To introduce a new Summarized

elementary computation:

This document describes what it is necessary to do to introduce a new elementary computation into *Code_Aster* .

In a few words, it is necessary:

- to introduce a small block of text into the catalog of one `type_element`
- to write a new routine FORTRAN of name `te00ij.f where` 00 `ij is` a number with 4 digits Contents

1

1 For

Code_Aster, an elementary computation corresponds to a couple (standard of finite element, computation option). Examples of the types of finite element (type_ element): MEDKTR

- 3: triangular element DKT with 3 nodes THER_
- PENTA15: element of thermal pentahedron to 15 nodes Examples

of computation options (option): RIGI_MECA

- : computation of the stiffness (elastic behavior) FLUX_ELGA
- : computation of heat flux knowing the temperature with the nodes In

the rest of this document, the example which will be used to us as discussion thread will be that of the computation of heat flux at the points of integration (FLUX_ELGA) of elements PENTA 15 of the modelization “3D” of the “THERMAL” phenomenon (type_ element = THER_PENTA15). One

will suppose that this elementary computation does not exist still but that option FLUX_ELGA already exists (for other finite elements) and that the type_ element THER_ PENTA15 also exists (it can already calculate other options). In this document, we will try to answer the questions: What is necessary

- to make to carry out this new elementary computation? Which
- source files is it necessary to modify or add? Other

relative questions with the finite elements are treated in other documents: [D5.02

- .03] How to introduce a new elementary computation option? (for example a new postprocessing) [D5.02
- .04] How to introduce a new family of finite elements (modelization)? [D5.02
- .01] How to introduce a new quantity or new components into an existing quantity? [D5.02
- .02] How to introduce a new type of mesh (type_ mesh) or a new element of reference (ELREFE)? We

already said in the abstract that the introduction of a new elementary computation into Code_Aster required 2 actions: the addition

- of a block of text in the catalog of the type_ element (here THER_ PENTA15) the addition
- (or modification) of a routine FORTRAN of name te00ij.f We

will detail these two actions successively. Modification

2 of the catalog of type_element the THER_PENTA15 To find

2.1 the name of the file to modify the first

difficulty of solving is to find the name of the file catalogues to modify. Let us notice already that the name of type_element which concerns us (THER_PENTA 15) us is not inevitably familiar. We can by means of discover it the command AFFE_MODELE on a mesh containing of the PENTA15 : MOTH = AFFE_MODELE

```
(MAILLAGE  = MAIL, AFFE=_F (
                     TOUT = "OUI', MODELISATION = "3D", PHENOMENE
                                  = "THERMAL")) In
```

the file .mess we can then see: ON

```
 the 132 MESHES          OF MAILLAGE the MAIL ONE A REQUEST
    the ASSIGNMENT OF 132 ONE A          PU
    TO AFFECT 124 MODELISATION ELEMENT

    FINI      TYPE of THEM NETS      3D NOMBRE          THER
    _PENTA                15 PENTA15      40 3D THER
    _FACE                 6 TRIA6 4       3D THER
    _FACE                 8 QUAD8 80       We                    see
```

that the modelization called "3D" applied to PENTA15 led to the assignment of finite elements of the type THER_PENTA 15. It is the name which we seek. We

could also have found this name by consulting the catalog: … /NEW11

```
/catalo/compelem/phenomene_modelisation__.catastrophes… PHENOMENE

   __THERMAL  CODE       "HT"…  MODELISATION

       __"3D" DIM 3               3 CODE "   3D_" ATTRIBUT
              __DIM_TOPO  _MAILLE=X3 MAILLE
              HEXA8 ELEMENT_      _THER_HEXA8 MAILLE
              PENTA6 ELEMENT_     _THER_PENTA6 MAILLE
              TETRA4 ELEMENT_     _THER_TETRA4 MAILLE
              PYRAM5 ELEMENT_     _THER_PYRAM5 MAILLE
              QUAD4 ELEMENT_      _THER_FACE4 MAILLE
              TRIA3 ELEMENT_      _THER_FACE3 MAILLE
              HEXA27 ELEMENT_     _THER_HEXA27 MAILLE
              HEXA20 ELEMENT_     _THER_HEXA20 MAILLE
              PENTA15 ELEMENT_    _THER_PENTA15 MAILLE
              TETRA10 ELEMENT_    _THER_TETRA10 MAILLE
              PYRAM13 ELEMENT_    _THER_PYRAM13 MAILLE
              QUAD9 ELEMENT_      _THER_FACE9 MAILLE
              QUAD8 ELEMENT_      _THER_FACE8 MAILLE
              TRIA6 ELEMENT_      _THER_FACE6 This catalog
```

