

Proposed documentation changes to SIDS for interface connectivity

(no changes necessary to software)

Chris Rumsey – c.l.rumsey@nasa.gov

General Interface Connectivity Structure Definition: GridConnectivity_t

...

8. If donor information is given, either PointListDonor alone, or CellListDonor **with or without** InterpolantsDonor, must be used. The use of PointListDonor is restricted to AButting1to1, whereas CellListDonor ~~plus InterpolantsDonor~~ can be used for any interface type.

9. Thus, for a GridConnectivityType that is not AButting1to1, there are 3 allowable levels of description concerning the donor information: (a) full, giving ZoneDonorName with CellListDonor plus InterpolantsDonor; (b) partial, giving ZoneDonorName with CellListDonor but no InterpolantsDonor; or (c) minimal, giving ZoneDonorName only.

...

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. 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. **In other words, when using InterpolantsDonor, the interpolants are always given with respect to the vertices of the donor zone. InterpolantsDonor is currently only defined for structured grids and certain basic unstructured grid element types.**

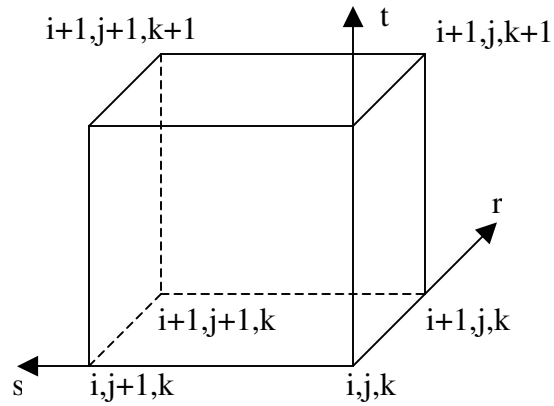
For structured grids, the interpolant value is given along each index direction, depending on the location within the cell. For example, if the point is located within the cell at a position 75% in the i-direction, 41% in the j-direction, and 20% in the k-direction, then InterpolantsDonor values (r, s, t) would be (0.75, 0.41, 0.20).

The interpolation function is a linear combination of the x, y, and z values at the surrounding nodes:

$$d = \sum_{i=1}^N W_i \cdot d_i$$

where d is the x, y, or z value at an interior point in the cell, d_i is the x, y, or z value at node i , and W_i is a weight at node i . The weights are functions of the parametric variables r, s and t (corresponding with the i, j, and k directions, respectively), which vary from 0 to 1, inclusively. For structured grids in 3-D, $N=8$. Note that for skewed, non-parallel grids, it is not always easy to determine the interpolants geometrically, and it may be necessary to solve an inverse problem using the interpolation function.

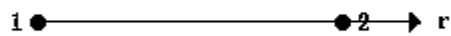
$$\begin{aligned}
W_{i,j,k} &= (1-r)(1-s)(1-t) \\
W_{i+1,j,k} &= r(1-s)(1-t) \\
W_{i+1,j+1,k} &= rs(1-t) \\
W_{i,j+1,k} &= (1-r)s(1-t) \\
W_{i,k,k+1} &= (1-r)(1-s)t \\
W_{i+1,j,k+1} &= r(1-s)t \\
W_{i+1,j+1,k+1} &= rst \\
W_{i,j+1,k+1} &= (1-r)st
\end{aligned}$$



For unstructured grids, InterpolantsDonor is defined only for the basic linear element types: BAR_2, TRI_3, QUAD_4, TETRA_4, PYRA_5, PENTA_6, and HEXA_8.

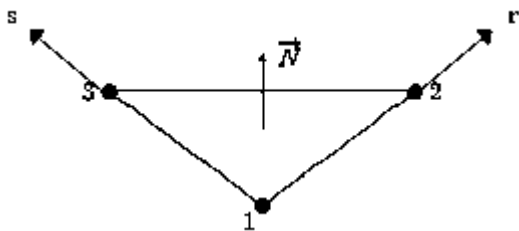
The directionality for the r , s , and t interpolants is defined as follows.

