Interface Logic for TAM-ALE3D

GasTurbnLab

Dana Gottfried and Sanford Fleeter

School of Mechanical Engineering
Purdue University

Outline

Code basics relating to inflow/exit boundary conditions

Ordering of inflow/exit boundaries

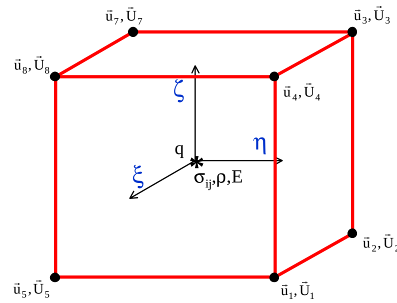
The four boundary algorithms required for the interface

Data storage structure

Element Variables

KINEMATIC (displacement, velocity, acceleration)

- stored at nodes
- vary tri-linearly over the element



$$\vec{U}(\xi,\eta,\zeta) = \vec{a} + \vec{b}\xi + \vec{c}\eta + \vec{d}\zeta + \vec{e}\xi\eta + \vec{f}\eta\zeta + \vec{g}\zeta\xi + \vec{h}\xi\eta\zeta$$

Stress, Density, Energy, q

- stored at element center
- constant over the element

Marching Forward in Time

Three separate procedures for each step forward in time

- Lagrange calculation
  - * Mesh moves with the material according to $\vec{F} = M\vec{a}$
- Mesh relaxation
  - * Move mesh back to original position or to any arbitrary position
- Advection calculation
  - * Find new nodal and elemental values for new mesh position