gives the name of all type_element associated with the various finite elements of the modelization "3D". For the type of mesh PENTA15 , the name of type_element is well THER_PENTA 15. Once

known name THER_PENTA 15, to find the name of the catalog concerning this finite element, it is necessary to make a grep in the files of catalo/typelem /*: grep - L
THER_PENTA15 /aster/NEW11/catalo/typelem/ *. What should give: /aster
NEW 11/typel EM to obstruct _th 3d_3.catastrophes It is

the name of the file which we must modify. If we

publish this file, we see that it contains a certain number of headings of heading: ENTETE

```
ELEMENT THER_HEXA20 MAILLE      HEXA20 ELREFE
   H20 GAUSS  __        RIGI=FPG  27 MASS=FPG  27 GANO=FPG  8 ELREFE
   QU8 GAUSS  __        RIGI=FPG  9 MASS=FPG   9 ENTETE
ELEMENT THER_HEXA27 MAILLE      HEXA27 ELREFE
   H27 GAUSS  __        RIGI=FPG  27 MASS=FPG  27 GANO=FPG  8 ELREFE
   QU9 GAUSS  __        RIGI=FPG  9 MASS=FPG   9 ENTETE
ELEMENT THER_HEXA8 MAILLE       HEXA8 ELREFE
   HE8 GAUSS  __        RIGI=FPG  8 MASS=FPG   8 GANO=FPG   8 ELREFE
   QU4 GAUSS  __        RIGI=FPG  4 MASS=FPG   4 ENTETE
ELEMENT THER_PENTA15 MAILLE     PENTA15 ELREFE
   P15 GAUSS  __        RIGI=FPG  21 MASS=FPG  21 GANO=FPG  21 ELREFE
   QU8 GAUSS  __        RIGI=FPG  9 MASS=FPG   9 ELREFE
   TR6 GAUSS  __        RIGI=FPG  6 MASS=FPG   6 One
```

of these headings relates to our element (THER_PENTA 15), but this file catalog type_element relates to also all the others others heading. What wants

to say that our new elementary computation will be (by default) accessible to all the finite elements described in the file catalogues. In general

, it is what one wishes because if one can do a calculation on a PENTA15 , one can also do it for the TETRA4, the PENTA 6, … In

the continuation of the document, one continues to refer to element THER_PENTA 15, but actually , our development relates to all the voluminal elements of the modelization “3D” (THER _TETRA4, THER_ TETRA 10,…, THER_HEXA 27). If one

did not wish to make this elementary computation available for the HEXA8 for example , one would modify the heading of L “HEXA 8: ENTETE __

```
ELEMENT THER_HEXA8 MAILLE       HEXA8 ELREFE
   HE8 GAUSS  __        RIGI=FPG  8 MASS=FPG   8 GANO=FPG   8 ELREFE
   QU4 GAUSS  __        RIGI=FPG  4 MASS=FPG   4 OPTION
   FLUX_ELGA   the -1 number
```

“- 1” compared to L” option FLUX_ELGA indicates that this elementary computation is not available for the HEXA8. A fatal error will be emitted if the user tries to use it. To modify

2.2 the file gener_th3d_3.catastrophes the block

of text to be added is: FLUX_ELGA

```
62 IN   CCAMASS      PCAMASS  NGEOMER  PGEOMER  CMATERC
                              PMATERC  DDL_THER  PTEMPER CTEMPSR
                              PTEMPSR  OUT EFLUXPG
                         PFLUX_R  In this
```

block of text, one can recognize different items: the name of

- the option which one wants “to carry out” (FLUX_ELGA) the number
- “62” which is the number of the routine te00ij.f associated with elementary computation here (: te0062.f) the block of
- the inputted fields of elementary computation (what follows key word IN) the block of

- the output fields of elementary computation (what follows key word OUT) Note :

a field is

either "IN" or " OUT", it cannot be "INOUT". The blocks of the fields "IN" and " OUT" are described by lists of couples (mode_local , parameter) . Here, there are 5 fields "IN" and only one field "OUT". Output field