BAR_2



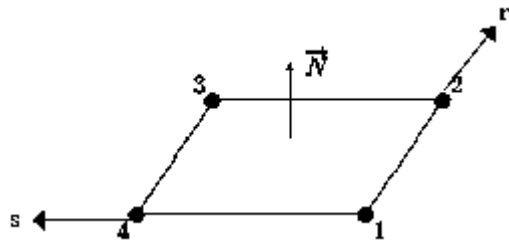
$$\begin{aligned}
W_1 &= (1-r) \\
W_2 &= r
\end{aligned}$$

TRI_3



$$\begin{aligned}
W_1 &= 1-r-s \\
W_2 &= r \\
W_3 &= s
\end{aligned}$$

QUAD_4



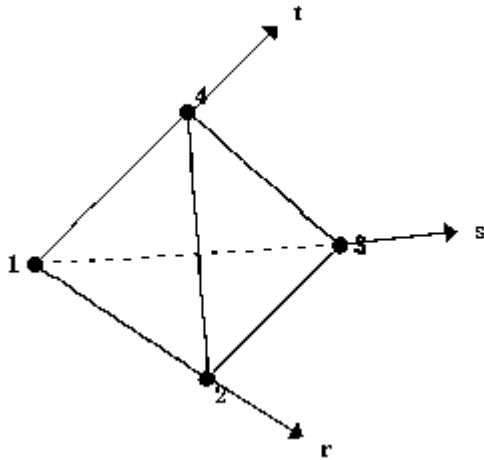
$$W_1 = (1-r)(1-s)$$

$$W_2 = r(1-s)$$

$$W_3 = rs$$

$$W_4 = (1-r)s$$

TETRA_4



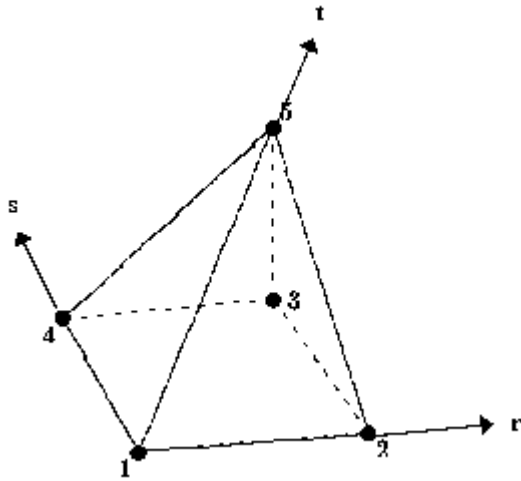
$$W_1 = 1-r-s-t$$

$$W_2 = r$$

$$W_3 = s$$

$$W_4 = t$$

PYRA_5



$$W_1 = (1-r)(1-s)(1-t)$$

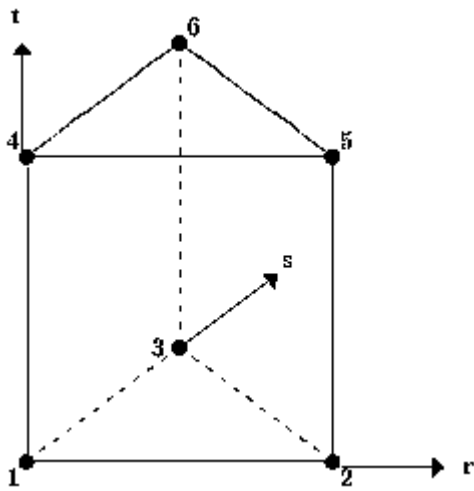
$$W_2 = r(1-s)(1-t)$$

$$W_3 = rs(1-t)$$

$$W_4 = (1-r)s(1-t)$$

$$W_5 = t$$

PENTA_6



$$W_1 = (1-r-s)(1-t)$$

$$W_2 = r(1-t)$$

