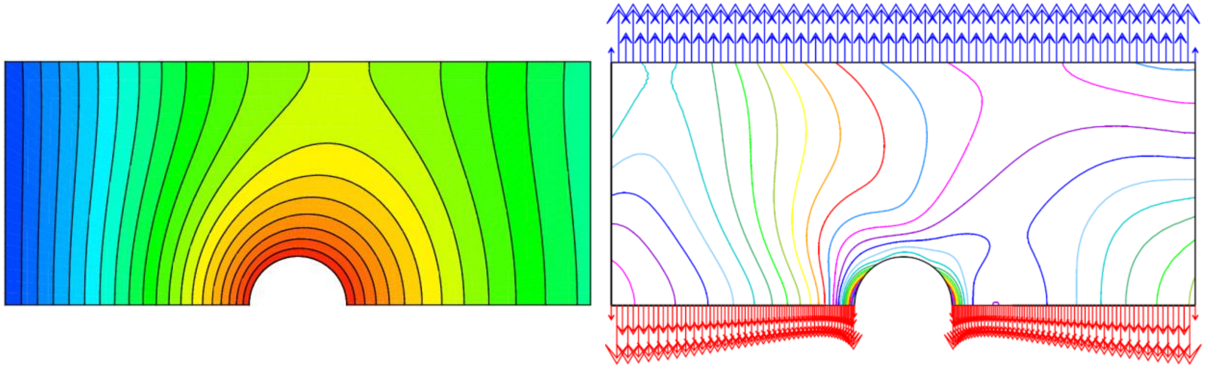
DE LA RECHERCHE À L'INDUSTRIE

www.cea.fr

STARTING WITH CAST3M

THERMOMECHANICAL CALCULATIONS

AVAILABLE ON: **HTTP://WWW-CAST3M.CEA.FR/INDEX.PHP?XML=FORMATIONS**



François DI PAOLA

LAST CHANGES: MARCH 27 2023

SUMMARY

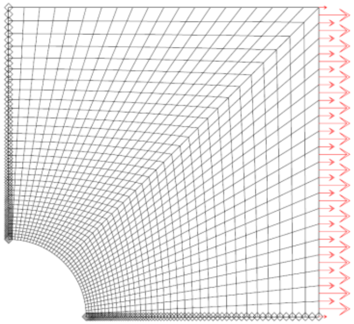
- **Introduction to Cast3M**
- **Gibiane language**
- **Tutorial class**
  thermo-mechanical behavior of a structure
  with a circular cavity
- **Complements**
- **Objects in Gibiane**

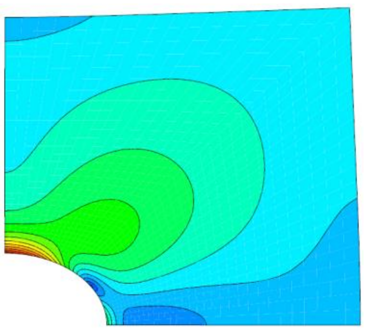
INTRODUCTION TO CAST3M

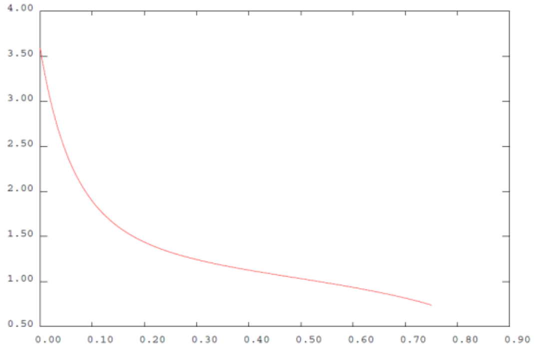
WHAT IS CAST3M?

A simulation software
using the **finite element method**
thermal and **mechanical** analysis of **structures** and **fluids**

- **Partial differential equations** solved thanks to the finite element method
- **Complete software:** solver, pre-processing and post-processing, visualization, reading/writing data…

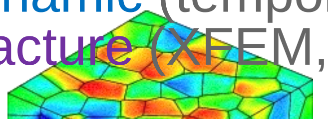


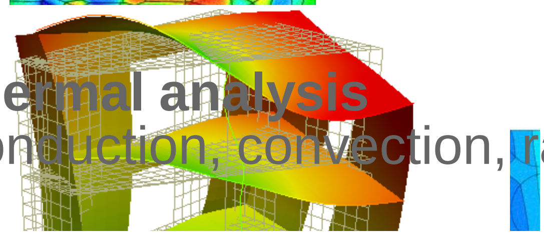




- Based on a programming language: **Gibiane** (objet-oriented)

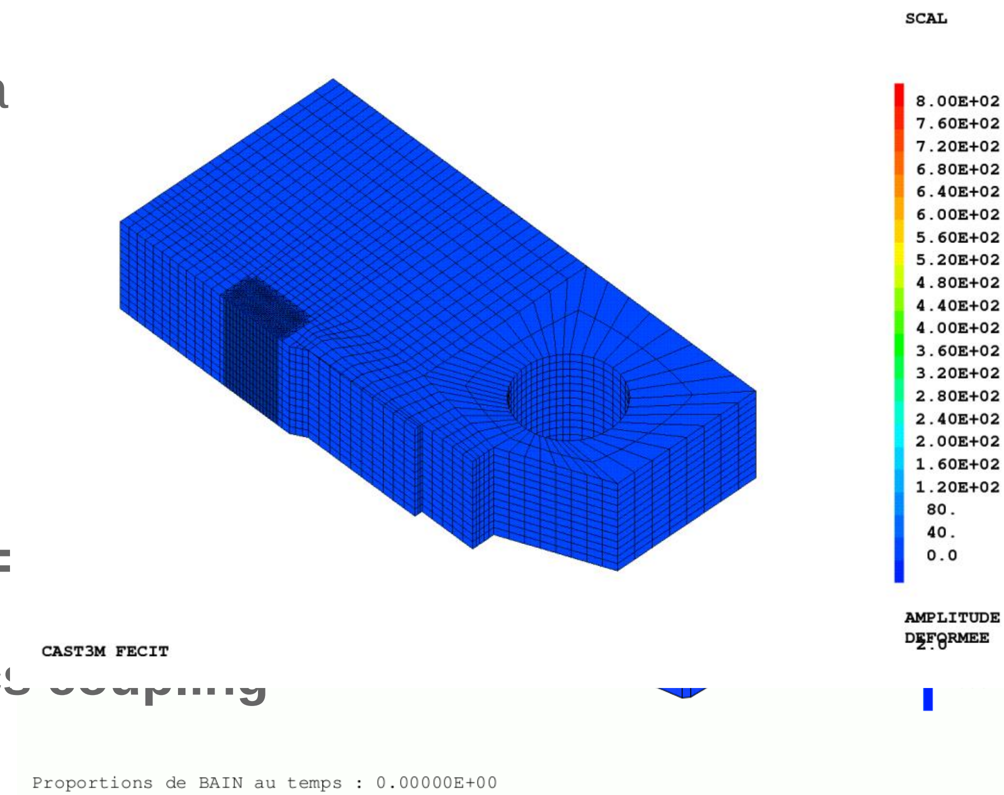
- **Structural mechanics**
  Quasi-static (non linear behavior, geometry, boundary conditions)
  Contact/friction, Buckling
  Dynamic (temporal, modal, fluid structure interaction)
  Fracture (XFEM, dynamic propagation, cohesive zones models, …)