2.2.1 Let us start

with the field "OUT" (elementary computation drank): PFLUX_R (

the parameter) is the name given to the flux field result for the option FLUX_ELGA. This name was selected when the catalog of the option (/aster/NEW11/catalo/options/FLUX_ELGA.cata) was introduced into the code. This name is used in all the catalogs of the finite elements which calculate option FLUX_ELGA. EFLUXPG (the _local

mode) is a local name with the catalog which one modifies. It makes it possible to describe structure of the local flux field for elements THER_PENTA15. We will return there later. Let us look at what

the catalog of option FLUX_ELGA concerning its output field contains: FLUX_ELGA <<

```
FLUX_ELGA
  : CALCUL  FLUX WITH Gauss points >> OPTION IN
PGEOMER
   GEOM_
        R << COORDINATED   OF NODES >>… OUT PFLUX
_R

   FLUX_
        R ELGA    One sees    that
```

parameter field PFLUX_R is a field of quantity FLUX_R and that it is a field by elements with Gauss points (ELGA) It is the "

schedule of conditions" of the development which we must carry out: we must calculate a field of FLUX_R on Gauss points of the THER_PENTA15. The choice of the type of field (here ELGA) is imposed by the option. On the other hand the components of the quantity FLUX_R which one will use are with the choice of element THER_PENTA15. It is the object of mode_local EFLUXPG . This _local mode

is described in a block of text of the catalog (in heading MODE_LOCAL) : MODE_LOCAL __…

```
EFLUXPG =
    FLUX
    _R ELGA   __RIGI (   FLUX FLUY FLUZ     ) One    sees there
that
```

this mode_local relates to quantity FLUX_R well and that it describes well structure of a field "ELGA" (it is in the schedule of conditions imposed by the option). The choice of the components FLUX, FLUY and FLUZ does not have anything astonishing since the element is 3D. The name of the components

that one can use in the description of a mode_local is given in the catalog of quantities (/aster/NEW11/ catalo/compelem/grandeur_simple__.cata file
). One can read there concerning quantity FLUX_R: FLUX _R = R FLUX

```
    FLUY   FLUZ    FLUX     _SUP     FLUY   _SUP FLUZ   _SUP FLUX_   INF
FLUY
                                          _INF FLUZ   _INF Normally   ,
in
```

this catalog, line of comment explains the meaning of each one of these components: << Standard FLUX_R

```
: R vectorial Flux     of heat in a material point of the continuous field
   : PHI = - lambda.gradient (T) FLUX: following
      component OX of PHI FLUY: following
      component OY of PHI FLUZ: following
      component OZ of PHI… Inputted fields
```

2.2.2 the list of the inputted fields

has same structure as that the output fields: it is a list of couples (mode_local, parameter). The parameters are imposed by the catalog of the option. For FLUX_ELGA one finds in the catalog: FLUX_ELGA << FLUX_ELGA

```
: CALCUL
        FLUX WITH GAUSS POINTS >> OPTION IN PGEOMER
GEOM
   _R <
        < PGEOMER  : COORDINATES
        OF THE NODES >> PMATERC ADRSJEVE
        << PMATERC  : MATERIAL FIELD
        >> PCAORIE CAORIE <
        < PCAORIE  : ORIENTATION
        LOCAL OF A BEAM ELEMENT OR PIPE, RESULTING FROM AFFE_CARA_ELEM
           KEY WORD ORIENTATION >> PCADISK CADISK <
        < PCADISK  : CHARACTERISTIC
        OF DISCRET, NEED FOR PROVIDING
           PRODUCT CONCEPT LE PAR AFFE_CARA_ELEM >> PCAGNPO CAGNPO  <
        < PCAGNPO  : GEOMETRICAL
        CHARACTERISTICS OF A SECTION OF POUTRE, NEED FOR PROVIDING
           PRODUCT CONCEPT LE PAR AFFE_CARA_ELEM >> PCACOQU CACOQU  <
        < PCACOQU  : CHARACTERISTIC
        OF COQUE, NEED FOR PROVIDING
           PRODUCT CONCEPT LE PAR AFFE_CARA_ELEM >> PCAMASS CAMASS  <
        < PCAMASS  : CHARACTERISTIC
        OF MASSIF, NEED FOR PROVIDING
           PRODUCT CONCEPT LE PAR AFFE_CARA_ELEM >> PTEMPER TEMP_R  <
        < PTEMPER  : TEMPERATURES
        CURRENT TIME >> PTEMPSR INST_R <
        < PTEMPSR  : CURRENT
        TIME >>  PNUMCOR NUMC_I <
        < PNUMCOR  : NIVEAU
        AND COUCHE OF A MULTI-LAYER MATERIAU >> PHARMON HARMON <
        < PHARMON  : NUMERO
        Of HARMONIC OF FOURIER >> Remarque: Texts
```