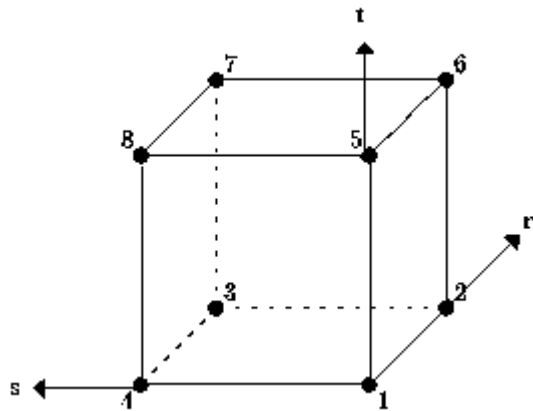
$$W_3 = s(1-t)$$

$$W_4 = (1-r-s)t$$

$$W_5 = rt$$

$$W_6 = st$$

HEXA_8



$$W_1 = (1-r)(1-s)(1-t)$$

$$W_2 = r(1-s)(1-t)$$

$$W_3 = rs(1-t)$$

$$W_4 = (1-r)s(1-t)$$

$$W_5 = (1-r)(1-s)t$$

$$W_6 = r(1-s)t$$

$$W_7 = rst$$

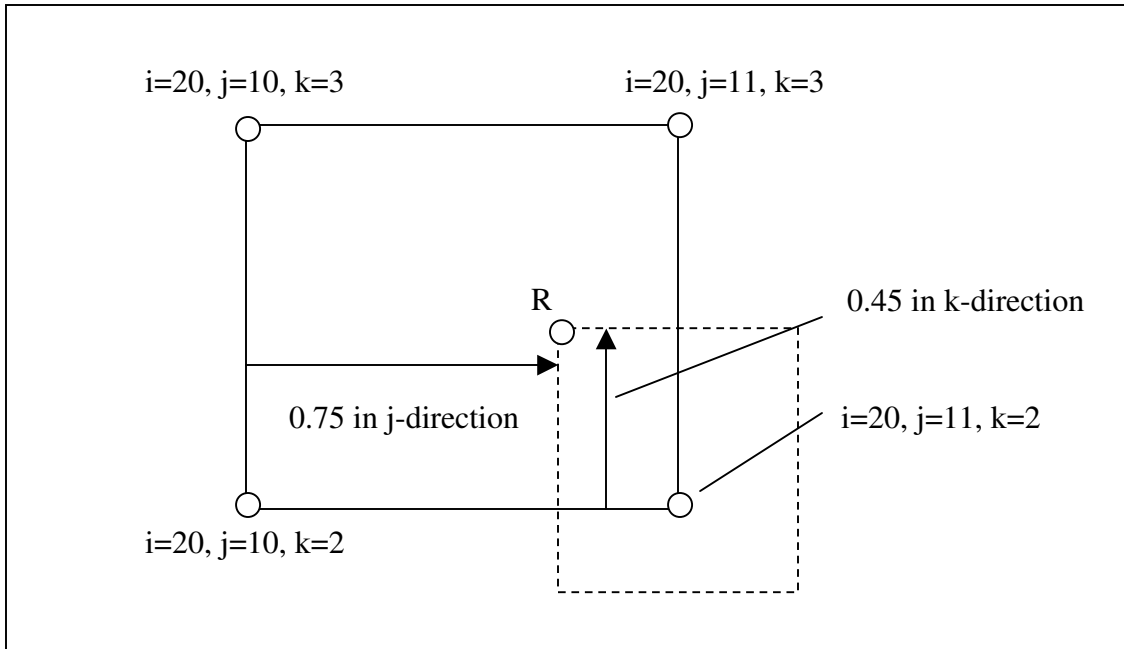
$$W_8 = (1-r)st$$

...

For mismatched or overset interfaces, the zone connectivity donor information, when given, is defined using **either the CellListDonor alone, or** the combination of CellListDonor and InterpolantsDonor. CellListDonor contains the list of donor cells **or elements** 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.