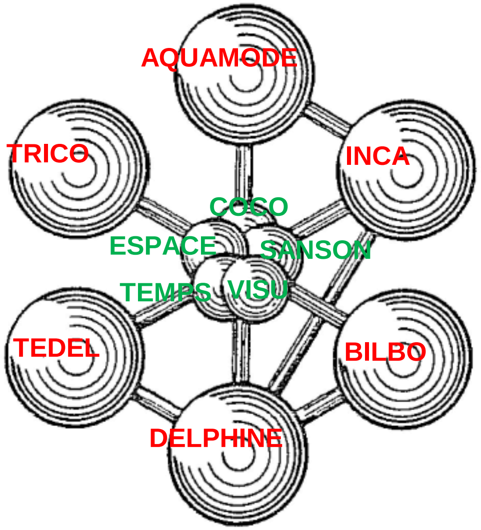
- **Thermal analysis**
  Conduction, convection, ra

- **Fluid mechanics**
- **Metallurgy**
- **Magneto-statics**
- **Multi species diffusion (F**
- **Thermo-hygro-mechanics coupling**



HISTORY

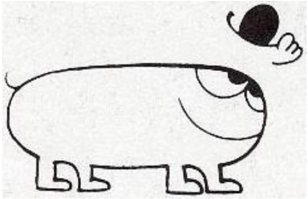
1970 CEASEMT: package of dedicated software:
COCO (mesh), ESPACE, TEMPS, VISU (post-processing),
SANSON (eq. properties)
TEDEL (beams, pipes), TRICO (shells), BILBO (solids)
PASTEL (2D, plasticity) ➔ INCA
DELPHINE (thermal), AQUAMODE (modal analysis)



1981 Start of **GIBI** (meshing software)

1983 Initiation of **Castem 2000** (based on GIBI)

1986 Official release of Castem2000
Procedures, introduction of fluid physics and other physics



1990 Development of dedicated applications (Toutatis, Esus, …)

2000 Castem 2000 ➔ **Cast3M**
Platforms for dev. of dedicated applications (Pléiades, Alliance, …)
New dedicated applications (Brasero, Gerboise, Rotor, …)

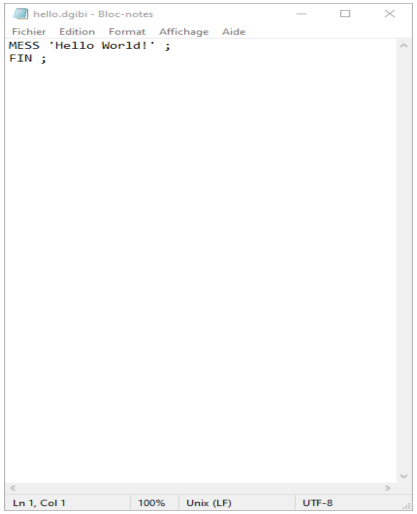
WHAT, WHERE, HOW, HOW MUCH, WHO?

- **Cross platform**
  Windows, Linux, macOS
- **Where can I download Cast3M?**
  http://www-cast3m.cea.fr/index.php?page=dlcastem
- **Access to the source code**
  Open collaboration
  Compiler / Linker are provided
- **Price**
  Free license, for education and research use
  Paid license, for enterprise use
- **Some users/customers**
  Universities, engineering schools …
  IRSN, EDF, SNCF, CNRS, Framatome, Air Liquide, CERN, …

  Reference FEM tool for IRSN for safety analysis of French nuclear installations
  Reference tool for Framatome for fracture mechanics

HOW TO LAUNCH CAST3M?

- **1) Write a Gibiane script in a text file and save it in a working directory**



- **2) Open a terminal / command prompt and go to the working directory**

  here are some basic Linux commands:

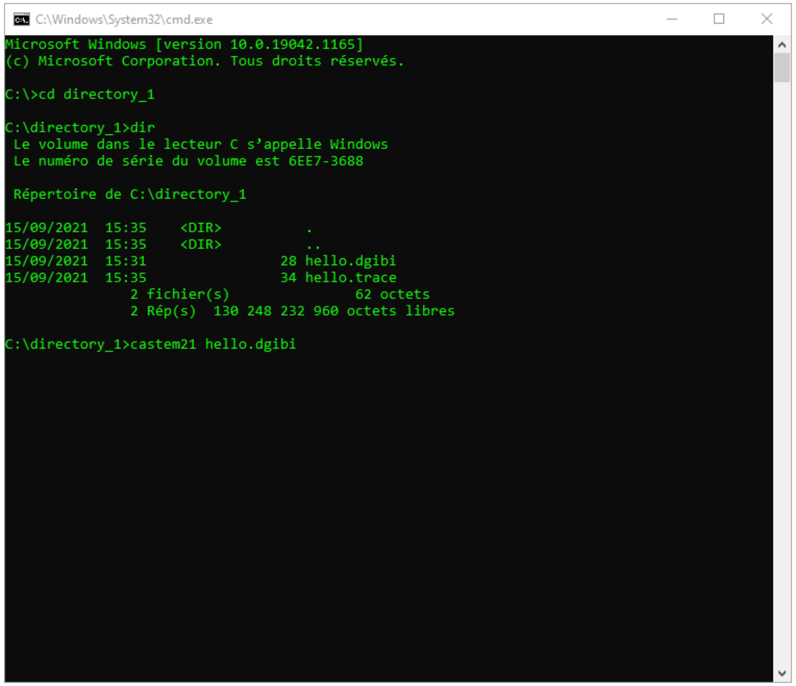
  `ls` list present files
  `cd foo/bar` change directory
  `pwd` print the current working directory