written

between "…" are comments which are printed in certain error messages in order to help the user to understand the context of its error. One should not hesitate to improve them or to enrich them. The 11 parameters of entry of option FLUX_ELGA are thus : PGEOMER GEOM_R PMATERC ADRSJEVE

```
PCAORIE  CAORIE PCADISK
CADISK  PCAGNPO
CAGNPO  PCACOQU
CACOQU  PCAMASS
CAMASS  PTEMPER
TEMP_R  PTEMPSR
INST  _R PNUMCOR
NUMC  _I PHARMON
HARMON  It
is necessary to choose  in
```

this list the parameters which will be useful for type_element the THER_PENTA15. After reflection , the 5 following parameters are retained: PTEMPER: it is

- the field of temperature of which it gradient PGEOMER should be calculated: the field
- of geometry of the element (its coordinates) is necessary to compute: the gradient. PMATERC: the material field
- is necessary to be able to know thermal conductivity (average) PCAMASS: it is
- the field containing the possible local coordinate system of the elements 3D. It is necessary if the material is not isotropic: the anisotropic characteristics are given in a local coordinate system. PTEMPSR: this field
- is used to transmit time of computation. It is necessary here because, in thermal, when the coefficients material are functions (DEFI_MATERIAU/THER_FO for example), these functions can depend on time. Once one

retained the parameters useful for type_element, it is necessary to allot to them to each one a mode_local. One will choose: DDL_THER PTEMPER

```
NGEOMER PGEOMER
CCAMASS  PCAMASS
CMATERC  PMATERC
CTEMPSR  PTEMPSR
These modes  local
```

must be described in heading MODE_LOCAL of the catalog : MODE _LOCAL…

```
DDL_THER = TEMP_R

    ELNO IDEN (TEMP   )… NGEOMER  =     GEOM
    _R
    ELNO  IDEN (X   Y Z)… CMATERC                =        ADRSJEVE

    ELEM  (I1)… CCAMASS          =       CAMASS

    ELEM  (C ALPHA   BETA     KAPPA         X Y    Z)…     CTEMPSR
                                            =       INST
    _R
    ELEM  (INST DELTAT   THETA     KHI     R RHO   )…    Let us comment on
                                                 these
    local
```

modes: DDL_THER This _local

2.2.2.1 mode indicates

that the local field expected by the finite element is a field on the nodes of element (ELNO). All the nodes of

the element (15 for a PENTA15) carry same components (IDEN). The list of the components carried by these nodes is reduced to only one component of quantity TEMP_R: TEMP. NGEOMER This _ mode

2.2.2.2 local indicates

that the local field expected by the finite element is a field on the nodes of element (ELNO). All the nodes carry

same components (IDEN). The list of the components
carried by these nodes is X, Y, Z because L " element is 3D. CMATERC This _ mode

2.2.2.3 local indicates

that the local field expected by L" finite element is a constant field on element (ELEM). The list of the components

carried by this element is I1. Note: The material field

> *is always associated with a _local mode (in general called CMATERC) definite in the same way : CMATERC = ADRSJEVE ELEM (I1) That translated the fact*

that one cannot affect (for example) materials different on the nodes from an element. There is one material by mesh . In addition, for

reasons of performance, the data-processing structure representing the material on the level of an element known as "is coded" (one speaks then about coded material), it is represented by an integer (I1 component of quantity ADRSJEVE) which is a memory address . One will see later how this material is used in the utility routines. CCAMASS This mode_

2.2.2.4 local indicates

that the local field expected by the finite element is a constant field on element (ELEM). 7 components

(C, ALPHA,…) are real numbers which make it possible to represent the change of Total reference → Local will not say We

here. CTEMPSR This mode_

2.2.2.5 local indicates

that the local field expected by the finite element is a constant field on element (ELEM). It is noticed that 6

components are indicated in the mode_local: INST , DELTAT , … However only the value of time INST is useful for us to be able to evaluate possible functions of time. The other components
will not be used and they could be withdrawn from the mode_local. However , it should not

be done too brutally. Indeed, mode_local CTEMPSR is used by other options which they need more than information (DELTAT: value of time step ,…) If one wanted to improve

the legibility of the catalog of our element, one could duplicate this mode_local: CTEMPSR = INST _R

```
ELEM  (INST DELTAT   THETA   KHI  R RHO  ) CTEMPS  1  =  INST  _R
ELEM  (INST) For   our   option
```

