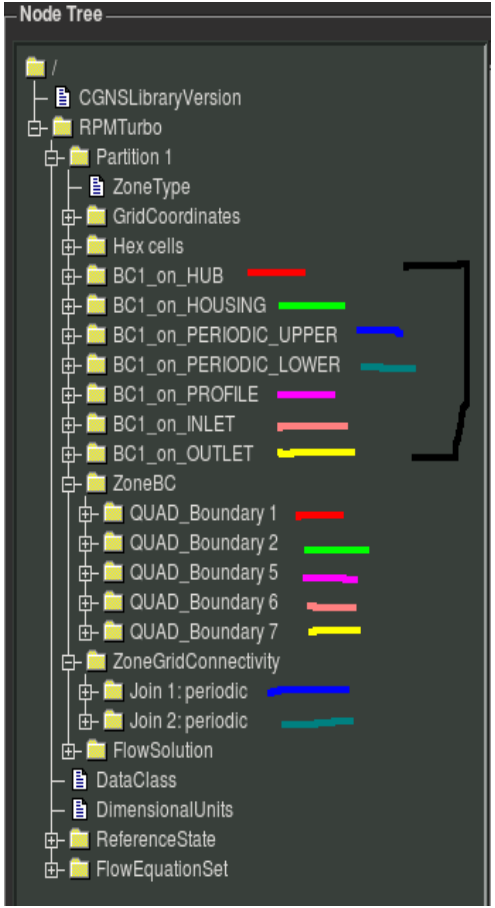


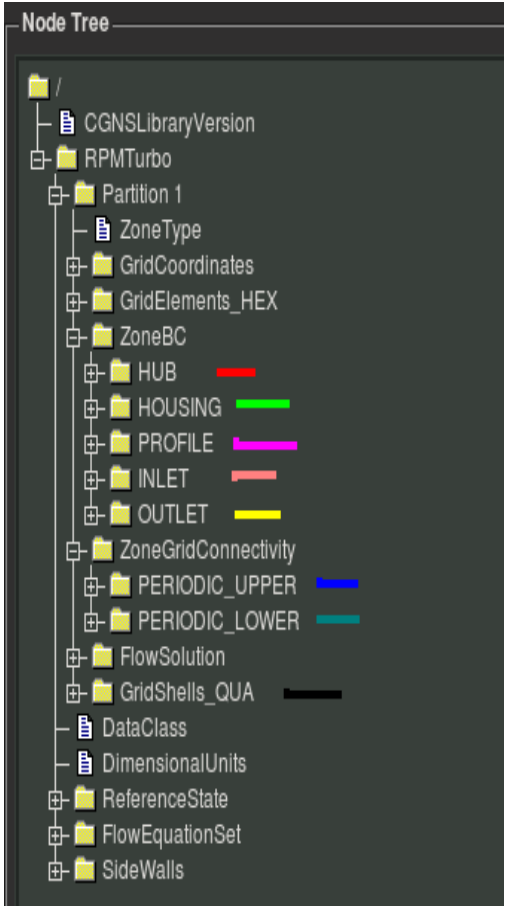
CGNS example files

2) Misleading idioms

For example, file SC10_steady.cgns:

- Each BC has its own element section.
- Here, we can see that the author named the element sections instead of the BCs!