- **3) Launch Cast3M on this file**

  `castem22 hello.dgibi`

You can also use it without .dgibi file (interactive mode):

`castem22`



THE CAST3M WEB SITE

- **The Cast3M web site: "*the place to be*"**

  http://www-cast3m.cea.fr

- *Cast3M presentation*
- *Training courses and video tutorials*
- *Documentation (manual pages, source code, examples)*
- *Anomaly and development reports*
- *Downloads*
- *Contact: Cast3M support*
- *Community: mailing list, Cast3M club*

GIBIANE: THE USER LANGAGE

INTRODUCTION TO GIBIANE

- **Language dedicated to FE calculation but also a programming language**
  - Classical objects (integer, floating-point, string, logical , tables…)
  - Flow control statements
  - Loops and iterations
  - Subroutines
  - Recursion

- **Interpreted language**
  - You can run the program as soon as you make changes to the file
  - You can run it in a interactive mode
- **Object-oriented language**
  - Everything in the program is treated as an object
  - No need to declare variables or to specify the type of a variable
- **French key words**
- **Easy to learn, easy to read**

GIBIANE: SYNTAX RULES

- **Statements lines**
  - 500 characters max per statement
  - A statement can be written on several lines
  - End with a semicolon **;**
  - The assignment operator is the equals sign **=**

- **Case insensitive**

```
TOTO = 3.14 ;
A    = 2. * tOTo ;
```

  variable `A` has the value of `6.28`

  excepted characters strings `'blabla'` ≠ `'BLABLA'`
  enclosed by single quotes `mot1 = 'Hello bro' ;`

- **Program ends**
  - with statement **`FIN ;`** ➔ Cast3M stop
  - empty line or EOF ➔ interactive mode

- A ***** in the **first column** means that the rest of the line is a **comment**

- **Empty lines accepted**

GIBIANE: SYNTAX RULES

- **No precedence in operators (from left to rigth)**

  `1+2*3     = 9` ➔ use parentheses
  `1+(2*3) = 7`

- **Prohibitions**

  No tab key ➔ incomprehensible error message
  No double quotes "

- **Guidelines**

  No special characters (é, ç, ~, œ …)
  Use line indentation (with spaces)
  Adjust the text editor
  syntax highlighting, switch tabulations by spaces,…

- **Common errors**

  Semicolon ; forgotten at the end of the statement
  ➔ the statement is not ended
  Single quote ' forgotten at the end of a string
  ➔ the string is not ended

GIBIANE: SYNTAX EXAMPLE

Not good!

```
N = 7 ; F = 1 ;i=1;
REPETER boucle1;
sI (i <eg N) ;
f = F * I ;I=i+1;
SINON;quitter BOUCLE1 ;
finsi;
FIN BOUCLE1;
message F ;FIN ;
```

Good!

```
** Initialisation d'un entier N
N = 7 ;

** Calcul de la factorielle du nombre entier N
F = 1 ;
I = 1 ;
REPETER BOUCLE1 ;
  SI (I <EG N) ;
    F = F * I ;
    I = I + 1 ;
  SINON ;
    QUITTER BOUCLE1 ;
  FINSI ;
FIN     BOUCLE1 ;

** Affichage du resultat
MESSAGE F ;

FIN ;
```

GIBIANE: OBJECTS

- **Definition**
  Any **data/result** with a defined type (possibly a sub-type) and name

- **Objects Names**
  User Defined
  Limited to 8 characters chosen in:
  letters a to z or A to Z, digits (0 to 9) and underscore (_)
  Traps
  more than 8 characters: additional characters ignored
  dash sign – ➔ not allowed
  letters with accents: é, è ➔ not allowed

- **Objects types**
  There are more than 40 types of objects
  A detailed list of the most often used objects is given at the end of the presentation **(link)**

GIBIANE: OBJECTS

- **Examples (non exhaustive)**

```
OBJ1 = 3 ;                            type: ENTIER (integer)
OBJ2 = 3.14 ;                         type: FLOTTANT (floating)
OBJ3 = 'How are you?' ;               type: MOT (string)
OBJ4 = VRAI ;                         type: LOGIQUE (logical)

poin1 = POIN 0. 0. ;                  type: POINT
poin2 = POIN 1. 3. ;                  type: POINT
OBJ5 = DROI 8 poin1 poin2 ;           type: MAILLAGE (mesh)

LIST OBJ5 ;
MAILLAGE 3520406 : 8 élément(S) de type SEG2
0 sous-référence(s)
1ère ligne  numéro élément : 2ème couleur : 3ème... noeud(s)
          1       2       3       4       5       6       7       8
       DEFA    DEFA    DEFA    DEFA    DEFA    DEFA    DEFA    DEFA
          1       3       4       5       6       7       8       9
          3       4       5       6       7       8       9       2
```

GIBIANE: OPERATORS

- **Definition**
  Any **processing** with a name (Gibiane instruction) that creates **new object(s)** from pre-existing object(s)

- **Operators Names**
  Pre-defined
  These are Gibiane instructions
  Case insensitive
  Only the four first characters are necessary and taken into account
  (`DROITE = DROI`)
  Excepted abbreviations
  `DROI` → `D` (or `d`)
  `CERC` → `C` (or `c`)

GIBIANE: OPERATORS

Examples of operator call

Common cases (single object on the left of the equals sign)

```
obj1 = OPER obj2 ;
obj3 = OPER obj4 obj5 ;
obj6 = obj7 OPER obj8 obj9 ;
```

Unusual cases (multiple objects on the left of the equals sign)

```
obj1 obj2 obj3 = OPER obj4 obj5 ;
```

The **"no name"** operator: POINTS creation

in 2 dimensions `Point1 = 0. 0. ;`

in 3 dimensions `Point1 = 0. 0. 0. ;`

GIBIANE: OPERATORS

- **Arguments order**
  do not matters if arguments have different types
  (with a few exceptions pointed in the manual)

  matters if same type arguments

- **Overwriting an object**
  Always possible, the overwritten object does not exists any longer

```
A = ‘Hello' ;     ➔ A has type MOT
B = 28 ;
C = 3 ;
A = B**C ;        ➔ A has type ENTIER, its value is 21952
```

- **Traps**
  Object name = operator name ➔ operator cannot be called excepted if you call it with quotes `A = 'OPER' B C ;` in upper case!

