vectorOps - non-uniform

# Interpolation/extrapolation operators

## myFaceAverage (Face2AdjacentFace)

### face2CellCenter interpolation

The value is the face normal velocity. At location , meaning the direction into the face of which you want the average:

Similarly, at location ,

### cellCenter2Face interpolation

Now, we finally must interpolate between these two values (which are currently at cell centers)

Solving for this yields

Substituting we have

Collecting terms we have

## myFace2Edge

Face to edge requires a linear interpolation since the cell edges are not necessarily midway between two cell faces. The cell edges are coincident with the two cell faces that are used for the interpolation.

Solving for yields

where

This routine only needs interpolation because the edges are of size and lie in between cell faces.

## myEdge2Node

Edge to node requires interpolation since the nodes are not necessarily midway between the edges.

Solving for we have

Where

The edge values in the fictitious cells are assumed to be unknown. Therefore, this routine requires forward and backward extrapolation from the first and last interior edges to the first and last nodes.

### Backward Extrapolation

Solving for yields

Where

### Forward Extrapolation

Solving for yields

Where

## myNode2Edge

Node to edge requires an ordinary average for interior data as well as extrapolation to get the fictitious edges.

### Backward Extrapolation

Solving for yields

Where

### Forward Extrapolation

Solving for yields

Where

## myFace2CellCenter

Let be the cell center value between and . The same formulas for the exterior extrapolation must be used as the ones in the cellCenter2Face to ensure that applying them consecutively results in the original input.

Face to cell center is simple average since the cell centers are mid way between the faces

To calculate the cell center on the exterior (fictive) cells, we must extrapolate:

### 1st order extrapolation

#### Backward Extrapolation

From the cellCenter2Face, we had

Solving for yields

Where

#### Forward Extrapolation

Where

### 2nd order extrapolation

#### Backward Extrapolation

Approximating several function values around we have

From the cellCenter2Face, we had

Solving for yields

Where

#### Forward Extrapolation

Where

### 1st order extrapolation (Taylor Expansion)

#### Backward Extrapolation

Let's approximate and about by Taylor expansions:

Now let's eliminate :

Solving for we have

Where

#### Forward Extrapolation

Where

## myCellCenter2Face

Let be the cell center value between and . Cell center to face requires a linear interpolation since the cell faces are not necessarily midway between two cell centers.

Solving for the face value yields

Where

## myCellCenter2Node

Cell centers to node requires linear interpolation between the cell centers in the x,y and z directions. This was accomplished by using the cell center2Face interpolation along all directions.

## myFace2Node

Face to node requires linear interpolation between two faces to get edge data, then again to land on the node.

# Derivative operators

## Face-based operators

myFaceAdvect

myFaceCurl

myFaceDiv

myFaceLap

## Node/Cell center based operators

myDiv

myCurl

myAdvect

myLap