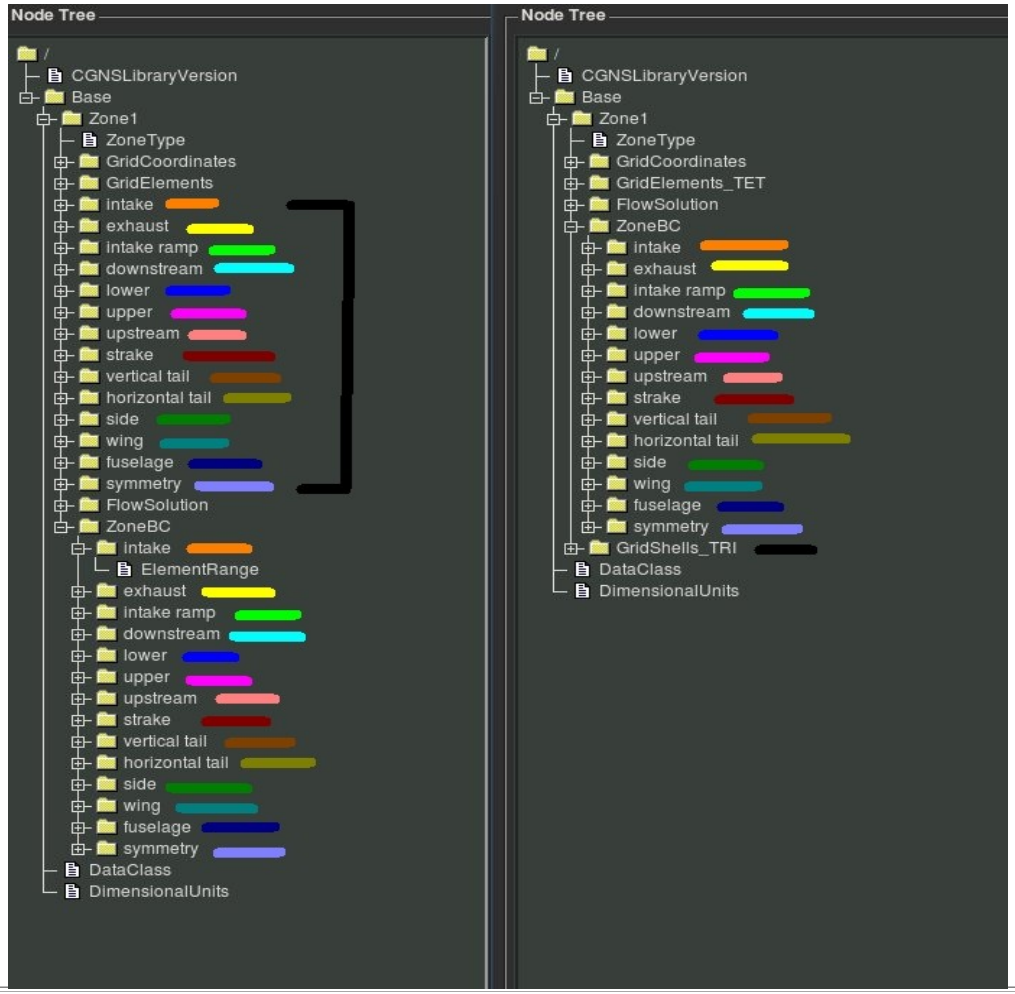


CGNS example files

2) Misleading idioms

Improvements on file SC10_steady.cgns:

- All elements grouped in sections that won't imply 1-to-1 relation with BCs.
- BCs get the names the original author intended.
- Readability improved, no need to plot to understand the contents.

