TAS Grid Data Structure

```
IMPLICIT REAL*8(A-H, O-Z)
READ(*) N_P_F, N_EL_F, N_ED_F, IDUMMY, N_BD_F, IDUMMY, DUMMY, &
        IDUMMY, IDUMMY, IDUMMY, IDUMMY, IDUMMY, IDUMMY, &
        IDUMMY, IDUMMY, IDUMMY, N_PRISM, N_PYR, N_RECT, N_HEX
READ(*) ((DS_ND(N,I), N=1,3), I=1,N_P_F)          :x,y,z coordinates
READ(*) ((IDS_EL(N,I), N=1,6),I=1,N_EL_F)
READ(*) ((IDS_ED(N,I),N=1,2),I=1,N_ED_F)
READ(*) ((IDS_BD(N,I),N=1,4),I=1,N_BD_F)
READ(*) ((JPRISM(N,I),N=1,9),I=1,N_PRISM)
READ(*) ((IPYR(N,I),N=1,5),I=1,N_PYR)
READ(*) ((JPYR(N,I),N=1,8),I=1,N_PYR)
READ(*) ((IDS_RECT(N,I),N=1,5),I=1,N_RECT)
READ(*) ((IDS_HEX(N,I),N=1,12),I=1,N_HEX)
```

Please note that TAS Hex format
is different from the TAS original format.

N_P_F: total number of nodes
N_EL_F: total number of tetrahedra
N_ED_F: total number of edges
N_BD_F: total numberof boundary triangles
N_PRISM: total number of prisms
N_PYR: total number of pyramids
N_RECT: total number of boundary rectangles
N_HEX: total number of hexahedra

TAS Grid Data Structure

((DS_ND(N,I), N=1,3), I=1,N_P_F)

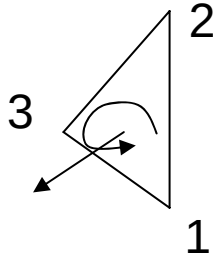
Node coordinates x(N=1), y(N=2), z(N=3)

((IDS_ED(N,I),N=1,2),I=1,N_ED_F)

Edge ID number → Node ID number

((IDS_BD(N,I),N=1,4),I=1,N_BD_F)

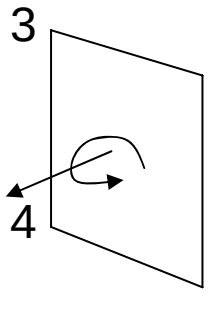
Node ID number of triangular boundary surface（N=1,2,3）
Zone ID number（N=4）



The direction determined by the right-handed screw rule shows the inward direction of domain.

((IDS_RECT(N,I),N=1,5),I=1,N_RECT)

Node ID number of rectangular boundary surface（N=1,2,3,4）
Zone ID number（N=5）



The direction determined by the right-handed screw rule shows the inward direction of domain.

Tetrahedra Data Structure

IDS_EL(n,i) n=1,6: Edge 1-6 are edges that construct a tetra, i.

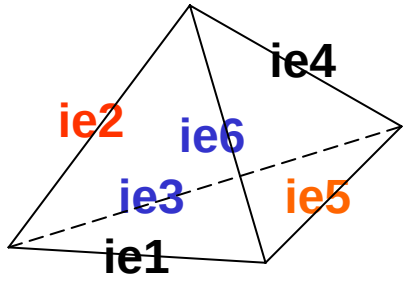
Rule

ie1-ie4, ie2-ie5, ie3-ie6 are skew lines with each other.

in01=ids_ed(in1_ed,ie1)
in02=ids_ed(in2_ed,ie1)
in03=ids_ed(in1_ed,ie4)
in04=ids_ed(in2_ed,ie4)

in11=ids_ed(in1_ed,ie2)
in12=ids_ed(in2_ed,ie2)
in13=ids_ed(in1_ed,ie5)
in14=ids_ed(in2_ed,ie5)

in21=ids_ed(in1_ed,ie3)
in22=ids_ed(in2_ed,ie3)
in23=ids_ed(in1_ed,ie6)
in24=ids_ed(in2_ed,ie6)



in01-04, in11-14, in21-24 are nodes that construct a tetrahedron, i.

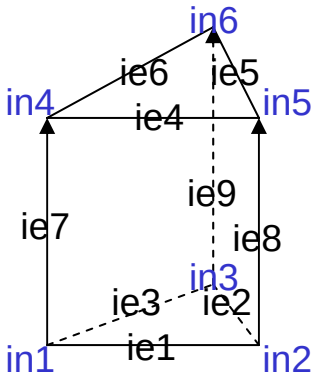
Prism Data Structure

JPRISM(n,i) n=1,9: Edge 1-9 are edges that construct a prism, i.