EXAMPLE 1

Say that you have a Structured zone in 3-D. Assume that at the interface between 2 zones you have the following situation:



In this particular example, the patching occurs on a “plane.” In other words, the two cells in 3-D have faces that abut in a 2-D sense. It is these faces that we are picturing here. The solid quad is the donor cell face, and the dashed quad is position of the receiving cell face relative to the donor cell. Note that since this is a 2-D-type of abutting case, one of the indices (in this case $i=20$, which represents i_{max}) of the donor cell is constant. For this example, the point “R” of the receiver cell is located within the donor cell pictured, and we wish to give the `CellListDonor` and `InterpolantsDonor` for it.

Because this is a structured grid, the `CellListDonor` in this case is given by:

`CellListDonor = (19, 10, 2).`

Here, we are using the convention established in “Structured Grid Notation and Indexing Conventions” that cell centers, face centers, and edge centers are indexed by the minimum i , j , and k indices of the connecting vertices.

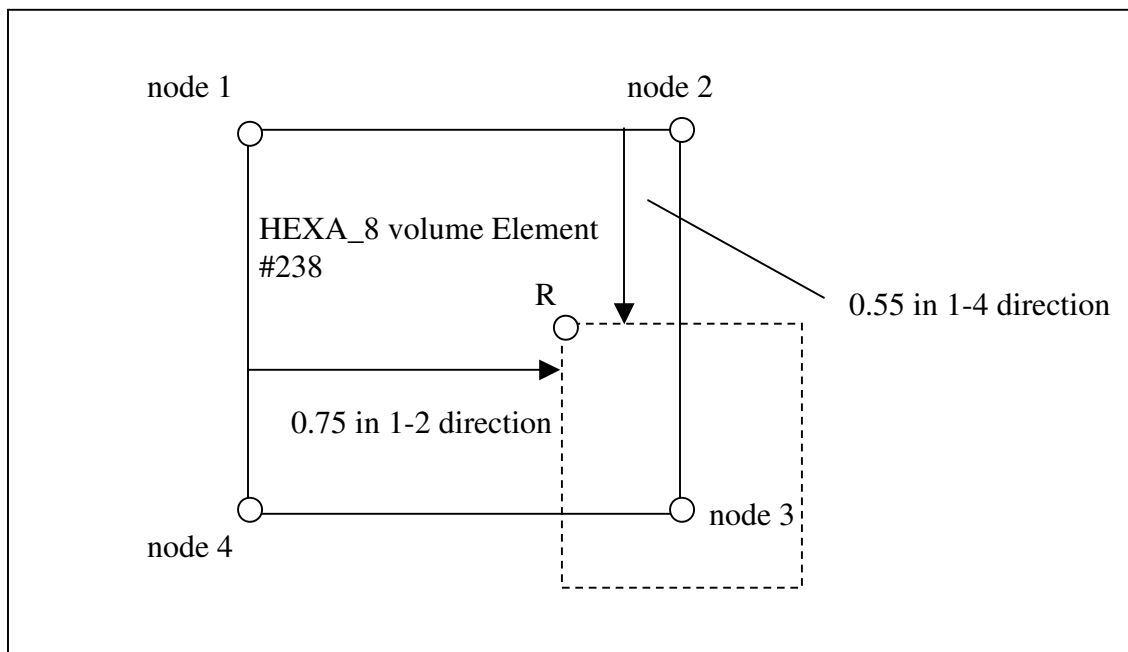
The `InterpolantsDonor` defines an interpolation value relative to the cell/element vertices. In this case, say that the point “R” is located 0.75 along the j -index direction and 0.45 along the k -index direction. (It also lies on the $i=20$, or i_{max} face.) Thus, in this example:

InterpolantsDonor = (1.0, 0.75, 0.45)

Note that if the donor zone was instead located on $i=1$ (imin face), then the CellListDonor would be (1, 10, 2) and the InterpolantsDonor would be (0.0, 0.75, 0.45).