  Objet name `c`, `C`, `d` or `D` !

GIBIANE: DIRECTIVES

- **Definition**
  Statement without assignment operator =
  Does not create a new object

- **Examples**

```
OPTI  'DIME' 3  'ELEM' 'CUB8' ;
OPTI  'TITR' 'Meshing of the structure' ;

DEPL mail1 'PLUS' (28.  -0.3  20.03) ;
```

OPTI is often the first statement of a Gibiane program
It defines Cast3M general options

Examples:

space dimension (1 to 3), elements type, mesh size,
output file name, axial symmetry hypothesis, and others…

GIBIANE: PROCEDURES

- **Definition**
  Set of Gibiane statements having a name with input and output arguments
  Similar to Fortran subroutine or a C function

- **Procedures names**
  As other objects (a procedure is an object with PROCEDUR type)

- **Declaration**

```
DEBP  my_proc  arg_e1*entier  arg_e2*flottant  ...  arg_en*mchaml ;
  statement 1 ;
  statement 2 ;
  ...
  statement k ;
FINP  arg_s1  arg_s2  ...  arg_sm ;
```

GIBIANE: PROCEDURES

- **Calling**
  As an operator or directive

```
obj1 obj2 ... objm = my_proc ent1 flot2 ... champn ;
```

- **Pre-existing procedures in Cast3M**
  List in: http://www-cast3m.cea.fr/index.php?page=notices

PASAPAS ➔ non-linear calculations
FLAMBAGE ➔ buckling calculations
DYNAMIC ➔ dynamics calculations
THERMIC ➔ thermal calculations
G_THETA ➔ line integrals computation (fracture)
…

other procedures to discover in the manual pages

GIBIANE: PROCEDURES

- **Traps**

`FINP` missing

➔ Cast3M stops, error message that can be misunderstood

`FINP` existing but missing **`;`**

➔ Cast3M stops, error message that can be misunderstood

Procedure called before it is declared

➔ Cast3M stops, error message in the = operator difficult to interpret

Debugging

```
INFO OPER ;                    OPTI 'LANG' 'ANGL' ;
```
➔ print the manual page of a operator/directive/procedure

```
OPTI 'DONN' 5 ;
```
➔ stop to run the file `.dgibi`
➔ run from scream: interactive prompt

```
OPTI 'DONN' 3 ;
```
➔ return to run the file `.dgibi` (from where it stops previously)

```
LIST OBJ1 ;
```
➔ print information about the object `OBJ1`

```
LIST 'RESU' OBJ1 ;
```
➔ printing is reduced to the headings

```
OPTI 'DEBU' 1 ;
```
➔ stop on a error inside a procedure

```
TRAC OBJ1 (OBJ2) ;
```
➔ plot an object (mesh, iso-values of fields, deformed mesh, …)

```
MESS 'Here I am!' ;
```
➔ print a message

DOCUMENTATION

- **Manual pages of operators/directives/procedures**
  1) The `INFO` directive, example: `INFO EXTR ;`
  2) Local html page: in the installation directory

  example on a Linux system: /home/john_doe/CAST3M_2022**/doc/index.html**

  example on a Windows system: C:\Cast3M\PCW_22**\doc\index.html**

  3) The web site: http://www-cast3m.cea.fr/index.php?page=notices
  dedicated to the up to date version !

- **Users manual**
  On the web site, see the "Documentation" tab

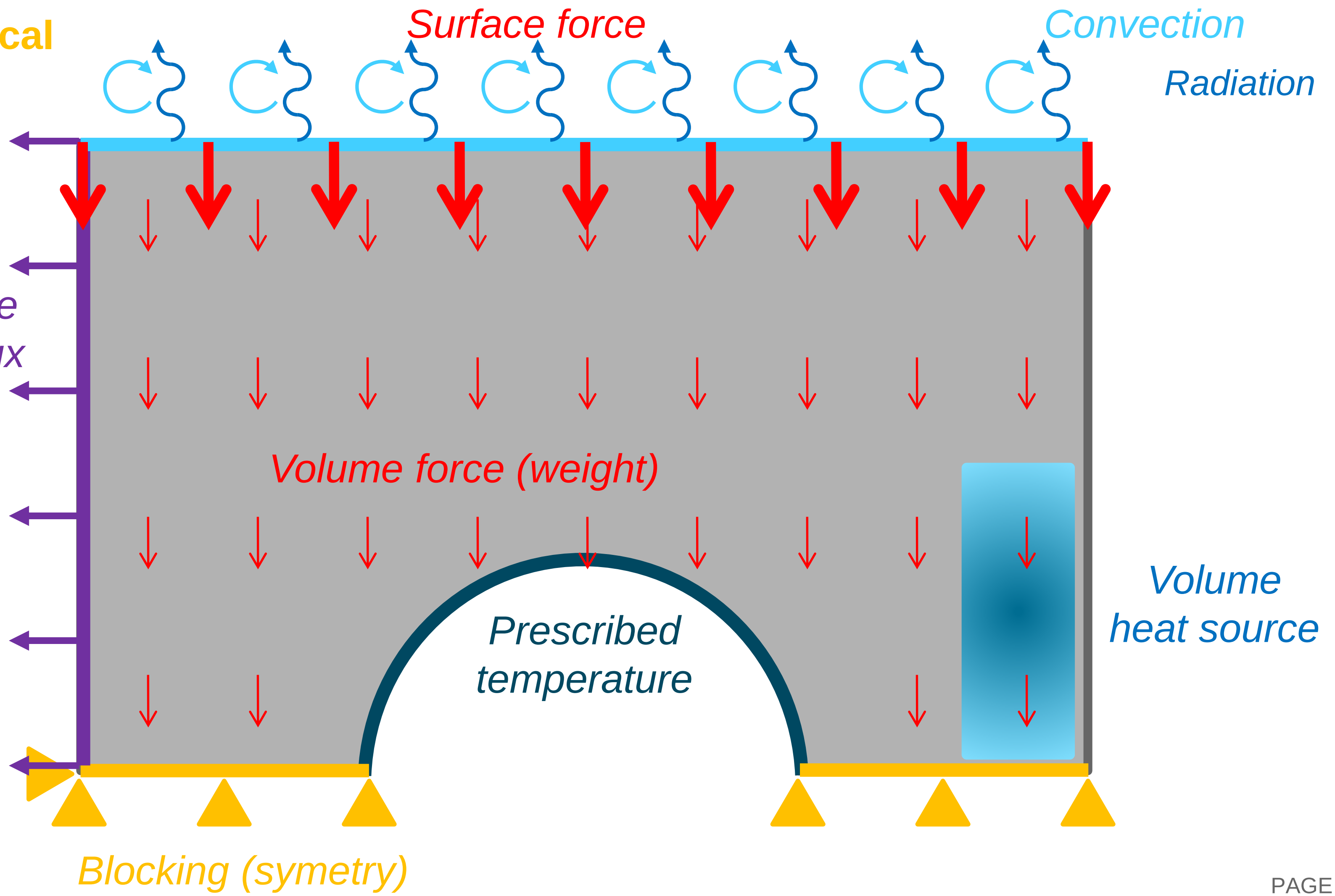
TUTORIAL CLASSES

MODELING OF THE THERMO-MECHANICAL BEHAVIOR OF A STRUCTURE WITH A CIRCULAR CAVITY

PROBLEM DESCRIPTION AND BOUNDARY CONDITIONS

Thermal

Mechanical



REMINDER

Method to perform finite element calculations (4 main steps)

1) Geometry description and meshing

a) Description of points, lines, surfaces, volumes

b) Discretization

2) Mathematical model definition

a) Definition of the model data (type of analyze, formulation, material behavior, types of elements)

b) Definition of the material properties (Young's modulus, density, …)

c) Definition of the geometrical properties (shell thickness, moments of inertia for beams, …)

d) Definition of the boundary conditions and loadings

f) Definition of the initial values

3) Solving the discretized model

a) Elementary stiffness matrix and elementary mass matrix calculation

b) Global matrix calculation

c) Introduction of the loadings and boundary conditions

e) Solving the system of equations

4) Analysis et post-processing

a) Local quantities (strains, stresses, displacements, …)

b) Global quantities (maximal strains, strain energy, …)

SOLUTION FILES

- **The solution files of this tutorial are in the examples base**

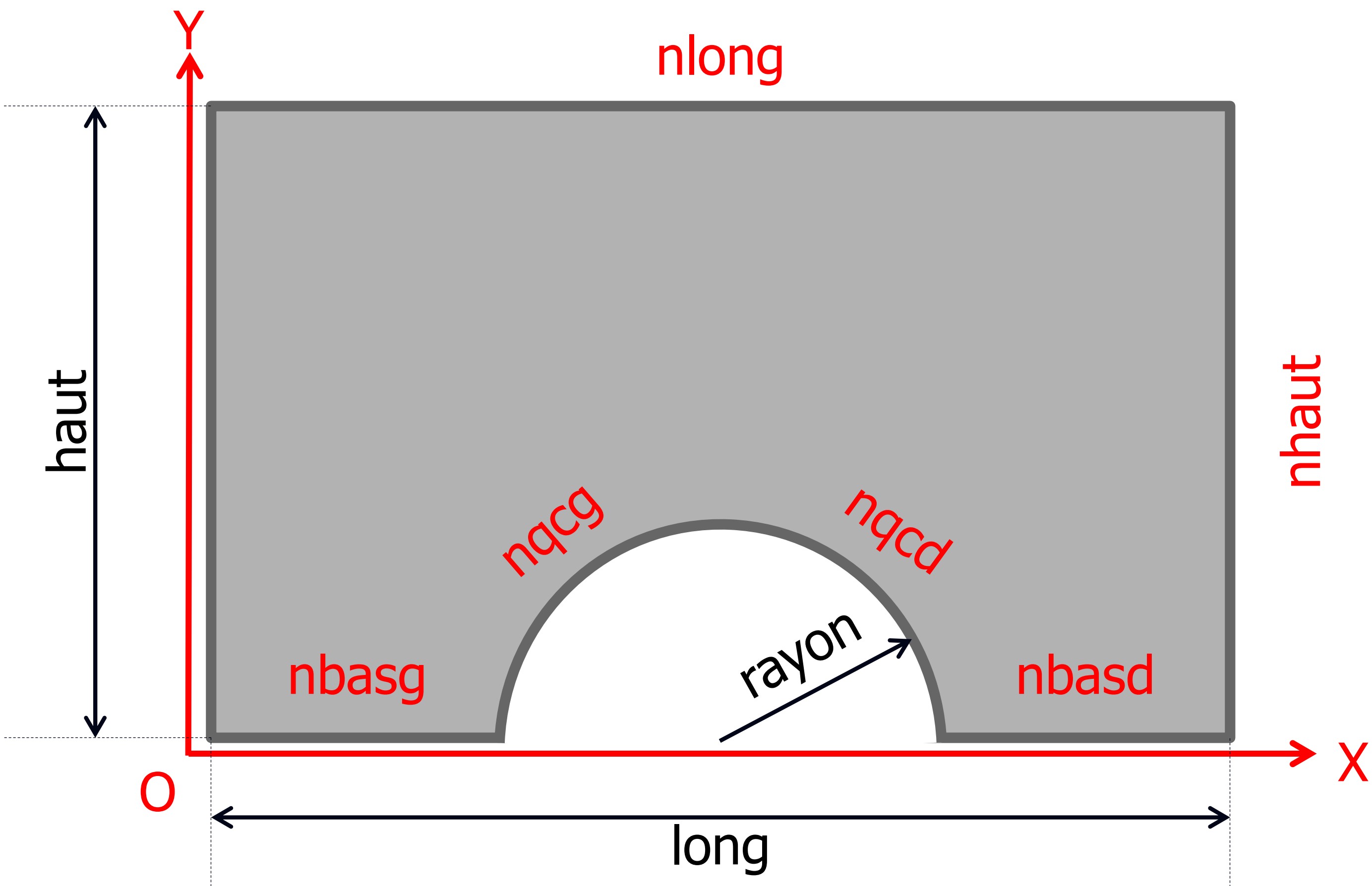
  Download them from the web site:
  http://www-cast3m.cea.fr/index.php?page=exemples