Mesh Relaxation

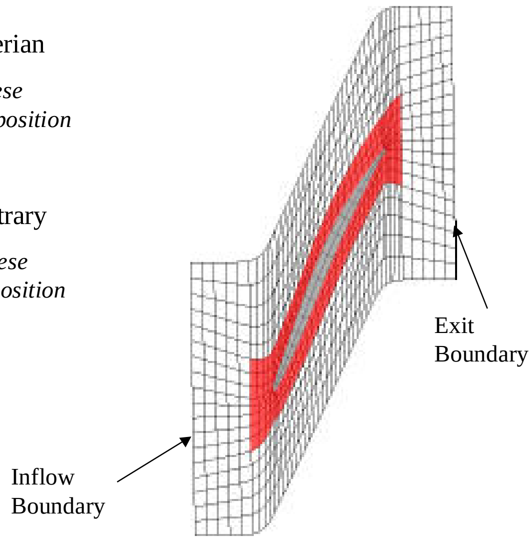
- Outer flow mesh is Eulerian

  Mesh relaxation moves these elements back to original position

- Inner flow mesh is arbitrary

  Mesh relaxation moves these elements to an arbitrary position

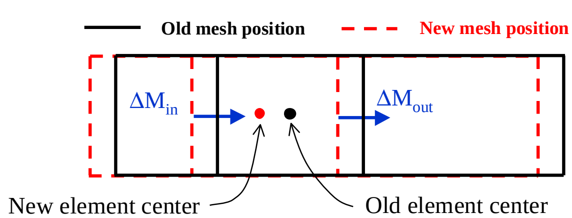
- Blade



Marching Forward in Time

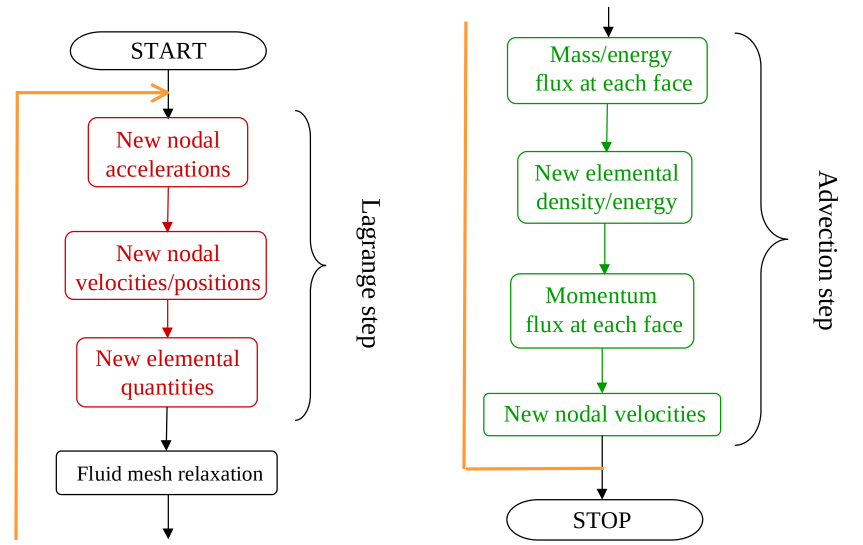
Advection

- Flow field is frozen
- Mesh moves through the flow

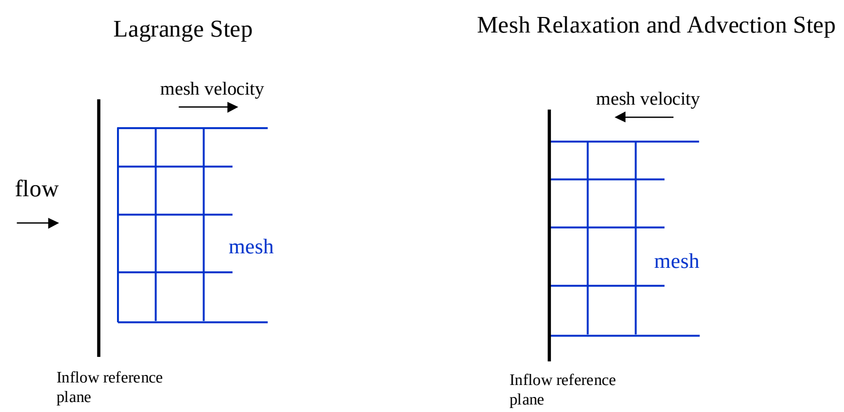


- New element mass = old element mass + ΔM_{in} - ΔM_{out}

TAM-ALE3D Flowchart



Inflow Boundary



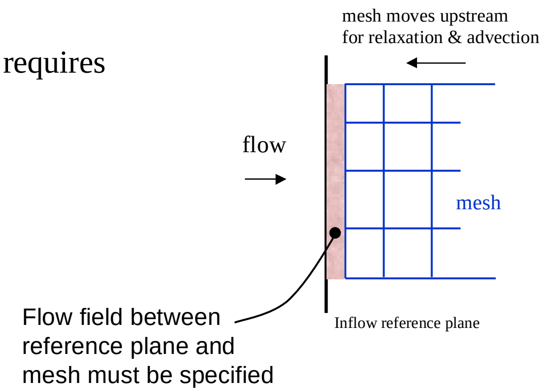
Mesh moves away from inflow reference plane during Lagrange step

Mesh moves back to inflow reference plane during advection step

Inflow Boundary

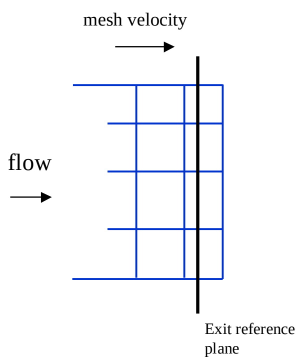
Required Data at Inflow -- the whole flow field

- Lagrange step requires
  - Inflow velocity
- Advection step requires
  - Inflow velocity
  - Inflow density
  - Inflow energy



