if one calculates the "vicinity" of mesh:

```
(F) ".VGE"          :  sd_voisinage
```

Data structures sd_maillage, sd_voisinage, sd_squelette and sd_grille

Contents

1 General information

a mesh (or `sd_maillage`) is a set of meshes of predefined types: `HEXA8`, `TRIA3`,…

These meshes are defined by nodes list which have coordinates. These are the nodes which connect meshes between them. All the coordinates of the nodes of the mesh form a field at nodes of the quantity “geometry” (`cham_no/GEOM_R`).

A mesh can also contain named mesh groups and nodes groups. These groups are unspecified: a mesh (for example) can belong to $0,1,2,\ldots,n$ groups.

Let us announce that for the static substructuring, a mesh can contain the superones (meshes having an unspecified number of nodes).

When the mesh is made of linear elements forming line continuous, it can contain a card of the mesh containing for each mesh the curvilinear abscisse of each node.

A squelette (or `sd_squelette`) is a mesh of visualization of the results for dynamic substructures. It is a mesh with some objects moreover.

A grid (or `sd_grille`) is a typical case of mesh for which all the nodes are aligned according to the directions of a local base.

2 Tree structure of the Data structures

```
sd_maillage (K8):: = record

        (O) ".DIME"     :    OJB      S      V       I
            ".NOMNOE"   :    OJB      S      N       K8
            ".COORDO"   :    sd_cham_no (GEOM_R)

        (F) ".GROUPENO":    OJB      XD   V       I      NO ()
        (F) ".PTRNOMNOE":    OJB      S      N       K24 %
```

if the mesh contains meshes:

```
|    (O) ".NOMMAI"  :  OJB      S      N      K8
         ".TYPMAIL":  OJB      S      V      I
         ".CONNEX"  :  OJB      XC     V      I      NU ()

     (F) ".GROUPEMA": OJB      XD     V      I      NO ()
     (F) ".PTRNOMMAI": OJB      S      N      K24 %
```

if the mesh contains the superones (static substructuring):

```
|    (O) ".NOMACR"  :  OJB      S       V     K8
         ".PARA_R"  :  OJB      S       V     R
         ".SUPMAIL":  OJB      XD      V     I      NO ()
     (F) ".TYPL"    :  OJB      S       V     I
```

% if the mesh contains a card of curvilinear abscisse:

```
(F) ".ABS_CURV"    :  sd_carte (ABSC_R)
```

% if the mesh were refined by Homard:

```
(F) ".ADAPTATION"  :  OJB    S     V    I    LONG= 1%
```

if the mesh were read with format "`MED`" :

```
                (F) "(11) .FORMES"    :  OJB    S     V    K32  LONG=2
```

```
sd_voisinage (K12)       :: = record

          (O)  "PTVOIS"          :   OJB  V  I
          (O)  "ELVOIS"          :   OJB  V  I

sd_squelette (K8)        :: = record

          (O)  "$vide"           :   sd_maillage
          (F)  ".INV.SKELETON"   :   OJB  V  I
          (F)  ".CORRES"         :   OJB  V  I
          (F)  "(9) .NOMSST"      :   OJB  V  K24 %

          if DEFI_SQUELETTE/Mesh:
          (F)    ".ANGL_NAUT"    :   OJB  V  R   dim = 3
                 ".TRANS"        :   OJB  V  R   dim = 3

sd_grille (K8)        :: = record

          (O)  "$vide"           :   sd_maillage
          (O)  ".GRLI"           :   OJB  V  I
          (O)  ".GRLR"           :   OJB  V  R
```

3 Contained jeveux objects

3.1 Object .DIME

".DIME" : S V I LONG = 6

```
V (1):    nb_no  :     many physical nodes of the mesh
V (2):    nb_nl  :     many nodes of Lagrange of the mesh
       nb_nl > 0:  there exist the superones: static substructuring (sss)
V (3):    nb_ma  :     number of meshes of the mesh
V (4):    nb_sm  :     many super-meshes of the mesh
       nb_sm  > 0:  static substructuring
V (5):    nb_sm_mx:  raising amongst super-meshes
V (6):    dim_coor:     /2  (if mesh 2D)
                     /3  (if mesh 3D)
```