  Three files will be used:

- *formation_debutant_1_maillage.dgibi*
- *formation_debutant_2_thermique.dgibi*
- *formation_debutant_3_mecanique.dgibi*

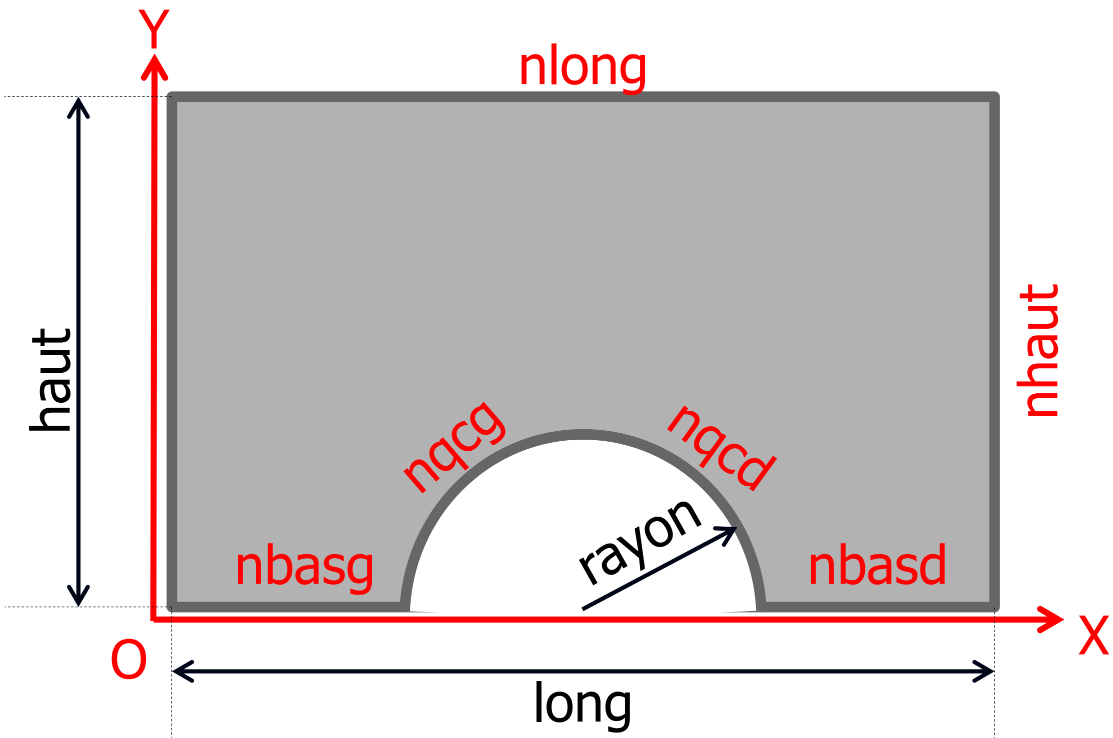
Objective: *to create a parametered mesh of the plate with the hole*

1. *locate the main points*
2. *mesh the closed border*
3. *mesh the internal surface*



General options and parameters

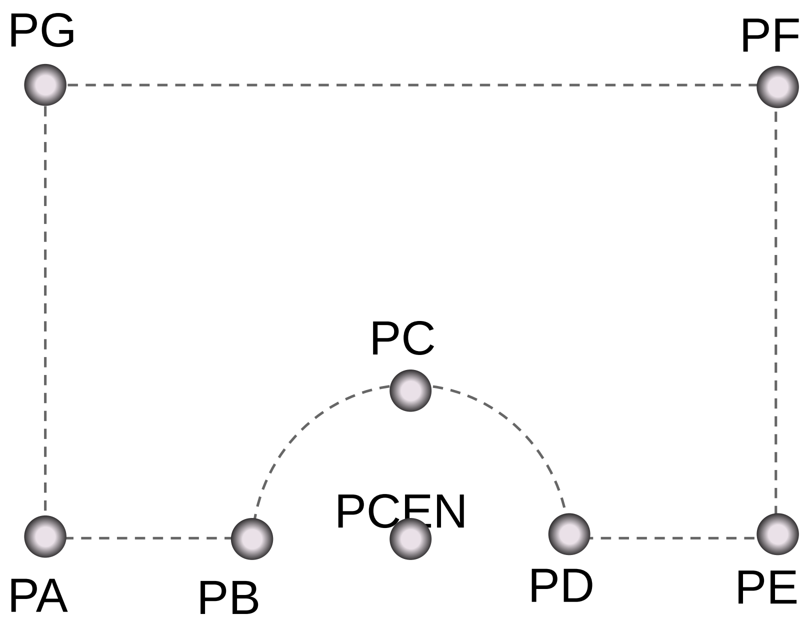
```
* GENERAL OPTIONS AND TYPE OF FINITE ELEMENTS
OPTI 'DIME' 2 'ELEM' 'QUA8' ;

* LENGTHS AND MESH DENSITY DEFINITIONS
LONG      = 24.E-1 ;
HAUT      = 10.E-1 ;
RAYON     = 2.E-1 ;

NLONG     = 24 ;
NHAUT     = 4 ;
NBASG     = 10 ;
NBASD     = 10 ;
NQCG      = 8 ;
NQCD      = 8 ;
```



New objects ENTIER, FLOTTANT, MOT

Points creation

```
* CREATION OF POINTS SUPPORTING THE MESH
PA         = 0. 0. ;
PB         = ((0.5 * LONG) – RAYON) 0. ;
PC         = (0.5 * LONG) RAYON ;
PD         = ((0.5 * LONG) + RAYON) 0. ;
PE         = LONG 0. ;
PF         = LONG HAUT ;
PG         = 0. HAUT ;
PCEN       = (0.5 * LONG) 0. ;
```



new object POINT

Lines and closed contour

```
* STRAIGHT LINES CONSTRUCTION (DROI or D)

LIAB      = DROI NBASG PA PB ;
LIDE      = DROI NBASD PD PE ;
LIEF      = D    NHAUT PE PF ;
LIFG      = D    NLONG PF PG ;
LIGA      = D    NHAUT PG PA ;

* CONSTRUCTION OF CIRCLES (CERC or C)
```