SIEF_ELGA_TEMP, one could then describe the local field of time by couple (CTEMPS1 PTEMPSR) To write (or modify

3) routine FORTRAN te0062.f We already

saw that the catalog of element THER_PENTA15 indicated that the computation of option FLUX_ELGA will occur in routine FORTRAN te0062.f. The object of this routine is to calculate (for a finite element) the local flux field on Gauss points of the element. In a general way
, a routine te00ij.f always has like drank to calculate fields “OUT” to leave its fields “IN”. Arguments of the routine

3.1 All the routines

te00ij.f have the same arguments . The reason is that the routine which calls them (te0000.f) is written once and for all and that one does not want to change it to each time one adds an elementary computation. The two only arguments

of the routines te00ij.f are arguments of entry : OPTION is a character string
containing the name of the option (for us: FLUX_ELGA) NOMTE is a character string
containing the name of type_element (for us: THER_PENTA15) These 2 arguments can be

used (or not). Argument OPTION is used

when one treats in the same routine te00ij.f 2 options (or more) which resemble each other . One can then write tests

like: … IF (OPTION.EQ. “FLUX_ELGA”

```
) THEN
… ELSE IF (OPTION.EQ. "FLUX_ELNO

") THEN… ENDIF In the same way,
   one
in general
```

treats all the elements of the same modelization (TETRA4, TETRA10,…) in the same routine te00ij.f. Argument NOMTE makes it possible to distinguish certain processing . One could for example imagine a small block of code specific to the pyramidal elements. But the true arguments of

the routines te00ij.f are actually their fields " IN” and their fields “OUT”. These arguments are “underground ” and one reaches it via 2 utility routines JEVECH and TECACH which one will present in the following paragraph. Presentation of some utilities

3.2 used in the te0062.f Routine JEVECH This routine makes it possible

3.2.1 to recover

the memory address of a local field. For example, in the routine te0062.f, one finds: CAL JEVECH ("PGEOMER ", “It, IGEOM)

```
the 1st argument of routine JEVECH
```

is the name of the parameter which interests us. The 2nd argument is “documentary
” (it is not used in the code). It indicates if the parameter is a field “IN” (access in reading L) or a field “OUT” (access in writing E). The 3rd argument (of output) is L
“addresses memory zone containing the local field. C” is to the programmer of the routine

te0062.f to ensure coherence between the use which he makes of this address and what is written in the catalog of the element concerning this parameter: Since parameter PGEOMER

is associated with quantity GEOM_R and that this quantity is of real type , address IGEOM is an address in common JEVEUX ZR Since mode_local NGEOMER (associate with PGEOMER) is described in the catalog by NGEOMER = GEOM_R ELNO IDEN (X Y Z) , it is to the programmer *of* knowing that the local field represented in memory with L "addresses ZR (IGEOM) is length 3*15 (3 components X, Y, Z for each of the 15 nodes) . More precisely , the programmer

must know qu" it will find in the JEVEUX memory: ZR (IGEOM-1+1) - > "X" of node 1 ZR (

```
IGEOM-1+2) - > "Y" of node 1 ZR (
IGEOM-1+3) - > "Z" of node 1 ZR (
IGEOM-1+4) - > "X" of node 2 ZR (
IGEOM-1+5) - > "X" of node 2…
ZR (IGEOM-1+45) - > "Z" of the node
15 Routine
ELREF4 to supplement… Routine
```