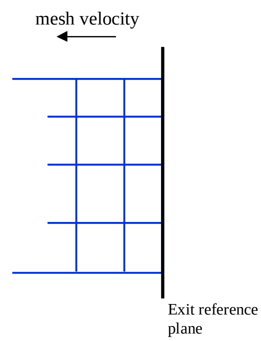
Exit Boundary

Lagrange Step



Mesh moves through
exit reference plane
during Lagrange step

Mesh Relaxation and Advection Step

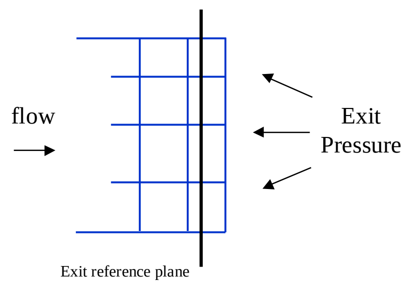


Mesh moves back to
exit reference plane
during advection step

Exit Boundary

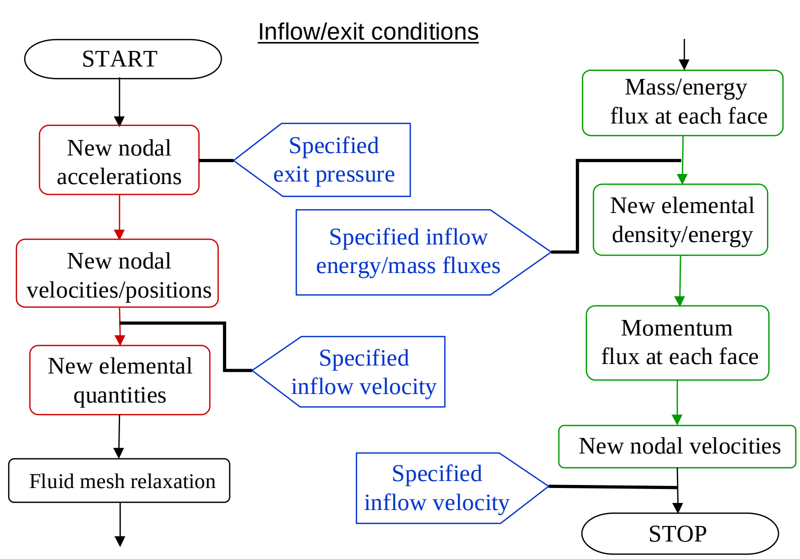
Required Data at Exit -- pressure

- Lagrange step requires
  pressure
- Advection step requires
  nothing



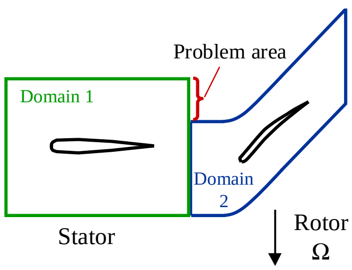
Exit pressure constrains the motion of the flow field at the exit boundary

Inflow/Exit Boundary Conditions

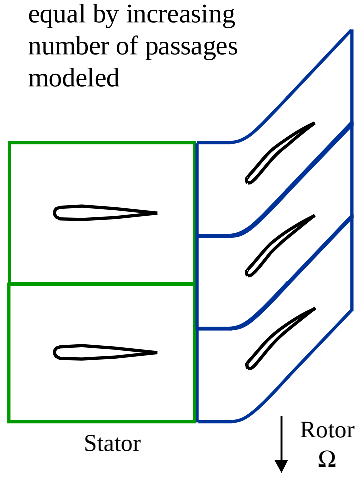


Blade Count Restrictions

Blade rows with different blade counts cannot be modeled with single blade passages



Interface area made equal by increasing number of passages modeled



Coordinate System

<u>Primary Coordinates</u>

Calculations performed in XYZ system

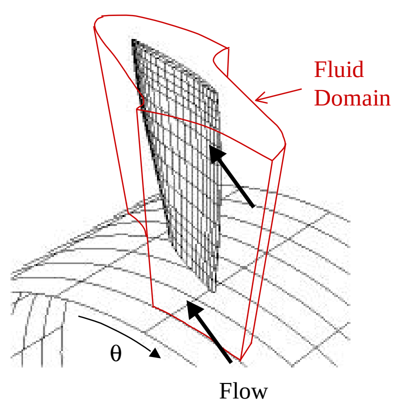
Z is along the engine axis

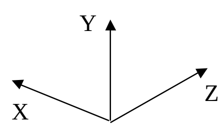
Inflow/Exit planes are planes parallel to the XY plane (constant Z)

