Proposal for extending the support of averaging interfaces
Robert Magnan

Currently (SIDS Document version 2.4.7), the specification of such interface is made through a AverageInterface_t which is a child either of a GridConnectivity_t or of a GridConnectivity1to1_t. It is however impossible to correctly specify some type of averaging interfaces. This problem appears mostly in conjunction with periodic boundaries and is typically observed in rotating machinery problems. The problem lies in the definition of the mesh patches on either side of the interface.

Under a GridConnectivity1to1_t the cells have to match one-to-one which is a severe limitation. Under the more general GridConnectivity_t the interpolation factors have to be provided which is not really meaningful since some type of averaging takes place at the interface. In fact, with periodic problems, the span on either side of the interface might be different and interpolation is obviously impossible.

To facilitate the specification of such non-matching patches in structured meshes, it would be nice if the SIDS was flexible enough to allow connectivity specification by two point range in the general GridConnectivity_t structure.

We thus propose to modify a few items of the SIDS to accommodate such interfaces. The following pages highlight the required changes in the current SIDS section. Very little modification seems required in the mid-level library.

Due to my lack of familiarity, the proposed modification does not address averaging interfaces with Overset connectivity.

General Interface Connectivity Structure Definition: `GridConnectivity_t`

`GridConnectivity_t` contains connectivity information for generalized multizone interfaces, and may be used for any mix of structured and unstructured zones. Its purpose is to describe mismatched-abutting and overset interfaces, but can also be used for 1-to-1 abutting interfaces and interfaces where more complex interpolation (i.e. mixing plane) takes place.

For abutting interfaces that are not 1-to-1, also referred to as patched or mismatched, an interface patch is the subrange of the face of a zone that touches one and only one other zone. This structure identifies the subrange of indices (or array of indices) that make up the interface and gives their image in the adjacent (donor) zone. It also identifies the name of the adjacent zone. If a given face of a zone touches several (say *N*) adjacent zones, then *N* different instances of `GridConnectivity_t` are needed to describe all the interfaces. For a single abutting interface, two instances of `GridConnectivity_t` are needed in the database - one for each adjacent zone.

For overset interfaces, this structure identifies the fringe points of a given zone that lie in one and only one other zone. If the fringe points of a zone lie in several (say *N*) overlapping zones, then *N* different instances of `GridConnectivity_t` are needed to describe the overlaps. It is possible with overset grids that a single fringe point may actually lie in several overlapping zones (though in typical usage, linkage to only one of the overlapping zones is kept). There is no restriction against a given fringe point being contained within multiple instances of `GridConnectivity_t`; therefore, this structure allows the description of a single fringe point lying in several overlapping zones.

```
GridConnectivityType_t := Enumeration(
    Null,
    Overset,
    Abutting,
    Abutting1to1,
    Averaging,
    UserDefined ) ;

GridConnectivity_t< int IndexDimension, int CellDimension > :=
    {
    List( Descriptor_t Descriptor1 ... DescriptorN ) ;                (o)

    GridConnectivityType_t GridConnectivityType ;                     (o/d)

    GridLocation_t GridLocation ;                                     (o/d)

    IndexRange_t<IndexDimension> PointRange ;                         (o:r:r:o)
    IndexRange_t<IndexDimension> PointRangeDonor ;                    (o:o:r:o)
    IndexArray_t<IndexDimension, PointListSize, int>  PointList ;     (r:o:o:r)
    IndexArray_t<IndexDimension, PointListSize, int>  PointListDonor ; (o:r:o:r)
    IndexArray_t<IndexDimension, PointListSize, int>  CellListDonor ; (r:o:o:o)

    Identifier(Zone_t) ZoneDonorName ;                                (r)

    DataArray_t <real, 2, [CellDimension, PointListSize]>
       InterpolantsDonor                                              (r:o:o:o)

    GridConnectivityProperty_t GridConnectivityProperty ;             (o)

    List( UserDefinedData_t UserDefinedData1 ... UserDefinedDataN ) ; (o)

    int Ordinal ;                                                     (o)
    } ;
```

Notes

1. Default names for the Descriptor_t and UserDefinedData_t lists are as shown; users

may choose other legitimate names. Legitimate names must be unique within a given instance of GridConnectivity_t and shall not include the names CellListDonor, GridConnectivityProperty, GridConnectivityType, GridLocation, InterpolantsDonor, Ordinal, PointList, PointListDonor, PointRange, or PointRangeDonor.