3.2.2 DFDM3D

to supplement… Routine

3.2.3 RCVALA

to supplement… Routine

3.2.4 TE0062 SUBROUTINE

TE0062 (OPTION

3.3 , NOMTE) IMPLICIT

```
      NUN CHARACTER*16 OPTION, NOMTE
      C --- DEBUT
      DECLARATIONS NORMALISEES

JEVEUX ---------------------------- INTEGER ZI COMMON /IVARJE/ZI (1) REAL
      *8 ZR COMMON
      /RVARJE/ZR (1) COMPLEX
      *16 ZC
      COMMON /CVARJE/ZC (1)
      LOGICAL ZL COMMON
      /LVARJE/ZL (1) CHARACTER
      *8 ZK
      8 CHARACTER*16 ZK16 CHARACTER
      *24 ZK24
      CHARACTER*32 ZK32
      CHARACTER*80 ZK80
      COMMON /KVARJE/ZK8
      (1), ZK 16(1), ZK 24(1)
      , ZK 32(1), ZK 80(1) C --- FIN DECLARATIONS NORMALISEES
JEVEUX ------------------------------ INTEGER ICODRE CHARACTER*8 NOMRES (1

      ) REAL*8 LAMBDA
      , FLUXX, FLUXY, FLUXZ, DFDX
      (27), DFDY (27), DFDZ (27), POIDS INTEGER JGANO, IPOIDS, IVF, IDFDE,
IGEOM
      , IMATE, NO, KP, NPG1, I, IFLUX, & ITEMPS, ITEMPE, NDIM, OUR
C----------------------------------------------------------------------

C -- recovery of information


concerning     the element of reference: CAL ELREF4 ("", "RIGI", NDIM, NO,
OUR
      , NPG1, IPOIDS, IVF, IDFDE, JGANO) C -- recovery of the addresses of

    the local fields: CAL JEVECH ("PGEOMER", "IT, IGEOM) CAL
      JEVECH ("PMATERC", "IT, IMATE) CAL
      JEVECH ("PTEMPSR", "IT, ITEMPS) CAL
      JEVECH ("PTEMPER", "IT, ITEMPE) CAL
      JEVECH ("PFLUX_R", "E", IFLUX) C
      -- recovery of conductivity

    LAMBDA: CAL RCVALA (ZI (IMATE),"", "THER", 1,
      "INST", ZR (ITEMPS), 1, "LAMBDA", & LAMBDA, ICODRE, 1) C -- buckle on
                Gauss points


:     C 20 KP=1, NPG1 C -- recovery
      of derivatives
of       the shape functions C (on the real element): CAL DFDM
         3D (NO, KP, IPOIDS, IDFDE, ZR (
        IGEOM), DFDX, DFDY, DFDZ, POIDS) C -- computation of the gradient of
the field of

temperature       : FLUXX=0.0D0 FLUXY=0.0D0 FLUXZ=0.0D0
        C 10 I=1, NO
        FLUXX=FLUXX
```

```
            +ZR (ITEMPE

            - 1+I) *DFDX (I)
              FLUXY=FLUXY+ZR (ITEMPE-1+I) *DFDY (I)
              FLUXZ=FLUXZ+ZR (ITEMPE-1+I) *DFDZ (I)
              10 CONTINUE C -- computation of flux and
        storage   in

           the field result: ZR (IFLUX+ (KP-1) *3) =-LAMBDA*FLUXX
             ZR (IFLUX+ (KP-1) *3+1)  =-LAMBDA*FLUXY
             ZR (IFLUX+ (KP-1) *3+2) =-LAMBDA*FLUXZ
             20 CONTINUE END Comments on this
        routine

           : It
```

should not be forgotten that this routine

must be able to treat all the elements 3D: TETRA4,…, HEXA27. This is why , tables DFDX , DFDY and DFDZ are dimensioned to 27 which is the “max” amongst nodes of a voluminal element. The family of Gauss points which

is used here is “RIGI “. C” is it which is given in argument of routine ELREF4. This family is coherent with the choice made in the catalog: EFLUXPG = FLUX_R ELGA RIGI (FLUX FLUY FLUZ) routine RCVALA is used

to recover the value of thermal conductivity (LAMBDA). As this parameter can be a function of time, one provides the value of L” urgent in argument of entry of routine RCVALA. gradient of the temperature

is calculated in two times. Routine DFDM3D makes it possible to calculate the gradient of the shape functions on the element into 1 Gauss point given. One can then use the gradient of the shape functions and the value of the temperature on the nodes to compute: the gradient of the temperature. The storage of flux calculated with

address IFLUX respects strictly the convention of storage of the local fields: 3 components FLUX, FLUY and FLUZ for each Gauss point . Details not used in the selected

4 example Description of the heading of a type_

4.1 element to supplement… Local Modes ELNO

__DIFF to supplement

4.2 … Conventions of names

for the local

4.3 modes to supplement… to supplement…

Names reserved for

4.4 certain local modes (ddl_meca, ddl_ther, ddl_acou) to supplement … local Fields of

standard elementary vector

4.5 or elementary matrix to supplement… optional Fields

, routine tecach.f

4.6 to supplement… non-available

Elementary computation:

4.7 “-1” to supplement… Family of Gauss points

“MATER”

4.8 to supplement…