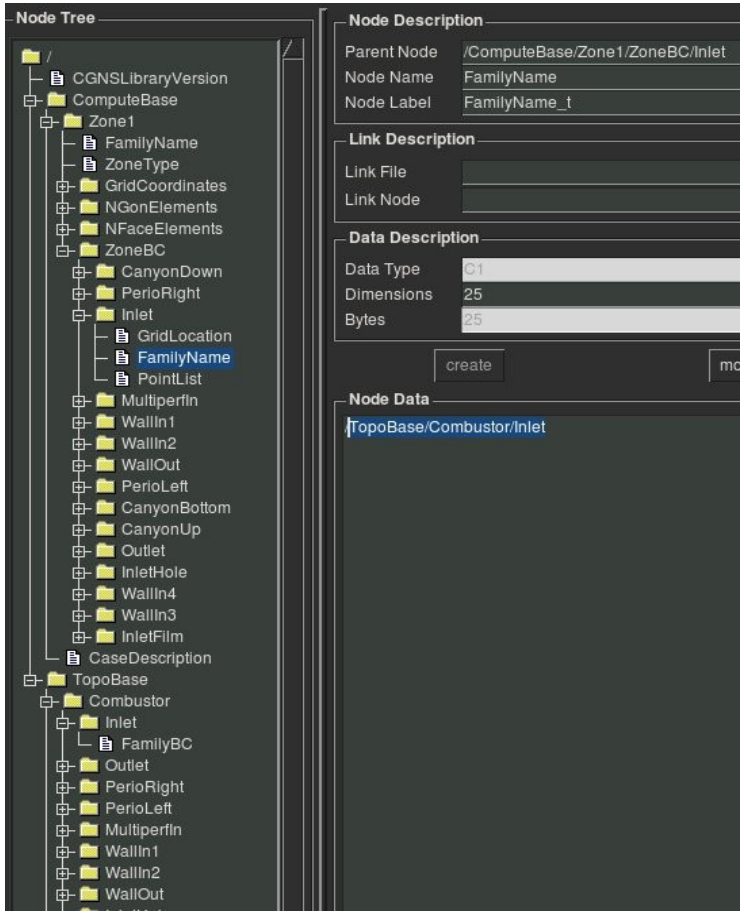


CGNS example files

2) Misleading idioms

Same improvements on file y17.cgns:

- All elements grouped in sections that won't imply 1-to-1 relation with BCs.
- ParentData were preserved.
- Readability improved, no need to plot to understand the contents.



CGNS example files

2) Misleading idioms

File trappedvtx_ngon.cgns:

- It's valid to define *Families* in separate base according to the format.
- *Cgnscheck* reports warnings:
  1. that TopoBase contains no zones
  2. that ComputeBase contains no families

CGNS example files

- Using bases for timesteps, instead of the standard BaseIterativeData/ZoneIterativeData (*MovingTransient.cgns*)
  - This idiom might restrict our ability to expand the API. Using many bases might need to be reserved for a specific purpose (e.g. mesh connectivity between Structured and Unstructured domains).

3) Inadvisable idioms

- Using separate bases for timesteps

CGNS example files

- Defining BCs on nodes for 3-D domains (*HeatingCoil.cgns, StaticMixer.cgns, MovingTransient.cgns*)
  - In this case, the bounding vertices will be located on multiple BC patches. *"If boundary conditions are imposed using collocation at vertices, then for this case there is no mechanism to determine which BC patch takes precedence for any of these bounding vertices."*
  - Not invalid, but how will a CFD model correctly set the BCs up? The ordering of the nodes will become important.

3) Inadvisable idioms

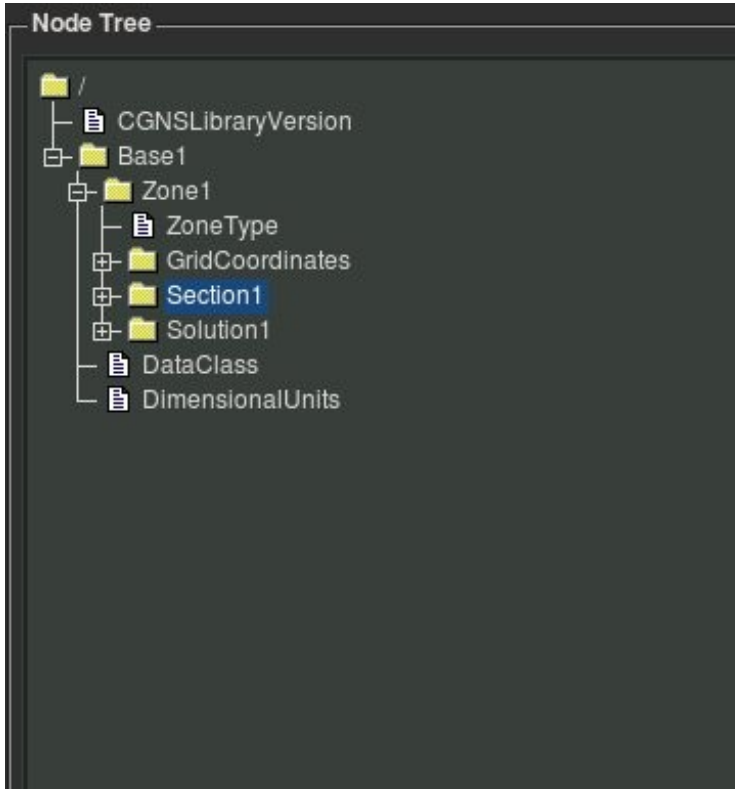
- BCs on Nodes, for 3D domains

CGNS example files

4) Missing info

- No unstructured file exists, that shows up the correct usage of Connectivity/Periodicity.
- Files with volume mesh only exist (no boundary elements and no BC_t nodes) (*tut21.cgns*).
- No files exist where there is a *PointList* that can't be treated as a *PointRange*.