Rule

One triangular face is constructed by edges of ie1,ie2, and ie3 and the other is by edges of ie4, ie5, and ie6.

in1=ids_ed(in1_ed,ie7)
in2=ids_ed(in1_ed,ie8)
in3=ids_ed(in1_ed,ie9)
in4=ids_ed(in2_ed,ie7)
in5=ids_ed(in2_ed,ie8)
in6=ids_ed(in2_ed,ie9)



Nodes of in1-in3 construct one triangular face, Nodes of in4-in6 construct the other one.

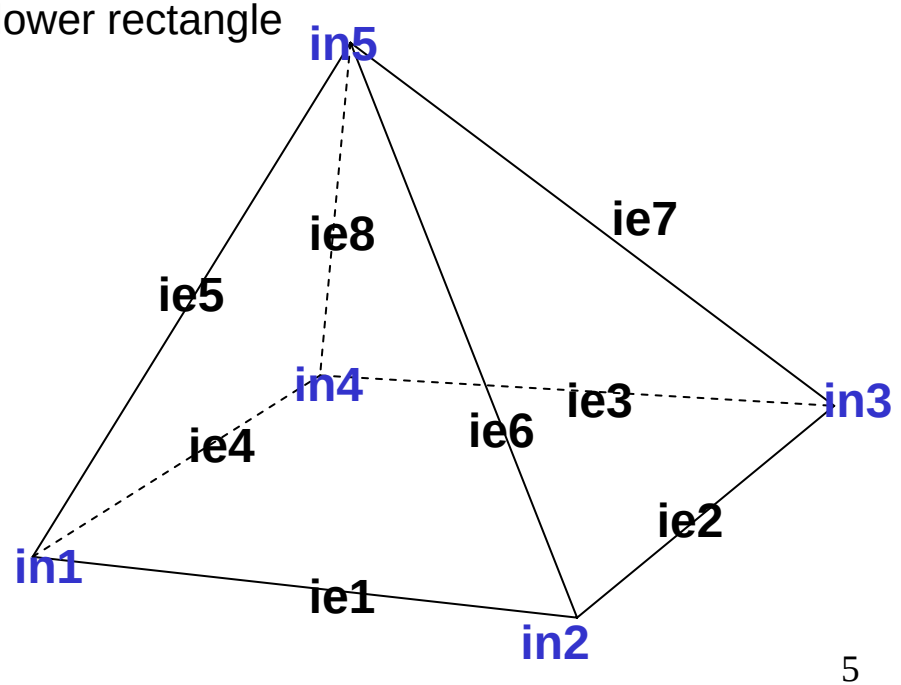
Pyramid Data Structure

IPYR(n,i) n=1,5: Node 1-5 are nodes that construct Pyramid, i.

JPYR(n,i) n=1,8: Edge1-8 are edges that construct Pyramid, i.

Rule

ie1, ie2, ie3, ie4:edges that construct a lower rectangle
ie5, ie6, ie7, ie8:edges that construct four side triangles
in1, in2, in3, in4:nodes that construct a lower rectangle

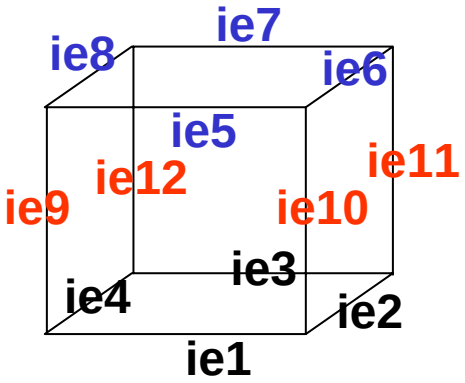


Hexahedra Data Structure

IDS_HEX(n,i) n=1,12: Edges 1-12 are edges that construct Hexa, i.

Rule

ie1-ie4 construct one face
ie5-ie8 construct another face
ie9-ie12 construct the other face



in1=ids_ed(in1_ed,ie9)
in2=ids_ed(in1_ed,ie10)
in3=ids_ed(in1_ed,ie11)
in4=ids_ed(in1_ed,ie12)
in5=ids_ed(in2_ed,ie9)
in6=ids_ed(in2_ed,ie10)
in7=ids_ed(in2_ed,ie11)
in8=ids_ed(in2_ed,ie12)
Nodes of in1～in4 construct one face, nodes of in5～in8 construct another face.