<u>Secondary Coordinates</u>

$\theta = \tan \frac{Y}{X}$ measured from X axis

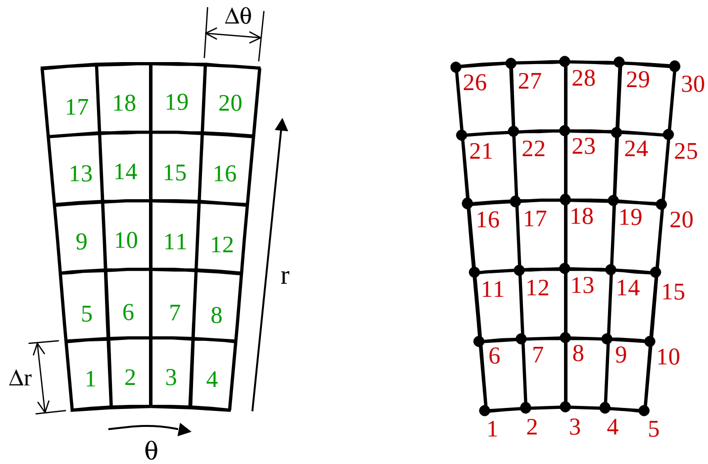
$r = \sqrt{X^2 + Y^2}$





Inflow/Exit Boundary Ordering

Face numbering | Node numbering



Inflow/Exit Boundary Ordering

Face numbering scheme start from lowest θ and r

- Inner loop -- increase through θ
- Outer loop -- increase through r

Assumptions for easier domain matching

- Inflow/exit are parallel to the XY plane
- Inflow/exit have the same number of faces
- Inflow/exit faces are generated with uniform Δθ and Δr

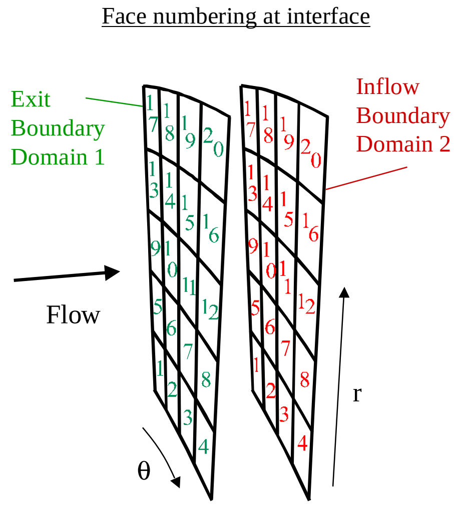
Nodes and faces with common radii are “on the same row”

Inflow/Exit Interface

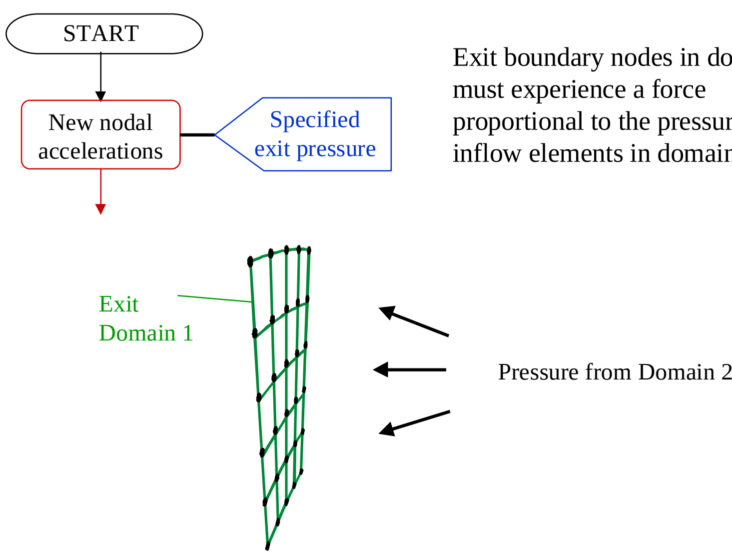
At time t = 0 inflow/exit faces are aligned

Domains rotate w.r.t. each other at angular rate Ω

As time increases the domains are offset from each other by $\theta = \Omega t$



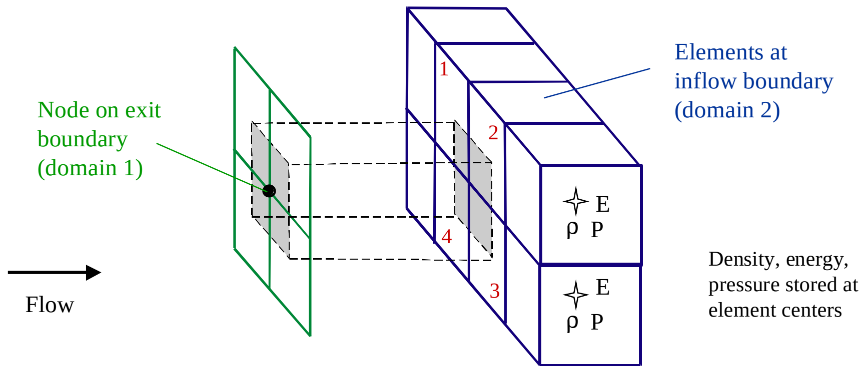
Exit Pressure Condition



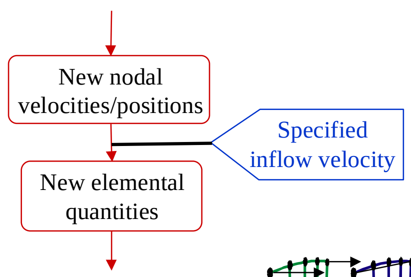
Exit boundary nodes in domain 1 must experience a force proportional to the pressure of the inflow elements in domain 2