3.2 Object .NOMNOE

".NOMNOE" : S N K8 LENGTH = nb_no

It is the pointer of names giving the correspondence:

nom_de_nœud ↔ numéro_de_nœud

3.3 Object .GROUPENO

".GROUPENO" : XD V I NO () VARI NB_OJB = nb_gno

- number of group_no nb_gno = NUTIOC (".GROUPENO")

Is V = ".GROUPENO" (nom_gno)

nb_no_gno = many nodes of nom_gno = LONUTI (V)

for I = 1, nb_no_gno

V (I): number of I ème node of nom_gno

important Remark :

A group_no can be empty (0 node). An empty group is represented by a vector of LONMAX=1 and LONUTI=0. When one wants to know the number of elements of a group, JELIRA/LONUTI thus should be used. When a group is created, it is necessary to think of informing (JEECRA) two attributes (LONMAX and LONUTI).

Caution:

The number of group_no of a mesh can change: one can modify a mesh (command DEFI_GROUP) to add group_no to him.

3.4 Object .PTRNOMNOE

"**. PTRNOMNOE** " : S N K24 LONG = nb_grno

It is the external pointer of name being used to store the names of nodes groups.

3.5 Object .NOMMAIL

"**.NOMMAI**" : S V K8 LENGTH = nb_ma

It is the pointer of names giving the correspondence:

nom_de_maille ↔ numero_de_maille

3.6 Object .TYPMAIL

"**.TYPMAIL**" : XC E I NO ("$.NOMMAI") NB_OJB = nb_ma

Is E = ".TYPMAIL" (nom_mail)

E : number of the type of mesh associated with the mesh with name: nom_mail

- the type of mesh is a name defined in the /compelem/typmail catalog : SEG2, TRIA3, QUAD4,..., HEXA20
- the known types of mesh of Aster are described in [U3.01]
- the correspondence: nom_de_type_de_maille ↔ numero_de_type_de_maille is accessible by the pointer from name: "&CATA.TM.NOMTM" cf [D4.04.01]. Object .CONNEX

3.7 ".CONNEX " :

XC V I NO ("$. NAMED ") NB_OJB = nb_ my Is V = ".

CONNEX " (nom_mail) V (1): number

of the 1 1st node of the mesh of name: nom_mail... V (N): number

of the last node of the mesh of name: nom_mail N = many nodes of nom_mail = LONMAX (V) the number of

nodes of a mesh is always the number of nodes associated with the type of mesh which to him is attached (see object "&CATA.TM.NBNO" [D4.04.01]) Object .GROUPEMA ".GROUPEMA

3.8 ": XD V I

NO () VARI NB_OJB = nb_ gma number of group_ma nb_

- gma = NUTIOC (".GROUPEMA ") Is V = ".GROUPEMA" (name

_gma) nb_ma_gma = number of meshes

of nom_gma = LONUTI (V) for I = 1, nb_ma_gma V

(I): number of the ième

mesh of nom_gma important Remark :

A group_ma can be

empty (0 mesh). An empty group is represented by a vector of LONMAX=1 and LONUTI=0. When one wants to know the number of elements of a group, JELIRA/LONUTI thus should be used. When a group is created, it is necessary to think of informing (JEECRA) two attributes (LONMAX and LONUTI). Caution: The number of

GROUP_MA D

"a mesh can change : one can modify a mesh (command DEFI_GROUP) to add group_ma to him. Object .PTRNOMMAI ". PTRNOMNOE

3.9 ": S N

K24 LONG = Nb _grma It is the external pointer of

name being used to store the names of mesh groups. Objects .NOMACR, .PARA_R,

3.10 .SUPMAIL and. TYPL ".NOMACR ": S V K8 LENGTH

= nb_sm for I = 1 , nb_sm V (I)

: name of MACR_ELEM

_STAT associated with super-mesh ".PARA_R": S V R LONG i

= 14*nb_ Sm for I = 1, nb_sm: V (

```
14* (I 1) +1): TX V (14*
    (i-1) +2): TY V (14*
    (i-1) +3): TZ V (14*
    (i-1) +4): alpha V (
    14* (i-1) +5): beta V (14
    * (i-1) +6): gamma V (
    14* (i-1) +7): PX V (14*
```

(i-1) +8): PY V (14*
(i-1) +9): PZ V (14*
(i-1) +13): dmini V
(14* (i-1) +14): dmaxi (
TX, TY, TZ) is the values

- of translation of the geometrical transformation associated with super-mesh I (alpha, beta, gamma) are
- the nautical angles (in radians) defining the rotation of the geometrical transformation, (PX, PY, PZ) defines
- the preceding center of rotation. That is to say macrost MACR_ELEM

_STAT associated with super-mesh I, the position the super one - mesh I am defined by isometry of the nodes of macrost. The isometry is the composition in the order: rotation THEN translation. Dmini: minimal distance

- between 2 nodes of the mesh I, dmaxi: maximum distance
- between 2 nodes of the mesh I. ".SUPMAIL": XD V I NO

() VARI NB_OJB = nb_ Sm Is V = ".SUPMAIL" (name

_sma) V is a vector containing

the numbers of the nodes of the super-mesh nom_sma. The nodes of a super-mesh can be of "physical" type or type "Lagrange". That is to say: inop a "physical

"

number of node of the super-mesh nom_sma inol a number of node
"Lagrange " of the super-mesh nom_sma 1 ≤ inop ≤ nb_no Nb
_no + 1 ≤ inol ≤
nb_no + Nb _nl V is the shape of recopy

of object ".CONX" of the MACR_ELEM_STAT [D4.08 .01]. V the connectivity of

super-meshes defines. The superones "are restuck

" by "physical" nodes. The nodes of "Lagrange"

inherited the MACR_ELEM_STAT are never common to several super-meshes. Nombre total of nodes (

"physical" + "Lagrange") of nom_sma = LONMAX (V) the pointer of names (intern

) of object ".SUPMAIL" gives the correspondence : number (super_maille) ↔

name (super_maille) ". TYPL ": S V I LONG

= nb_nl for I = 1 , nb_nl V (I)

: /-1 if the node of
"Lagrange " I am of type "before" /-2 if the node of "Lagrange
" I am of type "after" Object .ADAPTATION ".ADAPTATION

3.11 ": S V I

LONG = 1 V (1) : level of adaptation (by Homard) of the mesh. This level is 0 in general

. It is non-zero only if the mesh were refined by Homard. This information is used only by Homard. Object (11) .FORMES "(11) .FORMES

3.12 ": S V LONG

K32 = 2 This object exists only if

the mesh were read with format "MED" V (1): "MED". V (2) : name
of mesh
MED. Objects .PTVOIS and .ELVOIS

3.13 These objects describe

the neighbors of meshes of a mesh. Each neighbor (i.e close

mesh) has a type: 3D PAR FACE: F3: 1 2D PAR

```
FACE: F2     : 2 3D PAR
STOPS: A3     : 3 2D
PAR STOPS: A2   : 4 1D
PAR STOPS: A1   : 5 3D
PAR SOMMET: S  3: 6 2D
PAR SOMMET: S  2: 7 1D
PAR SOMMET: S  1: 8 0D
PAR SOMMET: S  0: 9 One
stores in  objects
```

.PTVOIS and .ELVOIS: For any M0 mesh of mesh:

Nombre total of neighbors NVTOT
For each neighbor MV : type
of the close number of the mesh
many
common tops between
M0 and MV: nso_com For is = 1, nso_com: Local
number in M0 Numéro
local in MV Specificity
of the data structure

4 squelette a squelette being the mesh

of restitution of a computation per substructuring, the data structure squelette is copied on that of data structure of a mesh. One finds all the elements described there previously. One adds object ".INV.SKELETON to it" in order to specify , for each node of the squelette, substructure of which he belongs in the beginning and to which node of this substructure he corresponds: That is to say nb_no the number of nodes

of the mesh included in the squelette. ".INV.SKELETON": OJB V I

LONG = 2 * nb_ No Is V = ".INV.SKELETON" for

I = 1, nb_no V (I):

number of the substructure
of origin of node I for I = nb_no + 1,2 * Nb
_no V (I): number of node I in
its substructure of origin If the squelette were obtained

from a squelette (DEFI_SQUELETTE/RECO_GLOBAL): ".CORRES": OJB V I LONG

= nb_no-nbmoi It is an optional object

, it exists only when one creates a squelette by modifying old (key word SQUELETTE is indicated in the catalog of the command). In this case, one will amalgamate the nodes of the interfaces (all interfaces or only those given by the user) according to a criterion indicated behind the key word CRITERE. One thus carries out the "resticking ". Here nbmoi is the number of nodes amalgamated with the dynamic interfaces (does not exist any more in the new squelette). For the node ino of the squelette
, .CORRES (ino) is the number of the node of the initial squelette (before "resticking"). ".NOMSST": LONG OJB V K8

= nbss optional Object which exists

in the case D "a squelette known as "classical" (initial squelette lorsqu" one informs the substructures which will form it). Its length is equal to the number of the substructures which form the squelette. It contains the names of the substructures
which form the squelette. Note: This object is not useful

apart from
command DEFI_SQUELETTE. Nevertheless, it must (unfortunately!) belong to the sd_squelette. Indeed, if one makes: SQUEL1= DEFI_SQUELETTE (...)

```
SQUEL2= DEFI_SQUELETTE (…,
SQUELETTE=SQUEL1,…) Object SQUEL1.NOMSST
```

will be used to build SQUEL2. If the squelette were obtained

from a mesh (DEFI_SQUELETTE/MAILLAGE): ".ANGL_NAUT": V K8 LENGTH =

3 nautical angles of the rotation

of the squelette compared to the mesh from which it is resulting. ".TRANS": V K8 LENGTH = 3 coordinates

of the vector which

gives the translation of the squelette compared to the mesh from which it is resulting. Specificity of structure

5 of given grid the specificity of a grid

is that its nodes are aligned according to the directions of a local base of the grid. This kind of mesh is used with algorithms based on methods with the finite differences. Consequently, the local base, the array of connection of $(X_{loc}, Y_{loc}, Z_{loc})$ the nodes and the distance between two aligned nodes of the grid must be stored by adding this information in the sd_maillage. That is to say nb_no the number of nodes of the mesh which form the grate. ".GRLI": OJB V I LONG =

6*nb_no In this object one stores

the array of connection of the nodes of the grid. That is to say V = ".GRLI": for I = 1: Nb _no V (6* (I

1) +1) : node following
the node according to the direction of the vector i V (6* (i-1) +2): preceding node X_{loc}
the node according to the direction of the vector i V (6* (i-1) +3): node following X_{loc}
the node according to the direction of the vector i V (6* (i-1) +4): preceding node Y_{loc}
the node according to the direction of the vector i V (6* (i-1) +5): node following Y_{loc}
the node according to the direction of the vector i V (6* (i-1) +6): preceding node Z_{loc}
the node according to the direction of the vector i If the node does not have a following Z_{loc}

or preceding i node in one of the three directions, the element corresponding of the vector is put at zero. In this case V the node is on the free surface of i the grid. ".GRLR": OJB V R LONG =

10+6*nb _no In this object one stores edge

of the grid the length of the smallest, the local base formulates and the distance between $(X_{loc}, Y_{loc}, Z_{loc})$ nodes of the grid. That is to say W (1): length of the $W='.GRLR'$

smallest edge of the grid W (2): component of the first

vector of the local X base of the grid X_{loc} W (3): component of the first
vector of the local Y base of the grid X_{loc} W (4): component of the first
vector of the local Z base of the grid X_{loc} W (5): component of the second

vector of X the local base of the grid Y_{loc} W (6): component of the second
vector of Y the local base of the grid Y_{loc} W (7): component of the second
vector of Z the local base of the grid Y_{loc} W (8): component of the third

vector C of X the local base of the grid Z_{lo} W (9): component of the third
vector of Y the local base of the grid Z_{loc} W (10): component of the third
vector of Z the local base of the grid Z_{loc} for I = 1: nb_no W (10+6*

(i-1) + 1) : absolute value
of the distance enters the node and the node V (6* (i-1) +1) according to i according to the direction of the vector W (10+6* (i-1) +2): absolute value X_{loc}

of the distance enters the node and the node V (6* (i-1) +2) preceding i according to the direction of the vector W (10+6* (i-1) +3): absolute value X_{loc}

of the distance enters the node and the node V (6* (i-1) +3) according to i according to the direction of the vector W (10+6* (i-1) +4): absolute value Y_{loc}

of the distance enters the node and the node V (6* (i-1) +4) preceding i according to the direction of the vector W (10+6* (i-1) +5): absolute value Y_{loc}

of the distance enters the node and the node V (6* (i-1) +5) according to i according to the direction of the vector W (10+6* (i-1) +6): absolute value Z_{loc}

of the distance between the node and the node V (6* (i-1) +6) preceding i according to the direction of the vector If the node does not have a following Z_{loc}

or preceding i node in one of the three directions, the element corresponding of the vector is put at zero. In this case W the node is on the free surface of i the grid.