2. ZoneDonorName must be equated to a zone identifier within the current CGNS database (i.e. it must be equal to one of the Zone_t identifiers contained in the current CGNSBase_t entity).
3. If GridConnectivityType is absent, then its default value is Overset.
4. GridLocation should be Vertex for Abutting, Abutting1to1 and Averaging interfaces; in other words, the connectivity information is always given with respect to the grid vertices. For Overset interfaces, GridLocation can be either Vertex or CellCenter. In any case, if GridLocation is absent, then its default value is Vertex.
5. One of PointRange and PointList must be specified, but not both.
6. If PointRange is specified, then an index ordering convention is needed to map receiver-zone grid points to donor-zone grid points. FORTRAN multidimensional array ordering is used.
7. If GridConnectivityType is Abutting1to1, Abutting or Averaging, then PointRange or PointList must define points associated with a face subrange (if the zone is structured, all points must be in a single computational grid plane); the donor-zone grid locations defined by PointRangeDonor, PointListDonor or CellListDonor must also be associated with a face subrange.
8. If GridConnectivityType is not Averaging, either PointListDonor alone, or CellListDonor plus InterpolantsDonor, must be used. The use of PointListDonor is restricted to Abutting1to1, whereas CellListDonor plus InterpolantsDonor can be used for any interface type.
9. If GridConnectivityType is Averaging, CellListDonor and InterpolantsDonor may not be used.
10. **PointRangeDonor** may only be used when the interface is **Averaging**. It contains the interface patch subrange of indices for the adjacent zone (whose identifier is given by ZoneDonorName). By convention the indices contained in PointRange and PointRangeDonor refer to vertices.

    Note: *We could also allow it when the GridConnectivityType is **Abutting1to1** but a **Transform** node would then be required and that would make GridConnectivity1to1 obsolete.*

The type of multizone interface connectivity may be Overset, Abutting, Abutting1to1 or Averaging. Overset refers to zones that overlap; for a 3-D configuration the overlap is a 3-D region. Abutting refers to zones that abut or touch, but do not overlap (other than the vertices and faces that make up the interface). Abutting1to1 is a special case of abutting interfaces where grid lines are continuous across the interface and all vertices on the interface are shared by the two adjacent zones. Averaging refers to all types of interfaces across which the exchange of information is not straightforward interpolation. Further specification of the averaging may be provided using the optional GridConnectivityProperty_t structure.

The interface grid points within the receiver (donor) zone may be specified by PointRange (PointRangeDonor) if they constitute a logically rectangular region (e.g. an abutting interface where an entire face of the receiver zone abuts with a part of a face of the donor zone). In all

other cases, PointList should be used to list the receiver-zone grid points making up the interface. For a structured-to-structured interface, all indices in PointRange, PointRangeDonor or PointList should have one index element in common (see note 7).

GridLocation identifies the location of indices within the receiver zone described by PointRange or PointList; it also identifies the location of indices defined by PointListDonor in the donor zone. For Overset interfaces, GridLocation may be either Vertex or CellCenter, allowing the description of the overlap region in the receiver zone to be consistent with the grid location used for storing the flow solution. For Abutting and Abutting1to1 interfaces, GridLocation should be Vertex. For Averaging interfaces, GridLocation should be Vertex.

GridLocation does not apply to CellListDonor or InterpolantsDonor. The CellListDonor is always an index or indices that define a particular cell or element, while the InterpolantsDonor defines an interpolation value relative to the cell/element vertices.

PointListDonor may only be used when the interface is Abutting1to1 or Averaging. It contains the images of all the receiver-zone interface points in the donor zone. If the zone is structured, all indices in PointListDonor should have one index element in common.

For mismatched (Abutting or Overset) interfaces, the zone connectivity is defined using the combination of CellListDonor and InterpolantsDonor. CellListDonor contains the list of donor cells in which each node of the receiver zone can be located. InterpolantsDonor contains the interpolation factors to locate the receiver nodes in the donor cells. InterpolantsDonor may be thought of as bi- or tri-linear interpolants (depending on CellDimension) in the cell of the donor zone.

For Averaging interfaces the connectivity is defined by indicating (either through a range or a list of points) the patches on either side of the interface. Those patches do not have to match in any way – not even geometrically. Some mean of interpolation should be defined using the optional child GridConnectivityProperty_t structure. No attempt should be made to match the points from one side to the points of the other side, the points merely serve to identify the connected regions. The number of points on each side do not have to match one another.

A GridConnectivityProperty_t data structure may be used to record special properties associated with particular connectivity patches, such as a periodic interface, or an interface where data is to be averaged in some way.

The UserDefinedData_t data structure allows arbitrary user-defined data to be stored in Descriptor_t and DataArray_t children without the restrictions or implicit meanings imposed on these node types at other node locations.

Ordinal is user-defined and has no restrictions on the values that it can contain. It is included for backward compatibility to assist implementation of the CGNS system into applications whose I/O depends heavily on the numbering of zone interfaces. Since there are no restrictions on the values contained in Ordinal (or that Ordinal is even provided), there is no guarantee that the interfaces for a given zone in an existing CGNS database will have sequential values from 1 to N without holes or repetitions. Use of Ordinal is discouraged and is on a user-beware basis.

The following table summarizes the valid combinations for specifying the receiver and donor regions.

GridConnectivityType	***Receiver (current zone) specification***	***Donor specification***	***Comment***
Overset	PR	CLD+ID	
	PL	CLD+ID	
Abutting	PR	CLD+ID	
	PL	CLD+ID	
Abutting1to1	PR	PLD	
Averaging	PR	PRD	two structured zones
	PR	PLD	at least one structured zone
	PL	PRD	at least one structured zone
	PL	PLD	zones of any type
UserDefined or Null	no restriction	no restriction	

Legend:
PR : PointRange
PL : PointList
PRD : PointRangeDonor
PLD : PointListDonor
CLD : CellListDonor
ID : InterpolantsDonor