Exit Pressure Condition

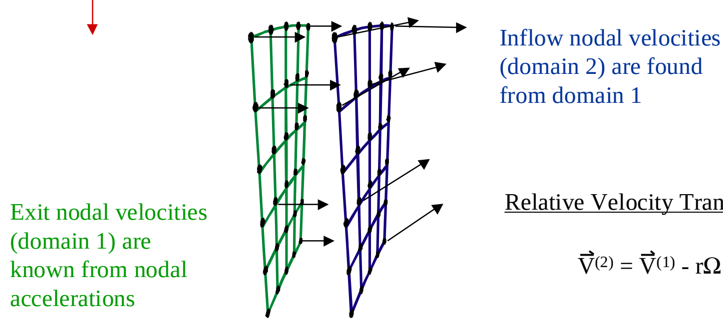
- For each exit boundary node (domain 1), locate four adjacent inflow elements (domain 2)
- Force on exit node due to pressure from inflow elements = $\sum_{i=1}^{4} P_i A_i$



Inflow Velocity Condition



Velocity of inflow nodes of domain 2 determined from velocity of exit nodes in domain 1



Relative Velocity Transformation

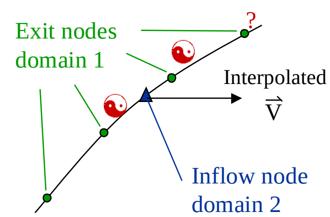
$$\vec{V}^{(2)} = \vec{V}^{(1)} - r\Omega\, \hat{i}_\theta$$

Inflow Velocity Condition

- Interpolate from nodes on the same radius
- Location of inflow boundary nodes relative to exit boundary nodes is a function of Ωt
- Interpolation method

Lower order method
Use the two nearest (checked) nodes with linear interpolation

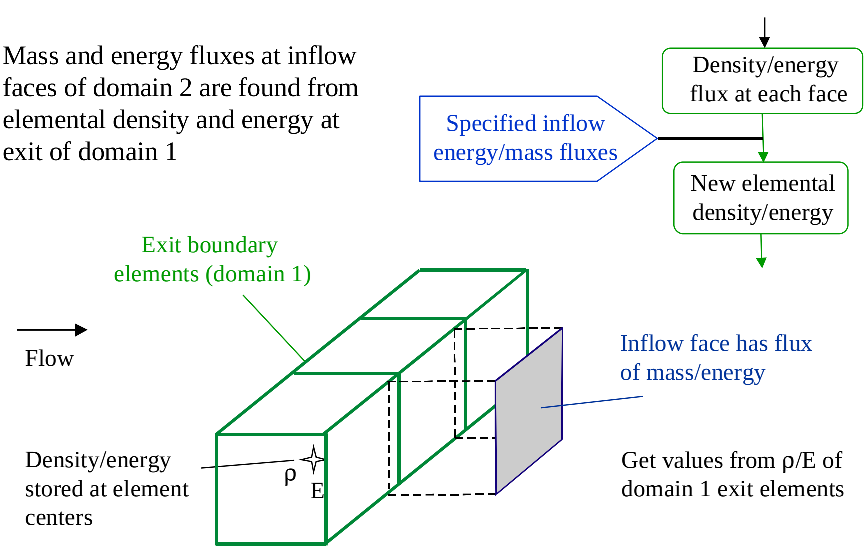
Higher order method
Use the two nearest (checked) nodes and next nearest node with quadratic interpolation



Inflow velocity transfer

Inflow Mass/Energy Flux Condition

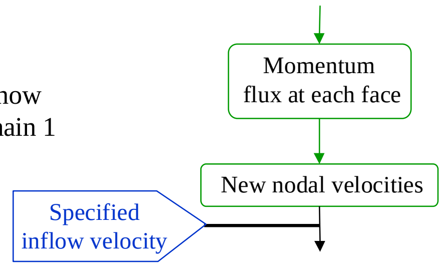
Mass and energy fluxes at inflow faces of domain 2 are found from elemental density and energy at exit of domain 1