EXAMPLE 2

As a second example, assume that you have the same setup as before, but now the grid is 3-D Unstructured. In this case, we no longer have a 3-D array of indices defining coordinate directions. Instead, we simply have a 1-D list of indices as well as a list of volume (and possibly face) elements composed of those indices. In this example we again are assuming the two zones abut in a 2-D sense. We now have the choice of describing the Donor in terms of its volume element or its boundary (face) element, if available. Here in this example, we use the volume element.



The HEXA_8 volume element has been appropriately numbered, using information given in “Unstructured Grid Element Numbering Conventions.” In this example, it is the 1-2-3-4 face of the volumetric element that is abutting with the other zone (but it could be any of its 6 faces).

The CellListDonor in this case is simply given by:

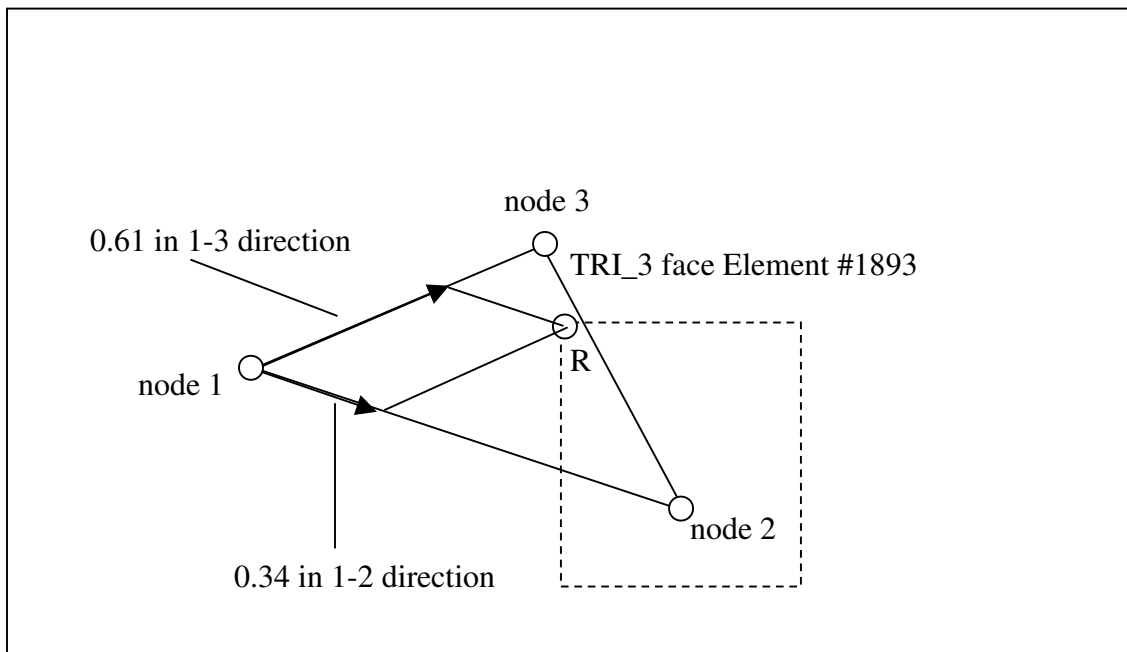
CellListDonor = (238).

Using the convention established above for HEXA_8 elements, the InterpolantsDonor would be:

InterpolantsDonor = (0.75, 0.55, 0.0)

EXAMPLE 3

As a third example, assume that you have two zones in a 3-D Unstructured grid with triangles and quads at its boundaries. Here the current zone (made up of quads faces) is abutting the donor zone (made up of triangle faces) in a 2-D sense. We again have the choice of describing the Donor in terms of its volume element or its boundary (face) element. Here in this example, we use the face element.



The CellListDonor in this case is simply given by:

CellListDonor = (1893).

Using the convention established above for TRI_3 elements, the InterpolantsDonor would be:

InterpolantsDonor = (0.34, 0.61)

In this case the third “dimension” of the InterpolantsDonor (although present) is not used, because by default the interpolation is only two-dimensional in the 2-D plane of the Donor face.