CGNS example files

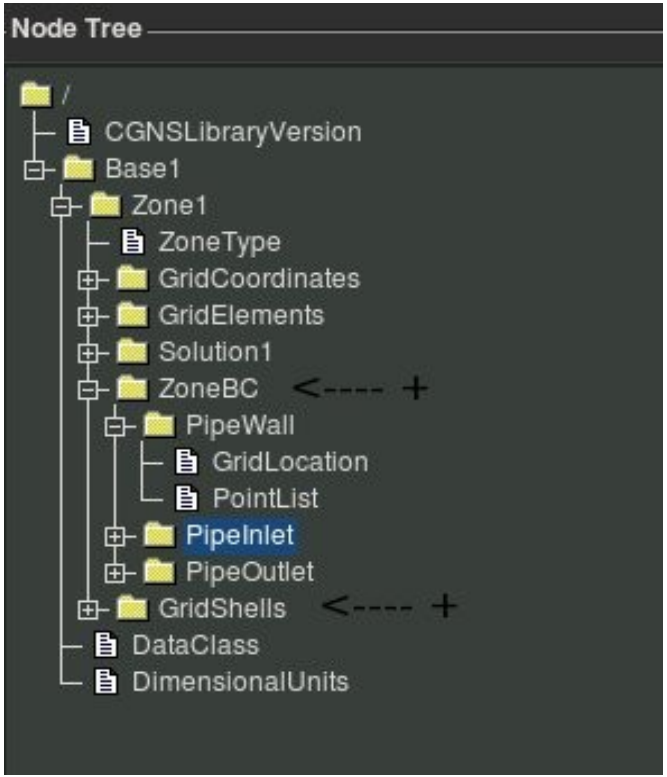


4) Missing info

File tut21.cgns:

- has no boundary elements
- no *BC_t* defined
- Is a simple and good example to display *PointLists*

CGNS example files



4) Missing info

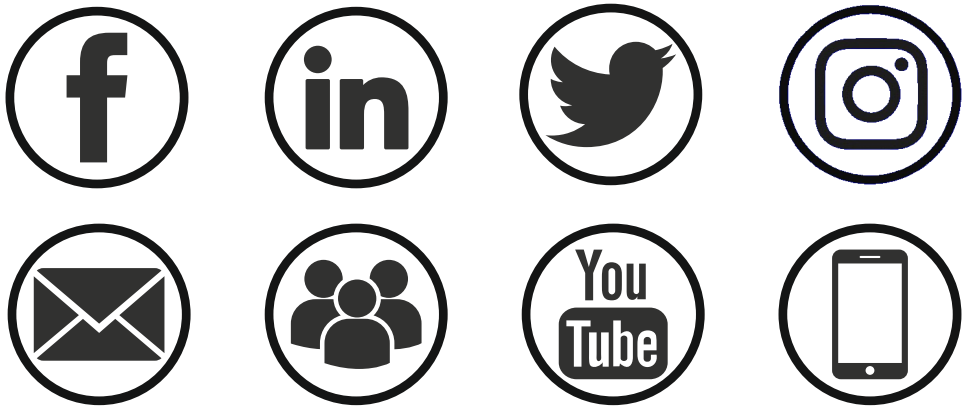
Improvements on file tut21.cgns:

- Surface elements added, with IDs that absolutely require the usage of *PointLists* on BCs
- BCs are set on the surface elements, using *PointLists*
- Helps completeness of the example files!

Remarks

- We improved 3 unstructured files in order to help you understand the way we think most appropriate for the example files. Any feedback from you is appreciated.
- We improved only 3 unstructured files in a way that nothing is changed on the mesh compared to the initial file and any additional data (such as results, parent data etc) would remain valid.
- The rest unstructured files might need much more effort, because reordering of faces will take place. This could affect the validity of the contained results and/or parent data.
-

BETA
SIMULATION SOLUTIONS



Stay connected