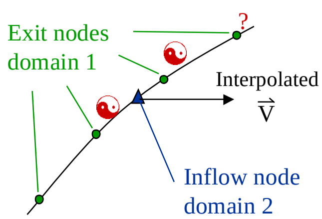


Inflow Velocity Condition after Advection

Advection found new nodal velocities

Inflow nodal velocities of domain 2 are now updated with velocities from exit of domain 1

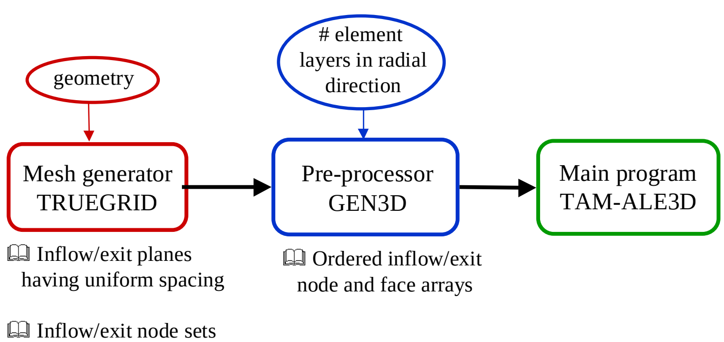




Use same interpolation method as used in previous velocity boundary condition

Data Input

Running a new case is a three step process



Data Storage Structure

Nodes

ninflnds -- number of inflow nodes (scalar)
nixflnds -- number of exit flow nodes (scalar)
inflnds() -- ordered list of inflow nodes (vector)
ixflnds() -- ordered list of exit flow nodes (vector)

Faces

ninflfce -- number of inflow faces (scalar)
nixflfce -- number of exit flow faces (scalar)
inflfce() -- ordered list of inflow faces (vector)
ixflfce() -- ordered list of exit flow faces (vector)

Other

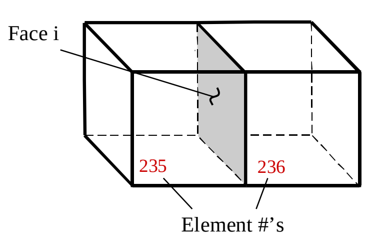
nlayers -- number of layers of elements in radial dir. (scalar)

Data Storage Structure

Element number

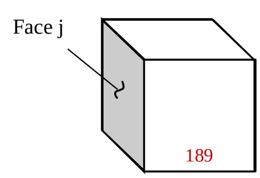
Derived from face numbers via *lmfa()* or *lmfb()*

Interior Face



lmfa(i) = 235 lmfb(i) = 236

Boundary Face



lmfa(j) = lmfb(j) = 189

Data Storage Structure

Scalars stored in common block *com.inc*

To include this common in a subroutine

```
include 'com.inc'
```

Vectors stored in individual common statements

Common statement in *.p file

```
arga  [inflnds]
```

Common statement in *.f file

```
pointer   ( iinflnds , inflnds (1) )
common /inflnds/ iinflnds
integer   inflnds
```

Example

For inflow faces find

1) pressure of associated element, 2) z - vel. of four nodes around the face

```
Subroutine example

arga [inflfce]
arga [p]
arga [zd]
arga [lmfa]
arga [inflnds]
include 'com.inc'

nnoderow = ninflnds / (nlayers + 1)
nfacerow = ninflfce / nlayers

do i = 1, ninflfce

    iskip = (i - 1) / nfacerow
    node1 = inflnds(i                    + iskip)
    node2 = inflnds(i + 1                + iskip)
    node3 = inflnds(i + 1 + nnoderow + iskip)
    node4 = inflnds(i       + nnoderow + iskip)

    iz = lmfa(inflfce(i))

    pressure = p(iz)

    velocity_node1 = zd(node1)
    velocity_node2 = zd(node2)
    velocity_node3 = zd(node3)
    velocity_node4 = zd(node4)
```

Summary

Inflow/exit faces have been ordered

Four interface algorithms must be developed

- Pressure transfer during Lagrange step *(domain 2 → 1)*
- Velocity transfer after Lagrange step *(domain 1 → 2)*
- Density/Energy transfer during advection step *(domain 1→2)*
- Velocity transfer after advection step *(domain 1 → 2)*