Proposal for the addition of point by point grid motion to the CGNS

March 2000

Presented by: Robert Bush, United Technology Research Center
Reviewed by: Christopher Rumsey, NASA Langley Research Center
Diane Poirier, ICEM CFD Engineering

ArbitraryGridMotion_t Data Structure

When a grid is in motion, it is often necessary to account for the position of each grid point as it deforms. When all grid points move at the same velocity, the grid keeps its original shape. This particular case of grid motion may be recorded under the `RigidGridMotion_t` data structure. On the other hand, if the grid points have different velocity, the mesh is deforming. The `ArbitraryGridMotion_t` data structure allows the CGNS file to contain information about arbitrary grid deformations. If not present, the grid is assumed to be rigid.

In addition to the creation of the `ArbitraryGridMotion_t` data structure to record the velocity of each grid point, it is proposed to allow multiple `GridCoordinates_t` nodes under a `Zone_t`. This would enable the storage of the instantaneous grid locations at different time steps or iterations. The original grid coordinates definition, as currently defined in the SIDS, would remain unchanged with the name "`GridCoordinates`".

It is proposed that the arbitrary grid motion be recorded independently for each zone of the CGNS base. Therefore the `ArbitraryGridMotion_t` data structure would be added under each the zone data structure (`Zone_t`). There may be zero to several `ArbitraryGridMotion_t` nodes under a `Zone_t` node. The multiple arbitrary grid motion definition may be associated to different iteration or time step in the computation. This association is recorded under the `IterativeOrTemporal_t` data structure.

SIDS definition of the `ArbitraryGridMotion_t` data structure:

The `ArbitraryGridMotion_t` data structure under the `Zone_t` data structure:

```
Zone_t<int CellDimension, int PhysicalDimension > :=
  {
  List( ArbitraryGridMotion_t ArbitraryGridMotion1, ...,
                              ArbitraryGridMotionN ) ;          (o)
  List( GridCoordinates_t<IndexDimension, VertexSize>
                        GridCoordinates, GridCoordinates1, ...,
                        GridCoordinatesN ) ;                    (o)
  ...
  }
```

The `ArbitraryGridMotion_t` data structure:

```
ArbitraryGridMotion_t :=
  {
  ArbitraryGridMotionType_t ArbitraryGridMotionType ;           (r)
  List(DataArray_t<real,1,VertexSize>
            GridVelocityX, GridVelocityY, ... ) ;               (o)
  List( Descriptor_t Descriptor1 ... DescriptorN ) ;            (o)
```

```
DataClass_t DataClass ;                                    (o)
DimensionalUnits_t DimensionalUnits ;                      (o)
GridLocation_t GridLocation ;                              (o/d)
Rind_t<IndexDimension> Rind                                (o/d)

}
```

The `DataArray_t` nodes are used to store the components of the grid velocity vector. The table below lists the new data-name identifiers proposed to record these vectors in the cartesian, cylindrical and spherical coordinate systems.

Data-Name Identifier	Description	Units
GridVelocityX	x-component of grid velocity at a grid point	L/T
GridVelocityY	y-component of grid velocity at a grid point	L/T
GridVelocityZ	z-component of grid velocity at a grid point	L/T
GridVelocityR	R-component of grid velocity at a grid point	L/T
GridVelocityTheta	Theta- component of grid velocity at a grid point	α/T
GridVelocityPhi	Phi-component of grid velocity at a grid point	α/T
GridVelocityXi	Xi-component of grid velocity at a grid point	L/T
GridVelocityEta	Eta-component of grid velocity at a grid point	L/T
GridVelocityZeta	Zeta-component of grid velocity at a grid point	L/T

Table of proposed data-name identifiers to record the grid velocity

Definitions:

- `ArbitraryGridMotionType_t` is an enumeration type that describes the type of arbitrary grid motion. The type is either `NonDeformingGrid` or `DeformingGrid`.

```
ArbitraryGridMotionType_t := Enumeration(
    NonDeformingGrid,
    DeformingGrid ) ;
```

Notes:

- The only required element of the `ArbitraryGridMotion_t` data structure is the `ArbitraryGridMotionType`. Thus, even if a deforming grid application does not require the storage of grid velocity data, the `ArbitraryGridMotion_t` node must exist (with `ArbitraryGridMotionType=DeformingGrid`) to indicate that deformed grid points (`GridCoordinate_t`) exist for this zone.

- The `DataClass_t`, `DimensionalUnits_t` and `Descriptor_t` nodes may optionally be specified under the `RigidGridMotion_t` nodes.

- Point by point grid velocity implies a deformation (or potentially only motion) of the grid points relative to each other. Because the original grid coordinates definition is to remain unchanged with the name "`GridCoordinates`", any deformed coordinates are to be written with a different name (e.g., "`GridCoordinates1`" or another used-defined name) and are to be pointed to using `GridCoordinatesPointers` in the data structure `IterativeOrTemmporalData_t`.

- Point by point grid velocity may also lead to relative motion of grid zones or blocks, or movement of grid along abutting interfaces. However, no attempt is made here to require that the `ZoneGridConnectivity_t` information be updated to be consistent with the new grid locations. The user is responsible to ensure that any `ZoneGridConnectivity_t` information is kept up to date.

- `Rind` is an optional field that indicates the number of rind planes included in the grid velocity data. It only applies to structured zones.

- The `GridLocation` specifies the location of the velocity data with respect to the grid; if absent, the data is assumed to coincide with grid vertices (i.e. `GridLocation = Vertex`).

ADF file mapping definition of the ArbitraryGridMotion_t data structure: