# Macro-order MACR_ADAP_MAIL

## 1 Goal

To adapt a grid with the software LOBSTER.

This operation is possible for a formed grid by mesh-points, segments, triangles, quadrangles, tetrahedrons, hexahedrons, pentahedrons. A field controlling the adaptation will have been possibly calculated. According to its value nets by mesh or node by node, or according to a geometrical directive, the software LOBSTER will modify the grid. It is also possible to interpolate fields, old grid towards the new one.

One can connect calculation and adaptation progressively in a process of improvement of calculation. This process can take place into only one passes, or divided into several stages by one CONTINUATION.

The software LOBSTER is presented on LE site: http://www.code-aster.org/outils/homard

One finds a description of the technique used there to modify the grids as well as examples.
More on LOBSTER, one can refer to the documents quoted in the bibliography.

Any external reference to LOBSTER must be done with:

G. Nicolas and T. Fouquet, " *Adaptive Mesh Refinement for Conformal Hexahedral Meshes* ", Finite Elements in Analysis and Design, vol. 67, pp. 1-12, 2013, DOI: 10.1016/j.finel.2012.11.008

# Contents

# 2 Syntax

```
MACR_ADAP_MAIL   (
#   choice of the type of adaptation
♦   ADAPTATION    =   /   'RAFF_DERA'
                    /   'REFINEMENT'
                    /   'DERAFFINEMENT'
                    /   'RAFF_DERA_ZONE'
                    /   'RAFFINEMENT_UNIFORME'
                    /   'DERAFFINEMENT_UNIFORME'
                    /   'NOTHING'
                    /   'MODIFICATION'
                    /   'READING'

#   grid to be modified
♦   MAILLAGE_N    =   man    [grid]

#   new grid
♦   MAILLAGE_NP1  =   Co (manp1)       [K8]

#   an additional grid
◊   MAILLAGE_NP1_ANNEXE =   Co (manp1ann)    [K8]

#   If the adaptation is free, (REFINEMENT, DERAFFINEMENT or RAFF_DERA), choice of the
structure containing the field controlling the adaptation:
    ♦   / RESULTAT_N = resun          [result]
          ♦ NOM_CHAM = nomsymb        [K16]
        / CHAM_GD = cham_gd_i         [cham_gd]
    ◊   NOM_CMP = l_cmp            [l_K8]
     #   Selection of the temporal parameter
        / NUME_ORDRE = order        [I]
        / INST = moment          [R]
            ◊ | PRECISION =  / prec [R]
                             / 1.0E-6  [DEFECT]
            ◊ | CRITERION =  / 'RELATIVE  [DEFECT]
                             / 'ABSOLUTE'
    ◊   USAGE_CMP =   / 'NORME_L2'  [DEFECT]
                      / 'ABSOLUTE'
                      / 'NORME_INFINIE'
                      / 'RELATIVE'
    ◊   USAGE_CHAMP =/ 'MESH'   [DEFECT]
                      / 'JUMP'
    ◊   ADAP_INIT =   /'TO KEEP'    [DEFECT]
                      /'TO REFINE'
                      / 'DERAFFINER'
#   Finsi

#   If the adaptation takes place according to geometrical zones, (RAFF_DERA_ZONE):
    ♦   ZONE = _F (
    #  Type of the zone
    ♦   TYPE = / 'RIGHT-ANGLED'
               / 'LIMPS'
               / 'DISC'
               / 'SPHERE'
               / 'CYLINDER'
               / 'DISQUE_PERCE'
               / 'PIPE'
```

```
        # Use of the zone
        ♦  USE =  / 'REFINEMENT'      [DEFECT]
                  / 'DERAFFINEMENT'

        #   for a rectangular box: extreme coordinates
            ♦ X_MINI = x_mini         [R]
            ♦  X_MAXI = x_maxi        [R]
            ♦  Y_MINI = y_mini        [R]
            ♦  Y_MAXI = y_maxi        [R]
        #   for a parallelepipedic box: extreme coordinates
            ♦  X_MINI = x_mini        [R]
            ♦  X_MAXI = x_maxi        [R]
            ♦  Y_MINI = y_mini        [R]
            ♦  Y_MAXI = y_maxi        [R]
            ♦  Z_MINI = z_mini        [R]
            ♦  Z_MAXI = z_maxi        [R]
        #   for a disc: centre and
            ♦  X_CENTRE  = x_centre       [R]
            ♦  Y_CENTRE  = y_centre       [R]
            ♦  RAY= ray           [R]
        #   for a sphere: centre and
            ♦  X_CENTRE  = x_centre       [R]
            ♦  Y_CENTRE  = y_centre       [R]
            ♦  Z_CENTRE  = z_centre       [R]
            ♦  RAY= ray           [R]
        #   for a cylinder: axis, bases, height and ray
            ♦  X_AXE  = x_axe             [R]
            ♦  Y_AXE  = y_axe             [R]
            ♦  Z_AXE  = z_axe             [R]
            ♦  X_BASE = x_base        [R]
            ♦  Y_BASE = y_base        [R]
            ♦  Z_BASE = z_base        [R]
            ♦  HEIGHT = height        [R]
            ♦  RAY= ray           [R]
        #   for a bored disc: centre, interior and external
            ♦  X_CENTRE  = x_centre       [R]
            ♦  Y_CENTRE  = y_centre       [R]
            ♦  RAYON_INT = rayon_int  [R]
            ♦  RAYON_EXT = rayon_ext  [R]
        #   for a pipe: axis, bases, height and rays interior and outside
            ♦  X_AXE  = x_axe             [R]
            ♦  Y_AXE  = y_axe             [R]
            ♦  Z_AXE  = z_axe             [R]
            ♦  X_BASE = x_base        [R]
            ♦  Y_BASE = y_base        [R]
            ♦  Z_BASE = z_base        [R]
            ♦  HEIGHT = height        [R]
            ♦  RAYON_INT = rayon_int  [R]
            ♦  RAYON_EXT = rayon_ext  [R]
        #   Finsi
            )
    #   Finsi


    #   If the adaptation includes free refinement (REFINEMENT or RAFF_DERA):
        ♦  /  CRIT_RAFF_PE =      crp [R]
           /  CRIT_RAFF_REL =  crr [R]
           /  CRIT_RAFF_ABS =  CRA [R]
```

```
      /  CRIT_RAFF_MS =   crms   [R]
#  Finsi


#  If the adaptation includes the déraffinement free one (DERAFFINEMENT or RAFF_DERA):
   ♦  /  CRIT_DERA_PE =      cdp    [R]
      /  CRIT_DERA_REL =  cdr    [R]
      /  CRIT_DERA_ABS =  half-value layer    [R]
      /  CRIT_DERA_MS =   cdms      [R]
#  Finsi

#  If the adaptation includes refinement:
   ◊  NIVE_MAX = nivmax             [I]
   ◊  DIAM_MIN = diamin             [R]
#  Finsi

#  If the adaptation includes déraffinement:
   ◊  NIVE_MIN = nivmin             [I]
#  Finsi

#  If the adaptation includes refinement or déraffinement:
   ◊  GROUP_MA = l_grma          [l_gr_maille]
   ◊  GROUP_NO = l_grno          [l_gr_nœud]
#  Finsi

#    Follow-up of a border with a grid
◊  MAILLAGE_FRONTIERE = maf          [grid]
   ◊  GROUP_MA_FRONT = l_grma    [l_gr_maille]

#    Follow-up of an analytical border
◊  FRONTIERE_ANALYTIQUE = _F  (
   # Name of the border
   ♦  NAME = name          [K]
   ♦  GROUP_MA = l_grma          [l_gr_maille]
   # Type of the border
   ♦  TYPE = / 'SPHERE'
             / 'CYLINDER'
             / 'CONE_A'
             / 'CONE_R'
             / 'TORUS'
   #    for a sphere: centre and
      ♦  X_CENTRE  = x_centre       [R]
      ♦  Y_CENTRE  = y_centre       [R]
      ♦  Z_CENTRE  = z_centre       [R]
      ♦  RAY = ray            [R]
   #  for a cylinder: axis, bases and ray
      ♦  X_AXE  = x_axe             [R]
      ♦  Y_AXE  = y_axe             [R]
      ♦  Z_AXE  = z_axe             [R]
      ♦  X_CENTRE  = x_centre       [R]
      ♦  Y_CENTRE  = y_centre       [R]
      ♦  Z_CENTRE  = z_centre       [R]
      ♦  RAY = ray            [R]
   #  for a cone defined by its angle: axis, center and angle
      ♦  X_AXE  = x_axe             [R]
      ♦  Y_AXE  = y_axe             [R]
      ♦  Z_AXE  = z_axe             [R]
```

```
        ♦  X_CENTRE  = x_centre       [R]
        ♦  Y_CENTRE  = y_centre       [R]
        ♦  Z_CENTRE  = z_centre       [R]
        ♦  ANGLE  = angle             [R]
    #   for a cone defined by its rays: centres and
        ♦  X_CENTRE  = x_centre       [R]
        ♦  Y_CENTRE  = y_centre       [R]
        ♦  Z_CENTRE  = z_centre       [R]
        ♦  RAY= ray            [R]
        ♦  X_CENTRE2 = x_centre2      [R]
        ♦  Y_CENTRE2 = y_centre2      [R]
        ♦  Z_CENTRE2 = z_centre2      [R]
        ♦  RAYON2 = rayon2            [R]
    #   for a torus: centre, axis and
        ♦  X_CENTRE  = x_centre       [R]
        ♦  Y_CENTRE  = y_centre       [R]
        ♦  Z_CENTRE  = z_centre       [R]
        ♦  X_AXE  = x_axe             [R]
        ♦  Y_AXE  = y_axe             [R]
        ♦  Z_AXE  = z_axe             [R]
        ♦  RAY= ray            [R]
        ♦  RAYON2 = rayon2            [R]
    #   Finsi
              )


#   If the adaptation is a modification, (MODIFICATION), choice of the type:
    ◊   Change of degree
        DEGREE =  / 'YES'
                  / 'NOT'          [DEFECT]
#   Finsi


#   Update of fields on the new grid
◊   MAJ_CHAM = _F(
    #   choice of the structure containing the field to be updated
    ♦   / RESULT = resu            [result]
           ♦ NOM_CHAM = nomsymb       [K16]
        / CHAM_GD = cham_gd           [cham_gd]
    ◊   NOM_CMP = l_cmp            [l_K8]
     #   Selection of the temporal parameter
        / NUME_ORDRE = order       [I]
        / INST = moment         [R]
           ◊ | PRECISION =  / prec [R]
                            / 1.0E-3  [DEFECT]
           | CRITERION =/ 'RELATIVE   [DEFECT]
                        /   'ABSOLUTE'
    #   choice of the type of update
    ◊ TYPE_MAJ = / 'CAR'           [DEFECT]
                 /'ISOP2'
    # name of the field of sizes which will contain the new field
    ♦ CHAM_MAJ = Co (chpmaj)             [K8]
    #   type of the updated field
    ♦ TYPE_CHAM =/ 'NOEU_TEMP_R'
                 / 'NOEU_DEPL_R'
                 / etc…
    )
```

```
# Addition of fields at exit on the new grid
◊ ADD_CHAM = _F(
   # name of the field of sizes which will contain the new field
   ♦ CHAM_MAJ = Co (chpadd)              [K8]
   #  category of the field to be created
   ♦ CHAM_CAT = / 'LEVEL'
                / 'QUALITY'
                / 'DIAMETRE'
     )

# History of the adaptation
◊ UNITE_HIST_IN = histin                 [I]
◊ UNITE_HIST_OUT = histout               [I]

# If the adaptation is a reading, (READING), choice of the model:
   ♦  MODEL  =  model     [model]
# Finsi

◊ NUMBER =      / 'YES'                  [DEFECT]
                / 'NOT'

◊ QUALITY =     /'NOT'                [DEFECT]
                / 'YES'

◊ DIAMETER =    /'NOT'                [DEFECT]
                / 'YES'

◊ CONNEXITY =   /'NOT'                [DEFECT]
                /'YES'

◊ SIZE =     /'NOT'                [DEFECT]
                /'YES'

◊ PROP_CALCUL =/'NOT'                 [DEFECT]
                / 'YES'

◊ INTERPENETRATION =      / 'YES'

◊ ELEMENTS_ACCEPTES = / 'LOBSTER'      [DEFECT]
                      / 'TO BE UNAWARE OF'

◊ LANGUAGE =       / 'French'    [DEFECT]
                / 'FRENCH'
                / 'ENGLISH'
                / 'ENGLISH'

◊ VERSION_HOMARD = / 'V11_10'          [DEFECT]
                   / 'V11_N'
                   / 'V11_N_PERSONNELLEMENT'

◊ SOFTWARE =    software      [K]
```

```
#   If the version is the version of development, (V11_N, V11_N_PERSO):
    ◊ UNIT =      unit                    [I]
#   Finsi



◊   INFORMATION =    / 1                          [DEFECT]
                     / 2
                     / 3
                     / 4

)
```

# 3 Description of an adaptation of grid

## 3.1 Outline general of an adaptation

The principle general of a calculation with adaptation of grid is the following:

Phase 1: Reading of the initial grid, M0
Definition of materials
Phase 2:
- definition of the model, the loadings on this grid M0
- calculation producing a result RESU0
- possible calculation of a field controlling refinement, CHAMP0

This initial phase is the standard phase of any calculation

Phase 3: Adaptation of the grid M0. One recovers a new grid, M1
Phase 4:
- definition of the model, the loadings on the grid M1,
- calculation producing a result RESU1,
- possible calculation of a field controlling it refinement, CHAMP1.

Phase 4 is similar to phase 2. The only thing which changed is the grid. So all the concepts while depending must be begun again. Today, there is no possibility neither of re-using the old concepts, nor to destroy them automatically.

Then, one can continue, as many times as one wants, the tandem phase 3/phase 4. That is done either by duplicating the instructions, or by writing a loop python.

See the reference [bib1] for a general presentation of the adaptation of grid and LOBSTER, accompanied by examples.

## 3.2 Operation of the macro-order

Phase 3 carries out the adaptation of the grid. It is activated by the macro-order MACR_ADAP_MAIL, described in this document. It has as an essential argument the name of the concept of the current grid and the name which one will give to the concept of the future grid. The other obligatory data is the type of adaptation which one wishes: refinement or the déraffinement free one, i.e. according to the values which a field takes on the meshs of the grid, or of a geometrical zone, or refinement or the déraffinement uniform one, i.e. all the meshs are treated same manner.

The other data depend then on the options selected.

In complement with the adaptation, LOBSTER can provide on request of the assessments on the quality or the diameter of the meshs of the grid, the connexity of the field of calculation, the sizes characteristic, the overstrained elements or a control of the not-interpenetration of the meshs. This information is obtained by the activation of the associated keywords. One will look with profit the order MACR_INFO_MAIL [U7.03.02] which makes it possible to obtain all this information, independently of any calculation.

## 3.3 Some comments

The adapted grid contains the same groups as the grid as starter, with the following rule: a group defines the same locus in the two grids.

- To use a group of nodes amounts defining specific places. The group in the adapted grid will be the list of the same nodes, neither more, nor less, to represent the same points; only their numbers will have possibly changed.
- To use a group of segments amounts defining lines. The group in the adapted grid will be the list of the segments which represent the same lines. According to the mode of adaptation, these segments will be either the same ones, except for the number, or the halves of the initial segments.

• To use a group of triangles and/or quadrangles amounts defining surfaces. The group in the adapted grid will be the list of the triangles and/or quadrangles which represent same surfaces. According to the mode of adaptation, these meshs will be either the same ones, except for the number, or the fractions of the initial meshs 2D.
• In the same way, to use a group of meshs 3D amounts defining volumes. The group in the adapted grid will be the list of the meshs 3D which represent same volumes. According to the mode of adaptation, these meshs will be either the same ones, except for the number, or the fractions of the initial meshs 3D.

The consequence is the following one. The loadings of mechanical or thermal calculation must exclusively be defined by groups of coherent dimension with the phenomenon which one wants to model.
Any other operation will lead to an error in calculation on the adapted grid. To use meshs defined by their number is impossible because classification will change. To use groups of nodes or meshs of the bad dimension will not describe the place completely.
For a more detailed and illustrated explanation, to look at:
http://www.code-aster.org/outils/homard/usage/regles.fr.htm#CL

When one wants to adapt several times of continuation a grid, it is fundamental to respect the chain of the grids strictly. With the first iteration, grid of entry of MACR_ADAP_MAIL is the initial grid of the case which one treats. Then, grid of entry of one MACR_ADAP_MAIL must be the grid of exit of MACR_ADAP_MAIL precedent. Caution: it is not enough to give a grid which is formally the same one, following a copy for example. It is imperative to provide the same concept. If one does not proceed thus, one will lose the history of refinement of the meshs and it will be impossible later on of déraffiner. More serious, LOBSTER not being informed more of additional cuttings which were introduced to ensure conformity, one will be led to cut out meshs by strongly degrading their quality. If this bad setting in data appears, an alarm is emitted.

The adaptation of grid is possible in mode CONTINUATION. The data necessary to the recovery are automatically filed then read again in the repertoire of conservation of the base necessary to *Code_Aster*. To use LOBSTER in continuation is thus done in the same way that to use *Code_Aster* in continuation.

In a general way, the essential impressions provided by LOBSTER are inserted in the file "mess"with the wire of the execution. In the event of error or in mode of information 3 or 4, more detailed impressions take place.

# 4 Operands

## 4.1 Operand ADAPTATION

♦ ADAPTATION = / 'RAFF_DERA'
/ 'REFINEMENT'
/ 'DERAFFINEMENT'
/ 'RAFF_DERA_ZONE'
/ 'RAFFINEMENT_UNIFORME'
/ 'DERAFFINEMENT_UNIFORME'
/ 'NOTHING'
/ 'MODIFICATION'
/ 'READING'

This operand makes it possible to define the type of adaptation wished.

Initially, one finds the modes of adaptations which are controlled by a field. In other words, the decision of (die) refining a mesh is caught according to the value of a computed field before on this mesh. The choice can be done between three alternatives:
- 'RAFF_DERA' : the grid is refined and déraffiné according to the field. It is the option recommended.
- 'REFINEMENT' : only the function of refinement is activated. No mesh will be déraffinée.
- 'DERAFFINEMENT' : it is the reverse; only the function of déraffinement is activated. No mesh will be refined.

In second place, one can decide to adapt the grid in geometrical zones defined by boxes. All the meshs whose at least two nodes are present in one of these boxes will be retained. That makes it possible to make refinements or déraffinements *a priori*, without to have done calculation.
- 'RAFF_DERA_ZONE' : the meshs of each definite box are refined or déraffinées.

Lastly, one can activate a uniform adaptation of a grid. In other words, all the meshs of the grid are treated same manner. The choice can be done between three alternatives:
- 'RAFFINEMENT_UNIFORME' : all the meshs are refined,
- 'DERAFFINEMENT_UNIFORME' : all the meshs are déraffinées,
- 'NOTHING' : all the meshs are preserved; the grid is the same one at the exit as at the entry.

**Note:**
> *When one applies an option of déraffinement, one does nothing but reconsider behind former refinements. It is necessary to understand this option like die-refinement. In particular, one will be able to never obtain a grid coarser than the initial grid.*
> *The options of refinement or déraffinement can apply only to part of the grid. That is obtained by the option of filtering GROUP_MA or GROUP_NO.*

Two complementary options exist :
The first allows the modification of grid, to change the degree of the grid :
- 'MODIFICATION' : the grid is modified overall.

The second makes it possible to read fields at the points of Gauss which were updated on the new grid :
- 'READING' : the fields at the points of Gauss are read.

## 4.2 Operand MAILLAGE_N

♦ MAILLAGE_N = man

Grid of the type [grid] to adapt or modify. Attention, the adaptation can relate only to the following meshs: mesh-points, segments, triangles, quadrangles, tetrahedrons, hexahedrons or pentahedrons. If one provides a comprising grid of other meshs, for example pyramids, two cases are possible: either a

stop in error, or an adaptation on the authorized zone and a restitution with identical moreover grid. The choice between these two operating processes is made by the keyword ELEMENTS_NON_HOMARD.
The grid is in degree 1 or 2, but it is not possible to mix both.
In all the cases, the presence of the enriched meshs HEXA27 is prohibited.

When the choice was made to read fields at the points of Gauss, one gives here the grid on which they are.

## 4.3 Operand MAILLAGE_NP1

◊ MAILLAGE_NP1 = Co (manp1)

The name of the concept of the type [grid] which will contain the grid resulting from the adaptation. This name must respect the usual constraints of the names of concept (8 characters to the maximum) and not to be already used.

## 4.4 Operand MAILLAGE_NP1_ANNEXE

♦ MAILLAGE_NP1_ANNEXE = Co (manp1ann)

This operand makes it possible to produce a grid similar to the grid obtained by the operand MAILLAGE_NP1, but of different degree. It is useful into thermomechanical where thermal calculation takes place on the grid in degree 1 and mechanics on the same grid but in degree 2. This name must respect the usual constraints of the names of concept (8 characters to the maximum) and not to be already used.

## 4.5 Choice of the field of piloting of the adaptation

In the case of a free adaptation, the piloting of the meshs to be refined or déraffiner is carried out by a field. This field is contained either in a structure of result, or in a field of sizes. This field can be a field of indicator of error to the digital direction of the term (QIRE_ELEM for example) but it is not obligatory; any field can be used. One can for example control the adaptation by the field of the constraints or by a field builds purposely like a distance or a criterion from damage. It is enough that this field is defined by its name as described in the documents [U4.81.01], [U4.81.02] or [U4.81.03].

If the field is a field with the nodes, the decision of refinement/déraffinement will be made on each edge according to the values of the field on its nodes. If the field is a constant field by element, it is this value which will control refinement/déraffinement mesh. If the field is a field with the nodes by element or the points of Gauss, the algorithm will be based on the maximum value in the mesh to decide refinement/déraffinement.

### 4.5.1 Operand RESULTAT_N

/ ◊ RESULTAT_N = resun

This operand makes it possible to indicate the concept of the type [result] which contains the field to be used for free adaptation.

#### 4.5.1.1 Operand NOM_CHAM

◊ NOM_CHAM = nomsymb

One specifies here which is the field which is used to control the adaptation.

**Caution:**
*The field must be present in the result; if it is absent, it is not calculated of office.*

### 4.5.2 Operand CHAM_GD

```
/   ◊ CHAM_GD   =  cham_gd_i
```

This operand makes it possible to indicate the concept of the type [cham_gd] which contains the field to be used to control the free adaptation.

### 4.5.3 Operand NOM_CMP

```
◊  NOM_CMP   =  l_cmp
```

Name of the component of the field which must be used to control the adaptation of grid. If several components are wished, to give the list here.
If no component is here defined, the order will take all those which exist in the transmitted field.

The type of taking into account of the components is controlled by USAGE_CMP.

### 4.5.4 Selection of the temporal parameter of the field

If the structure of result contains the field necessary only for one sequence number, nothing is to be specified. In fact the values of the field to this sequence number will be used.
If not, it is necessary to specify about which number it is. That is done by the designation of a sequence number or a value of moment. To refer to the document [U4.71.00] for the details on these keywords.

### 4.5.5 Operand USAGE_CMP

```
◊  USAGE_CMP =  /  'NORME_L2'          [DEFECT]
                /  'NORME_INFINIE'
                /  'ABSOLUTE'
                /  'RELATIVE'
```

One specifies here how to treat the various components of the field controlling the adaptation. One leaves the principle which refinement carries on the great examined values and, symmetrically, déraffinement relates to the small values. By default, one will filter refinement and déraffinement by examining the L2 standard of the components of the field on the meshs (or the nodes), i.e. the square root of the sum of the squares of the values of the components (standard known as Euclidean).
If several components were retained, one can choose between two types of standard: either the L2 standard, by default choice, or the infinite standard, i.e. largest of the absolute values of the components.

If only one component is retained to control the adaptation, choices NORME_L2, NORME_INFINIE and ABSOLUTE are equivalent: one will examine the absolute value of the field. An alternative is possible: to use RELATIVE allows to control the adaptation with the gross amounts of the field. In this case, for a field whose values are negative, bearing refinement on the maximum values, in fact the zones where the value is close to 0 will be refined; symmetrically, déraffinement will relate to the zones where the value is very large negatively.

### 4.5.6 Operand USAGE_CHAMP

```
◊  USAGE_CHAMP =/'MESH'       [DEFECT]
                /'JUMP'
```

By default, the piloting of the adaptation is done by the sorting of the values of the transmitted field, mesh by mesh or node by node.
With the alternative JUMP, LOBSTER one will sort on the jump of the field between meshs, according to the following process. For each mesh, LOBSTER starts by calculating the maximum of absolute

deviation between the value of the field on the current mesh and its value on each close mesh. This maximum is allotted to the current mesh. Then, one sorts the meshs on these maximum departures according to the usual criteria.
In 2D, the examined neighbors are the triangles/quadrangles which divide an edge with the mesh in progress.
In 3D, in fact the voluminal meshs divide a triangular or quadrangular face with the current mesh.
If the field is defined by node, the neighbors are the nodes which divide an edge with the current node.

**Note:**
*This option makes it possible to easily adapt the grid by setting like objective a regular variation of a field of a mesh to the other. Thus, to choose the type JUMP and the field SIEF_ELGA allows to obtain a grid where the strong variations of constraints of a mesh to its neighbor will attenuate.*

### 4.5.7 Operand ADAP_INIT

```
◊  ADAP_INIT =  /  'TO KEEP'        [DEFECT]
                /  'TO REFINE'
                /  'DERAFFINER'
```

When the field controlling the adaptation is defined on all the grid, this option is without effect.
In the contrary case, it makes it possible to specify how the areas are treated where this field is not defined.
With the alternative TO KEEP, the meshs of the areas where the indicator is not defined are *a priori* kept such as they are. It is the option by default.
With the alternative TO REFINE , the meshs of the areas where the indicator is not defined are *a priori* cut out.
With the alternative DERAFFINER , the meshs of the areas where the indicator is not defined are *a priori* reactivated.
Obviously, that is only one initialization and these decisions can evolve to ensure the conformity of the grid, taking into account the decisions brought to the close meshs.

**Note:**
*This option is particularly useful when one uses an indicator built starting from a field. For example, one uses the damage calculated in a given area whereas elsewhere, it is not defined. One will be able to choose not to touch with the other meshs or to try of the déraffiner.*
*For example: in calculations of excavation, meshs are withdrawn from the model to simulate the digging; these meshs do not carry any more a value of indicator. If nothing is done, they remain in the grid. If the alternative is chosen DERAFFINER , they will be déraffinées as the progression of the excavation.*

## 4.6 Operand CRIT_RAFF_xxxx

In the case of free adaptation implying of the refinement of grid, it is necessary to define a high criterion of refinement. All the meshs for which the field is higher than this criterion will be refined. It is important to look at a posteriori the pace of the distribution of the field. That is possible thanks to the impressions carried out by LOBSTER in the file mess. One will find there in particular a table presenting this distribution in the form of histogram; to see the chapter 5 for an example with accompanying notes.

For the choice of the criterion, four alternatives are possible:

### 4.6.1 Operand CRIT_RAFF_PE

```
◊  /CRIT_RAFF_PE =  crp
```

The criterion is defined by a proportion of meshs to refine. It is a real number ranging between 0 and 1. The process is the following:

- calculation amongst meshs N corresponding to the proportion defined by crp that is to say N = crp X full number of meshs
- refinement of N meshs with the strongest value of the field.

### 4.6.2 Operand CRIT_RAFF_ABS

```
/CRIT_RAFF_ABS = CRA
```

The criterion is defined by an absolute value of the field. All the meshs with a value higher than this value will be refined.

### 4.6.3 Operand CRIT_RAFF_REL

```
/CRIT_RAFF_REL = crr
```

The criterion is defined by a relative value of the field. It is a number ranging between 0 and 1. The process is the following:

- calculation of the minimal and maximum values of the indicator,
- calculation of the value corresponding to the necessary proportion: v= vmin + crr (vmax-vmin),
- refinement of all the meshs where the field is higher than this value.

### 4.6.4 Operand CRIT_RAFF_MS

```
◊  /CRIT_RAFF_MS =  crms
```

The criterion is defined by an absolute value of the field, calculated according to the average and of the standard deviation of this field. All the meshs with a value higher than this value will be refined. The criterion is worth: average + n*sigma, where N is the provided, strictly positive coefficient.

## 4.7 Operand CRIT_DERA_xxxx

In the case of free adaptation implying of déraffinement, it is necessary to define a low criterion of déraffinement. All the meshs where the field is lower than this criterion will be déraffinées. Three alternatives are possible.

### 4.7.1 Operand CRIT_DERA_PE

```
◊  /CRIT_DERA_PE =  cdp
```

The criterion is defined by a proportion of meshs in déraffiner. It is a number ranging between 0 and 1. The process is the following:

- calculation amongst meshs N corresponding to the proportion defined by cdp that is to say N = cdp X
- déraffinement N meshs with the low value of field.

### 4.7.2 Operand CRIT_DERA_ABS

```
/CRIT_DERA_ABS = half-value layer
```

The criterion is defined by an absolute value of the field. All the meshs with a value of field lower than this value will be déraffinées.

### 4.7.3 Operand CRIT_DERA_REL

/CRIT_DERA_REL = cdr

The criterion is defined by a relative value of the field. It is a number ranging between 0 and 1. The process is the following:
- calculation of the minimal and maximum values of the indicator,
- calculation of the error value *v* corresponding to the proportion cdr such as: v= vmin + cdr (vmax-vmin),
- déraffinement of all the meshs where the field is lower than this value.

### 4.7.4 Operand CRIT_DERA_MS

◊ /CRIT_DERA_MS = cdms

The criterion is defined by an absolute value of the field, calculated according to the average and of the standard deviation of this field. All the meshs with a value lower than this value will be déraffinées. The criterion is worth: average - n*sigma, where N is the provided, strictly positive coefficient.

## 4.8 Keyword ZONE

♦ ZONE = _F (

In the case of a request for adaptation per zone, it is necessary to define at least a zone. This keyword is to be employed as many times as one wants to define zones of adaptation. The principle is the following: one defines a zone by coordinates then each mesh of which at least one of the edges is in this zone will be selected.
One has the choice between several types of zones.

**Caution:**
*For a calculation which would be 2D, the types of zone are in fact of the rectangles or the circles. But as the concept of grid strictly 2D is unknown in Code_Aster at the time of the creation of the orders, one will suppose that the 3 ème coordinate Z is worthless.*

### 4.8.1 Type of the zone

♦ TYPE= / 'RIGHT-ANGLED'
/ 'LIMPS'
/ 'DISC'
/ 'SPHERE'
/ 'CYLINDER'
/ 'DISQUE_PERCE'
/ 'PIPE'

This operand makes it possible to define the type of zone wished.

### 4.8.2 Use of the zone

♦ USAGE= / 'REFINEMENT' [DEFECT]
'DERAFFINEMENT'

This operand makes it possible to define the use allotted to the zone. With the choice 'REFINEMENT', all the edges whose two ends belong to the zone will be cut, With the choice' DERAFFINEMENT', all the edges whose two ends belong to the zone will be reactivated.

### 4.8.3 Case of the rectangle

#### 4.8.3.1 Operands X_MINI, X_MAXI, Y_MINI, Y_MAXI

♦ X_MINI = x_mini

- X_MAXI = x_maxi
- Y_MINI = y_mini
- Y_MAXI = y_maxi

They are the extreme values of the coordinates of the rectangle including the meshs to be refined.

## 4.8.4 Case of the parallelepipedic box

### 4.8.4.1 Operands X_MINI, X_MAXI, Y_MINI, Y_MAXI, Z_MINI, Z_MAXI

- X_MINI = x_mini
- X_MAXI = x_maxi
- Y_MINI = y_mini
- Y_MAXI = y_maxi
- Z_MINI = z_mini
- Z_MAXI = z_maxi

They are the extreme values of the coordinates of the box including the meshs to be refined.

## 4.8.5 Case of the disc

### 4.8.5.1 Operands X_CENTRE, Y_CENTRE

- X_CENTER = x_centre
- Y_CENTER = y_centre

They are the coordinates of the center of the disc.

### 4.8.5.2 Operand RAY

- RAY = ray

It is the ray of the disc.

## 4.8.6 Case of the sphere

### 4.8.6.1 Operands X_CENTRE, Y_CENTRE, Z_CENTRE

- X_CENTER = x_centre
- Y_CENTER = y_centre
- Z_CENTER = z_centre

They are the coordinates of the center of the sphere.

### 4.8.6.2 Operand RAY

- RAY = ray

It is the ray of the sphere.

## 4.8.7 Case of the cylinder

The cylinder is defined by an axis and a ray. It is limited by two plans perpendicular to the axis. The foreground is positioned by a point on the axis. The second plan is distant first a height, in the direction of the definite axial vector.

### 4.8.7.1 Operands X_AXE, Y_AXE, Z_AXE

- X_AXE = x_axe
- Y_AXE = y_axe
- Z_AXE = z_axe

They are the coordinates of the directing vector of the axis of the cylinder. The orientation does not have importance. The vector is not necessarily normalized.

#### 4.8.7.2 Operands X_BASE, Y_BASE, Z_BASE

- X_BASE = x_base
- Y_BASE = y_base
- Z_BASE = z_base

They are the punctual coordinates at the base of the cylinder and located on the axis.

#### 4.8.7.3 Operand RAY

- RAYON = ray

It is the ray of the cylinder.

#### 4.8.7.4 Operand HEIGHT

- HEIGHT = height

It is the height of the cylinder.

### 4.8.8 Case of a bored disc

#### 4.8.8.1 Operands X_CENTRE, Y_CENTRE

- X_CENTER = x_centre
- Y_CENTER = y_centre

They are the coordinates of the center of the disc.

#### 4.8.8.2 Operands RAYON_INT, RAYON_EXT

- RAYON_INT = rayon_int
- RAYON_EXT = rayon_ext

They are the rays interior and outside of the bored disc.

### 4.8.9 Case of the pipe

The pipe is defined by an axis and its rays interior and outside. It is limited by two plans perpendicular to the axis. The foreground is positioned by a point on the axis. The second plan is distant first a height, in the direction of the definite axial vector.

#### 4.8.9.1 Operands X_AXE, Y_AXE, Z_AXE

- X_AXE = x_axe
- Y_AXE = y_axe
- Z_AXE = z_axe

They are the coordinates of the directing vector of the axis of the pipe. The orientation does not have importance. The vector is not necessarily normalized.

#### 4.8.9.2 Operands X_BASE, Y_BASE, Z_BASE

- X_BASE = x_base
- Y_BASE = y_base
- Z_BASE = z_base

They are the punctual coordinates at the base of the pipe and located on the axis.

#### 4.8.9.3 Operands RAYON_INT, RAYON_EXT

- RAYON_INT = rayon_int
- RAYON_EXT = rayon_ext

They are the rays interior and outside of the pipe.

#### 4.8.9.4 Operand HEIGHT

- HEIGHT = height

It is the height of the pipe.

## 4.9 Operands GROUP_MA / GROUPE_NO

◊ GROUP_MA = l_grma
◊ GROUP_NO = l_grno

If this option is absent, the piloting of the adaptation applies to all the grid. If one wishes to restrict this piloting with part of the grid, one gives here the list of the groups which define this part.
Example 1, to refine an area of the grid uniformly: for uniform refinement is asked and one gives the list of the groups of meshs forming this area.

Example 2, to apply the field of piloting of the adaptation only to certain areas: one asks for refinement/déraffinement with the field and one provides the list of the groups of meshs forming this area.

**Note:**

*For all the meshs 1D, 2D or 3D contained in the groups of the list, there is refinement according to the criteria selected. For the meshs 0D or the nodes contained in the groups, one retains the edges whose two ends are in these lists.*
*The meshs selected are adapted, but the adaptation will go certainly one further being able to provide a grid in conformity in exit.*

## 4.10 Operand DIAM_MIN

◊ DIAM_MIN = diamin

It is pointed out that the diameter of a mesh is the length of the greatest segment than it is possible to trace inside. For a triangle or a tetrahedron, the diameter is the length on the largest side. For a quadrangle, a hexahedron or a pentahedron, the diameter is the length of the largest diagonal.
To give a value to diamin allows not to make a grid extremely fine. A mesh which would be selected like front being refined because of field of piloting or of the geometrical zone but whose diameter is already lower than this minimal value diamin will not be cut out; it will be kept such as it is. Caution: it is nevertheless possible that with final it is nevertheless cut out if its neighbors are it, to respect the conformity of the final grid.
By default, no limit is given: one can obtain meshs as small as one wants.

## 4.11 Operand NIVE_MAX

◊ NIVE_MAX = nivmax

It is the maximum level of refinement of the grid. In other words a mesh of the initial grid could not be divided more nivmax time in the whole of the process. That makes it possible to ensure that the grid will not become extremely fine in the vicinity of a singularity: the minimal size of an edge will be its initial size divided by $2^{nivmax}$.
By default, no limit is given: one can cut out as much as one wants.

## 4.12 Operand NIVE_MIN

◊ NIVE_MIN = nivmin

It is the minimal level of déraffinement grid. I.e. only meshs resulting from at least nivmin cuttings of grid can be déraffinées. That makes it possible to ensure that one from there will not go up too high in déraffinement: one thus keeps a minimal smoothness with the grid.
By default, no limit is given: one can déraffiner until finding the initial grid.

## 4.13 Keyword MAILLAGE_FRONTIERE

◊ MAILLAGE_FRONTIERE = maf

The choice of this option makes it possible the process of adjustment to follow the curve of the edges of the grid. The option applies exclusively on board 1D. For edges 2D, it is necessary to use the option FRONTIERE_ANALYTIQUE. A concept here is provided *Code_Aster* of type grid who contains a fine grid of the unidimensional edges of the geometry. This grid is not thus formed *a priori* that segments. Their lengths are much lower than those of the segments of edge of the grid to adapt. If the process of adjustment is brought to cut a segment of edge, the new node will be placed on the grid of the border. Thus the angles will be softened as adaptations.
The location of the various edges is done by the groups with the following rules:

- the edges are described by groups of segments;
- an edge is described by the same name of group in the grid of calculation and the grid of the borders;
- an edge can have only zero or two ends;
- it neither essential nor is disadvised including the rectilinear edges;
- the edge can as well be external, more running, which interns, to separate two materials.
- the edge is not necessarily plan; it can be a space curve 3D like the intersection of two cylinders for example.

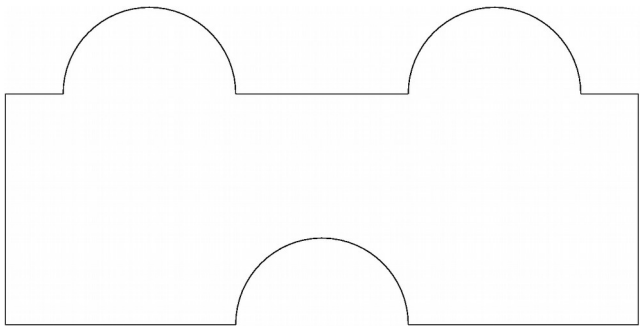
Known as differently, a group of segments of edge must comprise a list of segments forming a line.

**Note:**

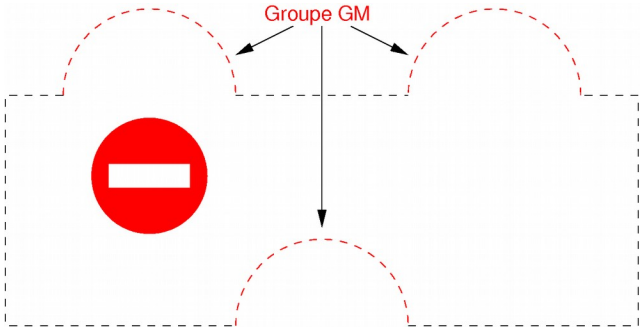
> *The cas-tests will be looked at zzzz121d, zzzz175a, zzzz175c and zzzz259a for examples of piloting of the follow-up of border as well as the Web site of LOBSTER for a graphic illustration of the got result.*

**Example:**

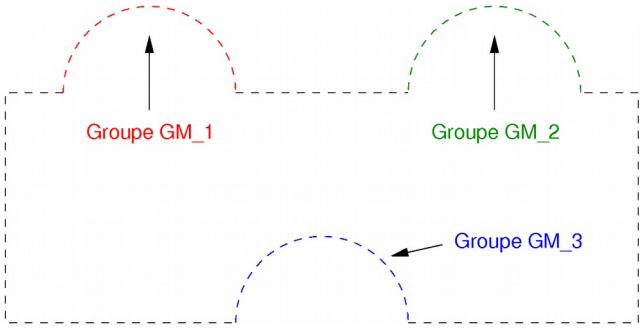
Let us consider a two-dimensional object whose border is not always rectilinear. This border will have been with a grid by elements SEG2 or SEG3 as well in the grid of calculation as in the additional grid. These meshs of edge are put in the same groups.



The bad solution is this one: to locate the meshs of the curved edges and to store them all in the same group. LOBSTER cannot manage a split edge; there will be stop with a message meaning that the line is in several pieces.

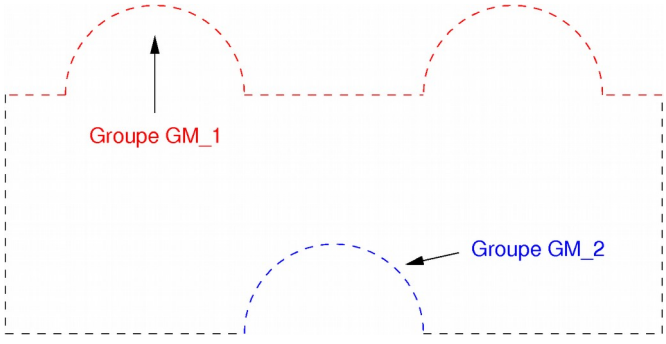



A first good way of making consists in creating groups as many as zones of interest.




Another acceptable solution consists in gathering section by section.

Between the two methods, not of difference for LOBSTER: essence is not to make the full rotation (if not, not of end) and not to cross (if not, too many ends!). One will choose the method easiest to realize in the maillor.




### 4.13.1 Operand GROUP_MA_FRONT

◊ GROUP_MA_FRONT = l_grma

If this option is absent, the follow-up of the border is done for all the groups defined in the grid of the border. If one wishes to restrict this follow-up with part of the border, one gives here the list of the groups of segments which define this part of border.

## 4.14 Keyword FRONTIERE_ANALYTIQUE

◊ FRONTIERE_ANALYTIQUE = _F (

This keyword is to be informed as many times as one wants to define analytical borders. The choice of this option makes it possible the process of adjustment to follow the curve of the edges of the grid. The option applies exclusively on board 2D. For edges 1D, it is necessary to use the option MAILLAGE_FRONTIERE. One provides here the analytical description of each border to be followed. If the process of adjustment is brought to cut a mesh of edge, the new node will be positioned on the border, via its description. Thus the angles will be softened as adaptations.

**Note:**

*The cas-test will be looked atS zzzz175a, zzzz175c and zzzz259a for an example of analytical follow-up of border.*
*When several times of continuation the adaptation is launched, it is essential that each analytical border is defined same manner with each invocation of MACR_ADAP_MAIL : name, type, list of groups, characteristics geometrical.*

### 4.14.1 Name of the border

♦ NAME = name [K]

This operand makes it possible to define the name associated with the border. The choice of this name is free.

### 4.14.2 Type of the border

♦ TYPE = / 'SPHERE'
/ 'CYLINDER'
/ 'CONE_A'
/ 'CONE_R'
/ 'TORUS'

This operand makes it possible to define the type of border wished: sphere, cylinder.

### 4.14.3 Operand GROUP_MA

♦ GROUP_MA = l_grma

One gives here the list of the groups of meshs which define the part of border represented by this analytical definition.

### 4.14.4 Case of the sphere

#### 4.14.4.1 Operands X_CENTRE, Y_CENTRE, Z_CENTRE

♦ X_CENTER = x_centre
♦ Y_CENTER = y_centre
♦ Z_CENTER = z_centre

They are the coordinates of the center of the sphere.

#### 4.14.4.2 Operand RAY

♦ RAY = ray

It is the ray of the sphere.

## 4.14.5 Case of the cylinder

The cylinder is defined by an axis, a point on the axis and a ray.

### 4.14.5.1 Operands X_AXE, Y_AXE, Z_AXE

♦ X_AXE = x_axe
♦ Y_AXE = y_axe
♦ Z_AXE = z_axe

They are the coordinates of the directing vector of the axis of the cylinder. The orientation does not have importance. The vector is not necessarily normalized.

### 4.14.5.2 Operands X_CENTRE, Y_CENTRE, Z_CENTRE

♦ X_CENTER = x_centre
♦ Y_CENTER = y_centre
♦ Z_CENTER = z_centre

They are the punctual coordinates located on the axis of the cylinder.

### 4.14.5.3 Operand RAY

♦ RAY = ray

It is the ray of the cylinder.

## 4.14.6 Case of the cone defined by an angle

The cone is defined by an axis, a center on the axis and an angle.

### 4.14.6.1 Operands X_AXE, Y_AXE, Z_AXE

♦ X_AXE = x_axe
♦ Y_AXE = y_axe
♦ Z_AXE = z_axe

They are the coordinates of the directing vector of the axis of the cone. The orientation does not have importance. The vector is not necessarily normalized.

### 4.14.6.2 Operands X_CENTRE, Y_CENTRE, Z_CENTRE

♦ X_CENTER = x_centre
♦ Y_CENTER = y_centre
♦ Z_CENTER = z_centre

They are the coordinates of the center on the axis of the cone.

### 4.14.6.3 Operand ANGLE

♦ ANGLE = angle

It is the angle in degree of the cone.

## 4.14.7 Case of the cone defined by rays

The cone is defined by two points on its axis and the two rays corresponding to these positions.

### 4.14.7.1 Operands X_CENTRE, Y_CENTRE, Z_CENTRE

- X_CENTER = x_centre
- Y_CENTER = y_centre
- Z_CENTER = z_centre

They are the coordinates of the first point located on the axis of the cylinder.

### 4.14.7.2 Operand RAY

- RAY = ray

It is the ray of the cone for this first position on the axis.

### 4.14.7.3 Operands X_CENTRE2, Y_CENTRE2, Z_CENTRE2

- X_CENTRE2 = x_centre2
- Y_CENTRE2 = y_centre2
- Z_CENTRE2 = z_centre2

They are the coordinates of the second point located on the axis of the cylinder.

### 4.14.7.4 Operand RAYON2

- RAYON2 = rayon2

It is the ray of the cone for this second position on the axis.

## 4.14.8 Case of the torus

The torus is defined by an axis, a centre and two.

### 4.14.8.1 Operands X_AXE, Y_AXE, Z_AXE

- X_AXE = x_axe
- Y_AXE = y_axe
- Z_AXE = z_axe

They are the coordinates of the directing vector of the axis of the torus. The orientation does not have importance. The vector is not necessarily normalized.

### 4.14.8.2 Operands X_CENTRE, Y_CENTRE, Z_CENTRE

- X_CENTER = x_centre
- Y_CENTER = y_centre
- Z_CENTER = z_centre

They are the coordinates of the center of the torus.

### 4.14.8.3 Operand RAY

♦ RAY = ray

It is the ray of revolution of the torus.

#### 4.14.8.4 Operand RAYON2

♦ RAYON2 = rayon2

It is the ray of the primary education circle which turns around the axis of the torus.

## 4.15 Keyword MAJ_CHAM

◊ MAJ_CHAM = _F (

This keyword is to be employed as many times as one has fields to update old grid towards the adapted grid. This field is contained either in a structure of result, or in a field of sizes.

**Note:**
*It should be taken care that the field is defined on a grid identical to the grid sunken under the operand MAILLAGE_N.*

### 4.15.1 Operand RESULT

/ ◊ RESULT = resu

Name of the concept [result] containing the field to be updated.

#### 4.15.1.1 Operand NOM_CHAM

◊ NOM_CHAM = nomsymb [K16]

Reference symbol of the field which one wishes to express on the new grid.

### 4.15.2 Operand CHAM_GD

/ ◊ CHAM_GD = cham_gd

Name of the concept [cham_gd] containing the field to be updated.

### 4.15.3 Operand NOM_CMP

◊ NOM_CMP = l_cmp

Name of the component of the field which must be updated. If several components are wished, to give the list here.

If no component is here defined, the order will take all those which exist in the transmitted field.

### 4.15.4 Selection of the temporal parameter of the field to be updated

The selection of the sequence number associated with the field to be interpolated is done by the designation of a sequence number or a value of moment. To refer to the document [U4.71.00] for the details on these keywords.

### 4.15.5 Operand TYPE_MAJ

◊ TYPE_MAJ = / 'CAR' [DEFECT]
/ 'ISOP2'

One specifies here the type of update wished.
By default, operation 'CAR', is… automatic: the interpolation is made according to the nature of the field.
For a constant field by element, if a mesh is cut out, the value of the field is deferred such as it is on the meshs girls.
For a field with Nœuds, the interpolation takes place with the functions of form P1 or P2 according to the support of the field. An alternative is possible for the fields with Nœuds expressed on a grid of degree 2. While specifying 'ISOP2', the interpolation is made by functions of P1 form expressed on the under-meshs of the element. This technique guarantees that the interpolated field respects the extreme values of the initial field on a mesh.
For a field at the points of Gauss, the update takes place only whenever the types of meshs are homogeneous between the entry and the exit: triangle or tetrahedron. For the quadrangles, the hexahedrons or the pentahedrons, kidney is not made because one cannot define the families of points of Gauss on the produced triangles or tetrahedrons. For a grid in triangles, one proceeds as follows:

- for an unchanged triangle after the adaptation, the values at the points of Gauss are renewed with the identical one.
- for a cut out triangle, into 2 or 4, one calculates the median value of the field on all the points of Gauss and this value is allotted to all the points of Gauss of all the triangles son.
- for a triangle which disappears (in the event of déraffinement), one calculates the average of the field on all the points of Gauss this triangle and of his/her brothers and this value is allotted to all the points of Gauss of the triangle father which reappears.

This mechanism makes it possible to guarantee that the extreme terminals are not crossed. Same the technqiue is applied to the tetrahedrons.

**Caution:**

*There cannot be of consistency check between the type requested and the true type of the field to be interpolated.*

### 4.15.6 Operand CHAM_MAJ

♦ CHAM_MAJ = Co (chpmaj) [K8]

Name of the concept which will contain the field expressed on the new grid. This concept should not exist. It will be automatically created.
A field with the nodes or the elements will be read automatically by the macro-order which requires the adaptation or the modification of the grid. It will be available in the game of calculation as of the end of the execution of the macro-order. On the other hand, that is not possible for a field expressed at the points of Gauss because Code_Aster needs to know the model for reading. It is necessary for that to proceed in 3 times. The field is calculated by the macro-order which required the adaptation or the modification of the grid, as for a field with the nodes. Then, a new model must be applied to the new grid by the order AFFE_MODELE. Lastly, the reading will be made by a new invocation of MACR_ADAP_MAIL with the help of the parameter ADAPTATION = 'READING' and supply of the grid and the model.

**Note:**

*The cas-test will be looked at zzzz175b for an example of update and reading of fields of various types.*

### 4.15.7 Operand TYPE_CHAM

♦ TYPE_CHAM = /'NOEU_DEPL_R'
/'NOEU_TEMP_R'
/'ELGA_SIEF_R'
/etc…

One indicates the type of the concept here to be updated on the new grid. The name of this type is built with the usual logic of *Code_Aster*. The first 4 characters are 'NOEU', 'ELEM' or 'ELGA'. One

finds then '_'. The following sequence defines the type of field: 'TEMP', 'DEPL', etc the name ends in '_R' for a real field.

Example: 'NOEU_TEMP_R', 'NOEU_DEPL_R', etc.

**Caution:**

*There cannot be of consistency check between the type requested and the true type of the field to be interpolated.*

## 4.16 Keyword ADD_CHAM

◊ ADD_CHAM = _F (

This keyword is to be employed to produce particular fields on the adapted grid.

### 4.16.1 Operand CHAM_GD

♦ CHAM_GD = Co (chpmaj) [K8]

Name of the concept which will contain the field expressed on the new grid. This concept should not exist. It will be automatically created.

**Note:**

*The cas-tests will be looked at zzzz121b and zzzz121f for an example of additions of fields of various types.*

### 4.16.2 Operand CHAM_CAT

♦ CHAM_CAT = /'LEVEL'
/'QUALITY'
/'DIAMETER'

One indicates here the category of the field which one wants to produce.
With 'LEVEL', the produced field will contain for each mesh its level. It is pointed out that a mesh of the initial level is level 0. Then a mesh resulting from *N* standard cuttings of an initial mesh will carry the level *N*. A mesh which ensures the transition from conformity between the level *N* and the level *n+1* the level will carry *n+0,5*.
With 'QUALITY', the produced field will contain the value of quality for each mesh.
With 'DIAMETER', the produced field will contain the value of the diameter for each mesh.

## 4.17 Operand MODEL

◊ MODEL = model [model]

**Caution:**

*This operand is active only when one chose 'READING'like type of adaptation.*

This operand makes it possible to specify the model which was affected with the grid on which the update of fields expressed at the points of Gauss took place.

**Note:**

*The cas-test will be looked at zzzz175b for an example of update and reading of fields at the points of Gauss.*

## 4.18 Operand DEGREE

◊ DEGREE = /'YES'

**Caution:**
*This operand is active only when one chose 'MODIFICATION'like type of adaptation.*

If the choice is 'YES', the degree of the grid is changed overall.

**Note:**
*Compound with MAJ_CHAM , the operand DEGREE can be used for example for the postprocessing of the pressure in THM (cf Doc. [U2.04.05]). For certain studies 3D with bulky grids, it can appear sometimes more performing that PROJ_CHAMP .*

## 4.19 Operand NUMBER

**Note:**
*One will consult the document [U7.03.02] describing the order MACR_INFO_MAIL for comments on the restitutions of the operands QUALITY , INTERPENETRATION , NUMBER , CONNEXITY and SIZE .*

◊ NUMBER = /'YES' [DEFECT]
/ 'NOT'

If the choice is 'YES', an assessment of the numbers of nodes and meshs are printed on the file of messages.

## 4.20 Operand QUALITY

◊ QUALITY = / 'NOT' [DEFECT]
/ 'YES'

If the choice is 'YES', an assessment of the quality of the meshs is printed on the file of message. The quality of a triangle is defined as being the relationship between the length on the largest side and the radius of the inscribed circle. The quality of a quadrangle is defined like the quotient of the product biggest length and averages on the sides and the diagonals by smallest of surfaces of the internal triangles to the quadrangles. In the same way, the quality of a tetrahedron is defined as being the relationship between the length on the largest side and the ray of the registered sphere. These reports are standardized to be worth 1 in the case of an equilateral triangle, of a square, a tetrahedron or an equilateral hexahedron. For any nonequilateral element, quality is higher than 1. See the reference [bib1] for detailed explanations.
The result is presented in the form of tables, with the extreme values.

The interpretation of the produced values depends on the digital method employed for calculation. According to whether the problem is isotropic or not, according to the speed of space variation of the data, according to the technique of calculation, the same mesh can lead to a good jacobien or not. Essence initially is to locate the frankly bad meshs. If it is observed that the maximum of quality exceeds 100, even 1,000 or 100,000, one must worry: one or more meshs are very deformed and the grid is certainly to begin again. In the second time, this information of quality must make it possible to compare two grids a priori correct, without much value. IF the problem is isotropic, one will may find it beneficial to use the grid with the distribution of quality nearest to 1.
One will find illustrations of values of quality of various meshs in [Réf.5].

## 4.21 Operand DIAMETER

◊ DIAMTRE = / 'NOT' [DEFECT]
/ 'YES'

If the choice is 'YES', an assessment of the diameters of the meshs is printed on the file of message. The diameter of a mesh is defined as the length of the greatest segment than it is possible to insert in the mesh.
For a triangle or a tetrahedron, the diameter corresponds to the length on the largest side.

For a quadrangle, a hexahedron, a pentahedron or a pyramid, the diameter are the maximum between the length on the largest side and the length of the largest diagonal.

The result is presented in the form of tables, with the extreme values.

## 4.22 Operand INTERPENETRATION

```
◊  INTERPENETRATION =  /  'NOT'                    [DEFECT]
                       /  'YES'
```

If the choice is 'YES', it is checked that the grid is correct from the point of view of covering: no mesh enters another.

## 4.23 Operand SIZE

```
◊  SIZE  =  /  'NOT'                    [DEFECT]
            /  'YES'
```

If the choice is 'YES', an assessment of the sizes of the under-fields is printed on the file of messages. A under-field is defined like a set of the same meshs dimension and pertaining to the same groups.

## 4.24 Operand CONNEXITY

```
◊  CONNEXITY =  /  'NOT'                    [DEFECT]
                /  'YES'
```

If the choice is 'YES', an assessment of the connexities is printed on the file of messages. It will be known then if the segments, the elements 2D (triangles and quadrangles joined together) or the meshs 3D (tetrahedrons, hexahedrons, pentaèDRES and pyramidES joined together) of only one holding or are divided into several blocks. One will also know the number of holes of the structure: crossing holes or internal holes.

## 4.25 Operand PROP_CALCUL

```
◊  PROP_CALCUL  =  /  'NOT'                    [DEFECT]
                   /  'YES'
```

If the choice is 'YES', a diagnosis on the properties of the meshs as elements for calculation is printed on the file of messages. One counts the number of overstrained elements: the elements of which all the tops are located on the edge. One counts the voluminal meshs (resp. surface) which touches the edge of the field but which is not bordered by surface meshs (resp. linear).

## 4.26 Histories

When one wants to divide a calculation into two independent command sets and that there was adaptation of grid in the first, it is necessary to be very careful for the management of the grid in the second calculation.
If L*E* the second calculation is the prolongation of the first by means of one CONTINUATION, the histories are stored in the base of *Code_Aster* and all occurs well.
On the other hand, if the second calculation is new and that it starts with BEGINNING , the simple reading of the grid will involve errors for a new adaptation. Indeed, the history of cuttings being

absent, this grid will be regarded as new. One will lose any possibility of déraffinement then and qualities of the meshs will be degraded. To cure that, one will file the history during adaptations of the first calculation and one will give this history as starter of the second. That is done by the logical units related to the files.

**Note:**
*One will look at the CAS-test zzzz175c for a practical application of this technique.*

### 4.26.1 Operand UNITE_HIST_OUT

```
◊  UNITE_HIST_OUT   =  unit                [I]
```

Candyou option defines a logical number of unit to file the history of the adaptations of the grid.

### 4.26.2 Operand UNITE_HIST_IN

```
◊  UNITE_HIST_IN    =  unit                [I]
```

Candyou option defines a logical number of unit to read again the history of the adaptations of the grid.

## 4.27 Operand LANGUAGE

```
◊  LANGUAGE  =  /  'French'          [DEFECT]
             /  'FRENCH'
             /  'ENGLISH'
             /  'ENGLISH'
```

This operand specifies the language in which the messages resulting from LOBSTER are printed.

## 4.28 Operand VERSION_HOMARD

```
◊  VERSION_HOMARD  =  /  'V11_10'  [DEFECT]
                      /  'V11_N'
                      /  'V11_N_PERSONNELLEMENT'
```

This operand makes it possible to select the LOBSTER version which is used for the adaptation. By default, LOBSTER 11.10 is launched. It is the version of reference. The choice 'V11_N' activate the version 11.n LOBSTER which is the version of development. The choice 'V11_N_PERSONNELLEMENT' activate a version of development specific to the user. This option makes it possible to the development team of LOBSTER to develop new features. She also makes it possible to make profit the user from an innovation in LOBSTER before the commissioning in *Code_Aster*.

## 4.29 Operand SOFTWARE

```
◊  SOFTWARE =   software           [K]
```

Candyou option proposes to use another interface of coupling enters *Code_Aster* and LOBSTER that provided by default in the repertoire of the tools associated with *Code_Aster*. This option is in fact

reserved to the development team of LOBSTER to develop new features. She makes it possible to test innovations before to have modified the macro-order of piloting.

## 4.30 Operand UNIT

```
◊  UNIT   =  unit              [I]
```

Candyou option is not possible that if one has activated the version of development from LOBSTER, 11.n. The data file transmitted by the user under this number of logical unit will be directly transmitted like complement to the file of LOBSTER configuration. This option in fact is intended to the development team of LOBSTER to develop new features. She makes it possible to test innovations before to have modified the macro-order of piloting.

## 4.31 Operand ELEMENTS_ACCEPTES

```
◊  ELEMENTS_ACCEPTES   =  /   'LOBSTER'     [DEFECT]
                          /   'IGNORE_PYRA'
```

In its current version, LOBSTER can read all the types of meshs but makes carry the adaptation only on some: mesh-points, segments, triangles, quadrangles, tetrahedrons, hexahedrons or pentahedrons. The grid is in degree 1 or 2, but it is not possible to mix both.
By retaining the option 'LOBSTER', the transmission D' a grid containing another thing which these types of meshs will involve a stop in error. It is the option by default.
By choosing the option 'IGNORE_PYRA', one will be able to transmit a comprising grid of the pyramids. L' adaptation will relate only to the zones authorized by LOBSTER. So in consequence of propagation of refinement, a closed area comes to be touched, there will be a stop in error. If not, when refinement is limited to the authorized zone, the other meshs are restored without change.

In all the cases, the presence of the enriched meshs HEXA27 is prohibited.

## 4.32 Operand INFORMATION

```
◊  INFORMATION  =  /  1
                   /  2
                   /  3
                   /  4
```

If INFORMATION 1 is worth, the impressions are minimal; one obtains only those which were explicitly required, the quality of the meshs for example, and the possible error messages.
If INFORMATION is worth 2, one will obtain the messages transmitted by the orders subjacent with the macro-order: IMPR_RESU, LIRE_MAILLAGE, LIRE_RESU.
If INFORMATION is worth 3, one will have the standard LOBSTER messages, recapitulating the execution.
If INFORMATION is worth 4, one will have all the messages transmitted by LOBSTER, for déboggage.

# 5 Example

One will look with profit the command files associated with the CAS-tests zzzz121a, B, C, D, E, F and zzzz175a, B. They express the processes of adjustment of grid either per sequence of the orders or in the form of a loop in language Python.

Here an example of parameter setting of the macro-order.

```
MACR_ADAP_MAIL   (
          ADAPTATION = 'RAFF_DERA',
          MAILLAGE_N = mun,
          MAILLAGE_NP1 = CO ("mdeux"),
          RESULTAT_N = remeun,
          NOM_CHAM = 'QIRE_ELEM',
          NOM_CMP = 'ERREST'
          NUME_ORDRE = 3,
          CRIT_RAFF_PE = 0.01,
          CRIT_DERA_PE = 0.25,
          NIVE_MAX = 5
          MAJ_CHAM  = _F   (
                RESULT = rethun,
                NOM_CHAM = 'TEMP',
                TYPE_CHAM = 'NOEU_TEMP_R',
                INST = 12.5,
                CHAM_MAJ = CO ("tempdeux")
                ),
          QUALITY = 'YES',
       )
```

This sequence will adapt the grid contained in the concept mun and to restore a concept grid of name mdeux. The adaptation is done by refinement and déraffinement free, according to the field contained in the field QIRE_ELEM result remeun, to the 3ème moment; the component used is ERREST. The meshs will be classified according to their level of error decreasing. First % will be refined; the 25% last will be candidates with déraffinement. No mesh of the final grid will have to result of more than 5 refinements.

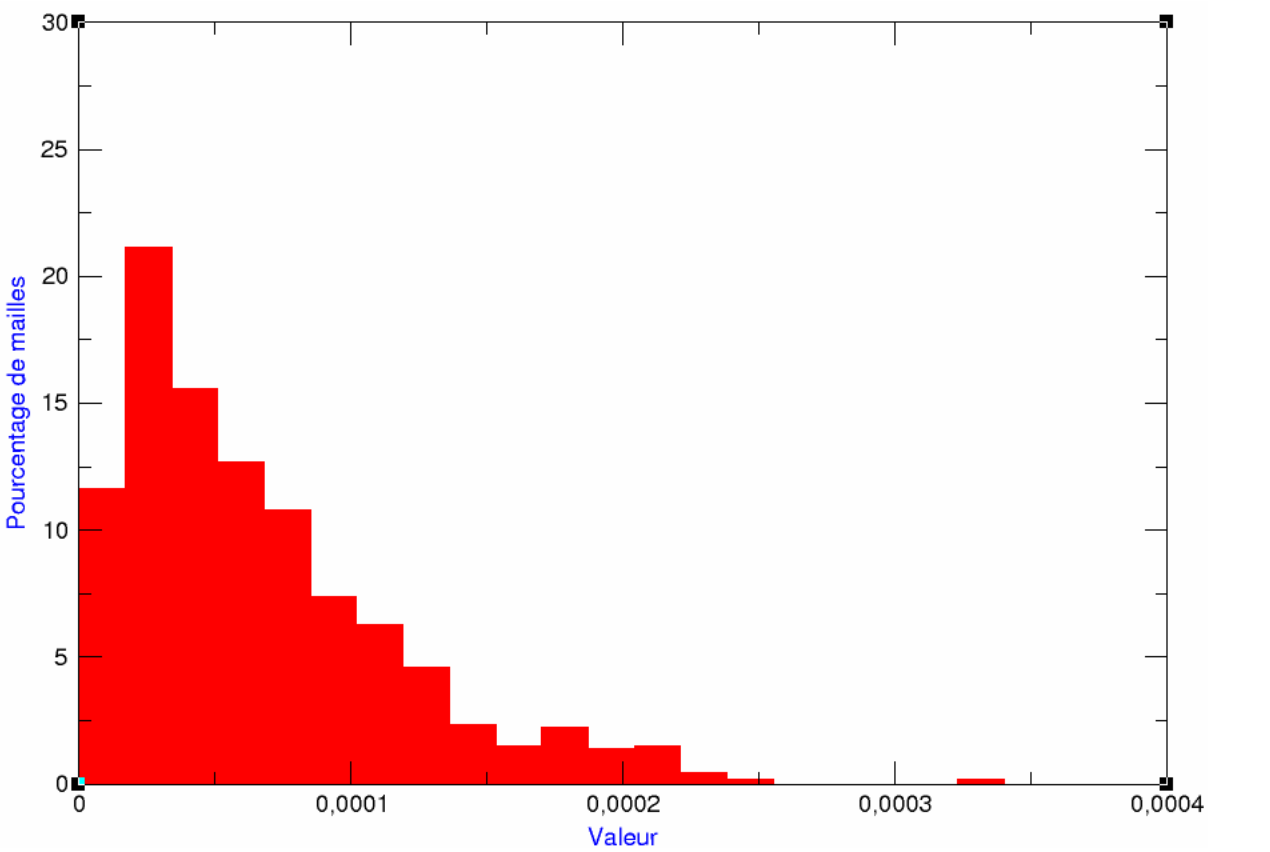
The field TEMP result rethun at the moment 12.5 is expressed on the grid mun. It will be expressed on the grid mdeux in the shape of the field of temperature to the nodes tempdeux.

A summary of the quality of the meshs of the new grid is produced. The interpenetration of the meshs is not controlled.

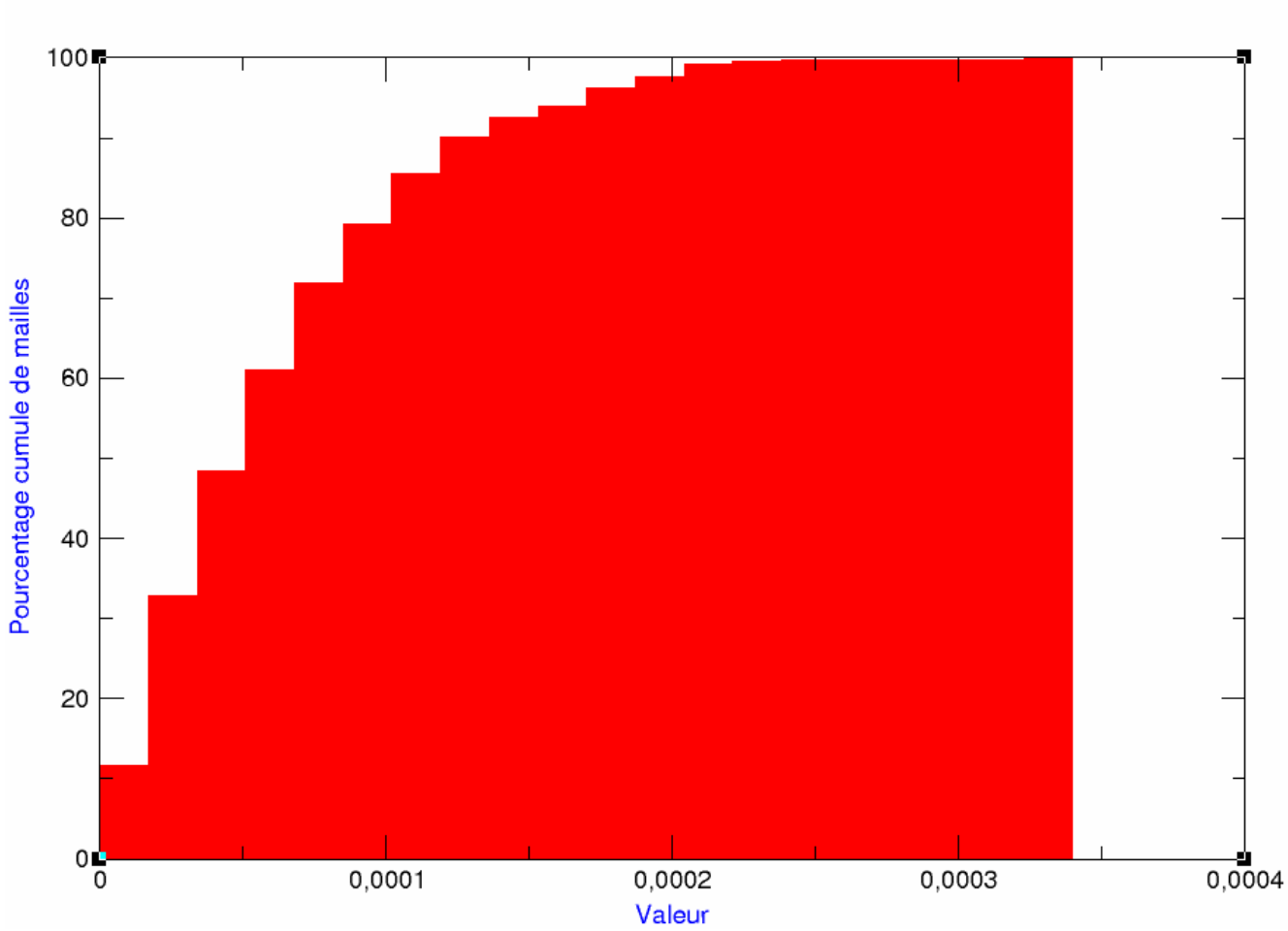
Here an example of the table presenting the distribution of the field controlling the adaptation of the grid.

```
      ************************************************************
      *               Field controlling the adaptation               *
      *            Value on         936 triangles          *
      ************************************************************
      *   Minimum:  0.35358E-05        Maximum:  0.33395E-03   *
      *   Average:  0.66371E-04     Standard deviation:  0.51323E-04   *
      ************************************************************
      *                  Function of distribution                 *
      *      Values        *         Many meshs           *
       *     Minis <  < Maximum  *     by class     *       office plurality
*
      *     * 10 ** - 4       *  in %  . number  *  in %  . number  *
      ************************************************************
      *    0,000 <   0,170 *  11.65.     109 *  11.65.     109 *
      *    0,170 <   0,340 *  21.15.     198 *  32.80.     307 *
      *    0,340 <   0,510 *  15.60.     146 *  48.40.     453 *
      *    0,510 <   0,680 *  12.71.     119 *  61.11.     572 *
      *    0,680 <   0,850 *  10.79.     101 *  71.90.     673 *
      *    0,850 <   1,020 *   7.37.      69 *  79.27.     742 *
      *    1,020 <   1,190 *   6.30.      59 *  85.58.     801 *
      *    1,190 <   1,360 *   4.59.      43 *  90.17.     844 *
      *    1,360 <   1,530 *   2.35.      22 *  92.52.     866 *
      *    1,530 <   1,700 *   1.50.      14 *  94.02.     880 *
      *    1,700 <   1,870 *   2.24.      21 *  96.26.     901 *
      *    1,870 <   2,040 *   1.39.      13 *  97.65.     914 *
      *    2,040 <   2,210 *   1.50.      14 *  99.15.     928 *
      *    2,210 <   2,380 *   0.43.       4 *  99.57.     932 *
      *    2,380 <   2,550 *   0.21.       2 *  99.79.     934 *
      *    2,550 <   2,720 *   0.00.       0 *  99.79.     934 *
      *    2,720 <   2,890 *   0.00.       0 *  99.79.     934 *
      *    2,890 <   3,060 *   0.00.       0 *  99.79.     934 *
      *    3,060 <   3,230 *   0.00.       0 *  99.79.     934 *
      *    3,230 <   3,400 *   0.21.       2 * 100.00.     936 *
      ************************************************************
```

The diagnosis on the distribution of the field controlling the adaptation of grid points out initially the extreme values recorded in calculation in progress. Here the minimum is of $0,353585x10^{-5}$ and the maximum of $0,33395x10^{-3}$. One specifies the median value, $0,66371x10^{-4}$, and the standard deviation, $0,51323x10^{-4}$. Then one presents the distribution by equidistant slice starting from the optimum value, 0. It is seen that for 880 triangles, the value of the field is lower than $1,70x10^{-4}$, that is to say 94,02% of the full number of triangles. Then, for 21 triangles the value of the field lies between $1,70x10^{-4}$ and $1,87x10^{-4}$, that is to say 2,24% of the full number of triangles. In cumulated, one thus notes that for 901= 880 + 21 triangles, the value of the field is lower than $1,87x10^{-4}$, that is to say 96,26% of the total. And so on. For example, for 99,79% of the meshs, the value of the field is lower than $2,55x10^{-4}$.




On the preceding figure, one can see the representation in the form of histogram of the percentages of meshs in each beach of value concerned. As one could also note it in the preceding table, one notes that very few meshs concentrate the strong values.
By visualizing a representation of the cumulated percentage of meshs in a beach of value given, there is the following figure.




From this distribution of the values, one can deduce two consequences on the strategies from refinement.
If one asks for a refinement on a relative criterion of the value of the field, keyword CRIT_RAFF_REL, that amounts selecting the meshs the elements which are on the right of the vertical line passing by this criterion. For example if one asks CRIT_RAFF_REL = 0.77, one will select all the meshs whose error is higher than $0.35358 \times 10^{-5} + 0{,}77 \times (0.33395 \times 10^{-3} - 0.35358 \times 10^{-5})$, that is to say $2{,}58 \times 10^{-3}$ . It is noted that corresponds to very few meshs: 2 only exceed this value, that is to say 0.21% of the total. One had the feeling to ask for an important refinement, 0.77 is a quarter *roughly speaking*, but one makes some does not refine almost nothing.
If one asks for a refinement on a percentage of meshs, keyword CRIT_RAFF_PE, that amounts selecting the meshs which are above the horizontal line passing by this criterion. For example if one asks CRIT_RAFF_PE = 0.10, one will select the 10% of the worst meshs, that is to say 93 meshs. It is the horizontal line with 90%. Among these meshs, “least worse” carry a value lower than $1{,}36 \times 10^{-4}$ , that is to say 40% of the maximum value. It is rather effective since the large variations will have been trapped.
The consequence of these remarks is that it is advisable to make a first analysis of the distribution of the values of the field before choosing the type and the values of the criteria of refinement. It is indeed useless, even expensive in term of increase in the size of grid, to refine in zones where the field is not very strong. The adaptation will be all the more powerful as one will have known to reduce the meshs to strong value until obtaining a balance in the grid.

# 6 Bibliography

1) G. Nicolas, T. Fouquet: “Software LOBSTER - Volume 1 - general Presentation”, report EDF H-I23-2008-04107-FR, October 2014.

2) G. Nicolas, T. Fouquet: “Software LOBSTER - Volume 2 – Algorithms of refinement and déraffinement of grids”, report EDF H-I23-2008-04108-FR, October 2014.

3) G. Nicolas, T. Fouquet: “Software LOBSTER - Volume 3 – Interfaces with, computer codes” EDF the report H-I23-2008-04118-FR, October 2014.

4) G. Nicolas, T. Fouquet: “Software LOBSTER - Volume 4 – Structures of data”, EDF report H-I23-2008-04120-FR, October 2014.

5) G. Nicolas and T. Fouquet, “ *Adaptive Mesh Refinement for Conformal Hexahedral Meshes* ”, Finite Elements in Analysis and Design, vol. 67, pp. 1-12, 2013, DOI: 10.1016/j.finel.2012.11.008