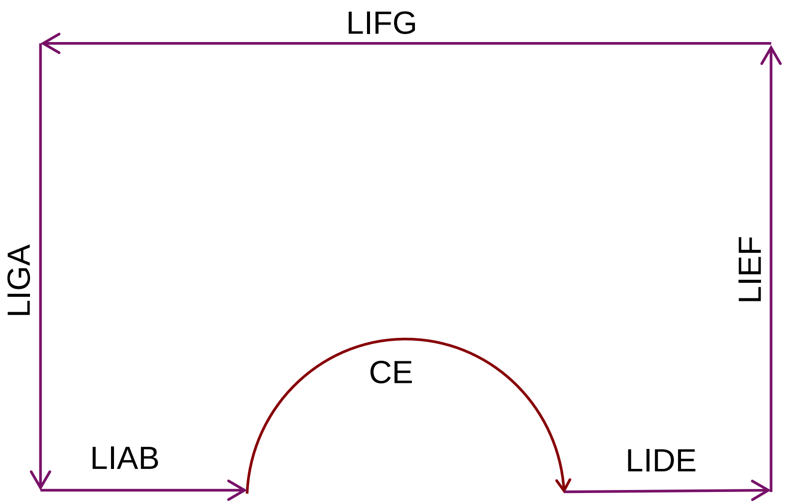
(It’s up to you: see the manual pages)

```
* CLOSED CONTOUR OBTAINED FROM THE ELEMENTARY LINES ASSEMBLY

CO = LIAB ET CE ET LIDE ET LIEF ET LIFG ET LIGA ;
```

new object MAILLAGE

Lines and closed contour

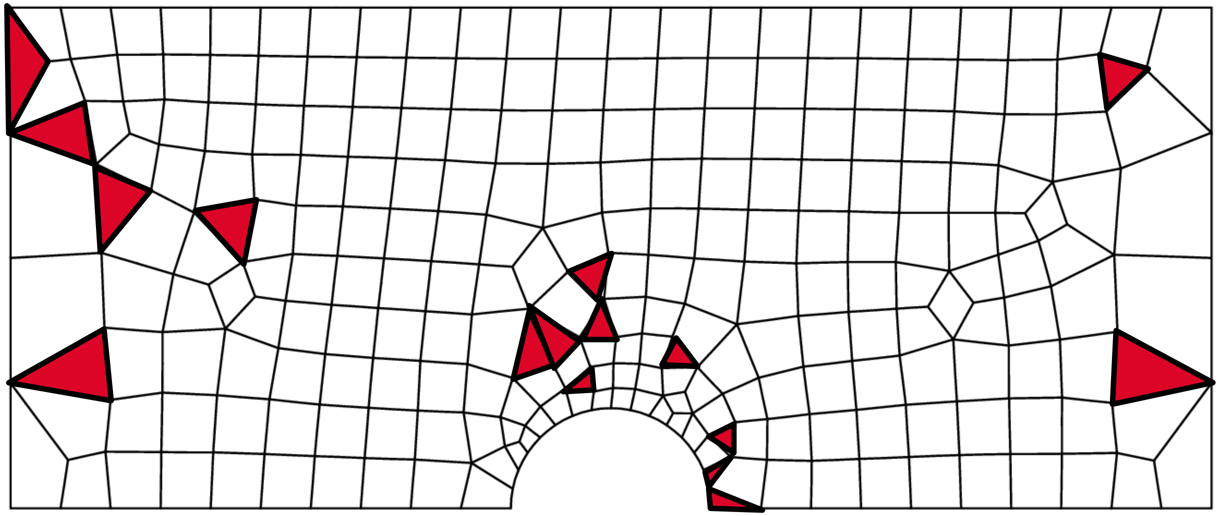
```
* STRAIGHT LINES CONSTRUCTION (DROI or D)

LIAB      = DROI NBASG PA PB ;
LIDE      = DROI NBASD PD PE ;
LIEF      = D    NHAUT PE PF ;
LIFG      = D    NLONG PF PG ;
LIGA      = D    NHAUT PG PA ;

* CONSTRUCTION OF CIRCLES (CERC or C)

CE1       = CERC NQCG PB PCEN PC ;
CE        = C    NQCD CE1 PCEN PD ;
```



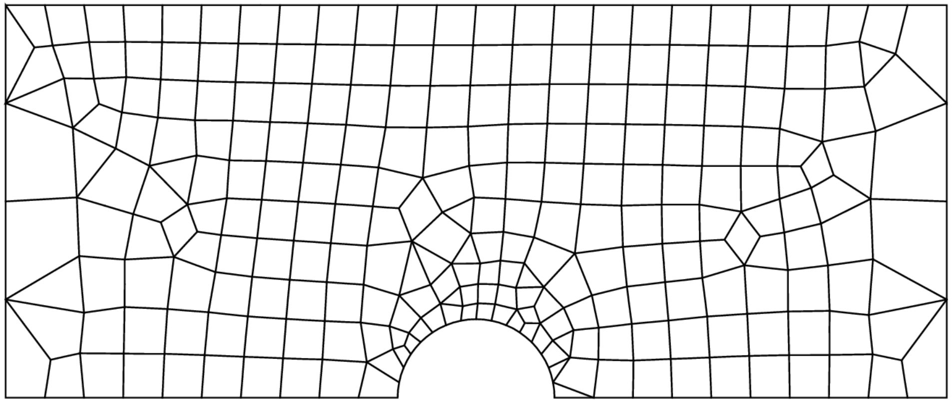

```
* CLOSED CONTOUR OBTAINED FROM THE ELEMENTARY LINES ASSEMBLY

CO = LIAB ET CE ET LIDE ET LIEF ET LIFG ET LIGA ;
```

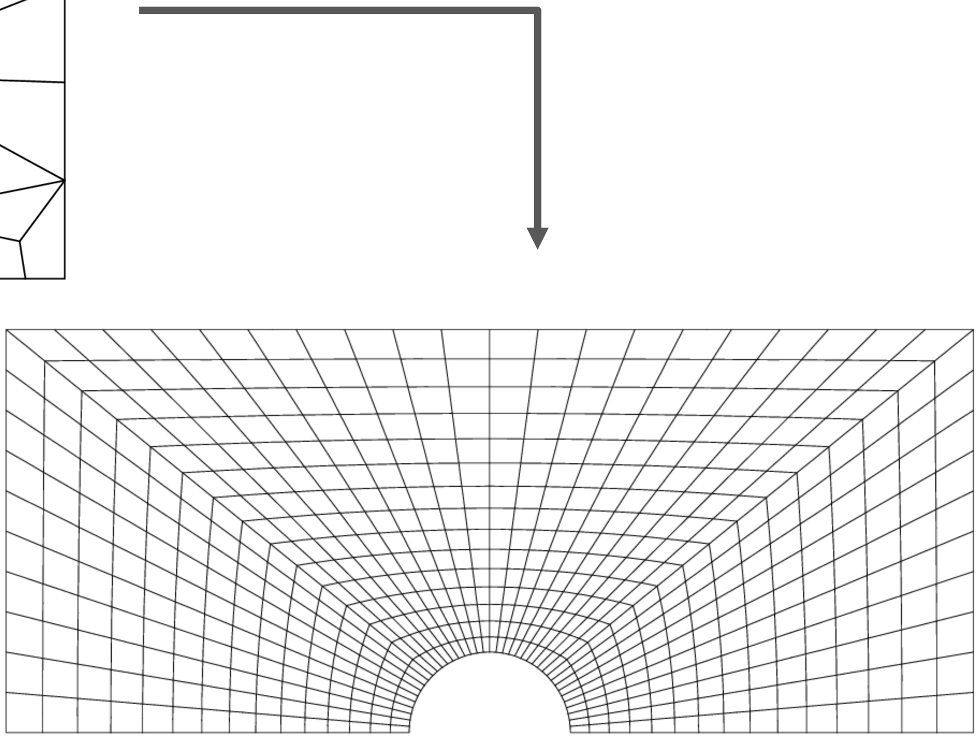
- **Surface meshing (free meshing inside the closed contour)**

```
SU        = SURF CO ;
TRAC SU ;
```



***Objective**: to create a **regular** mesh of the half-plate with a hole*

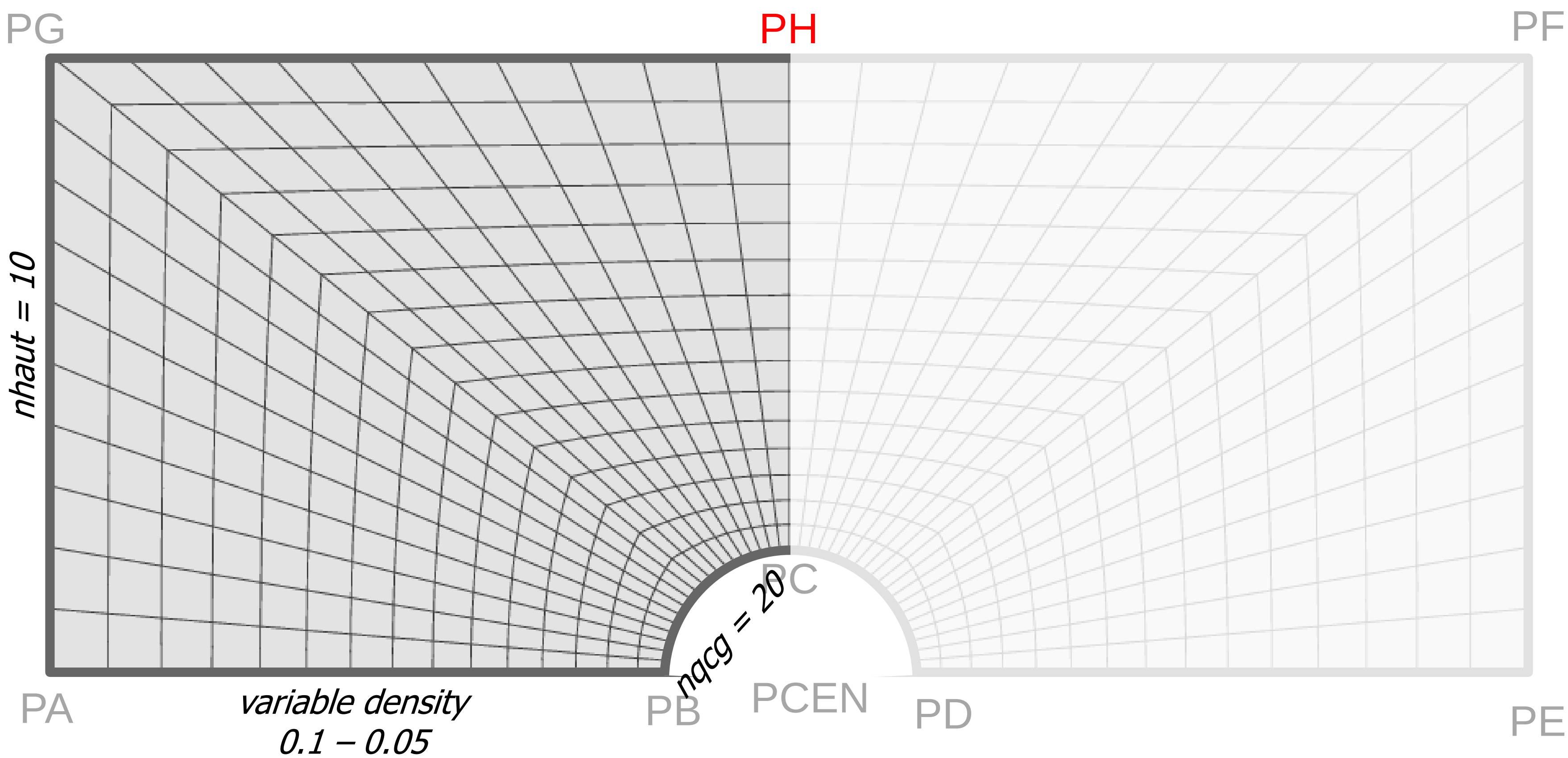




- contains only quadrangle
- controlled elements size
- respect the symmetry of the shape

Objective: *to create a regular mesh of the half-plate with a hole*

regular mesh, symmetry, variable elements size

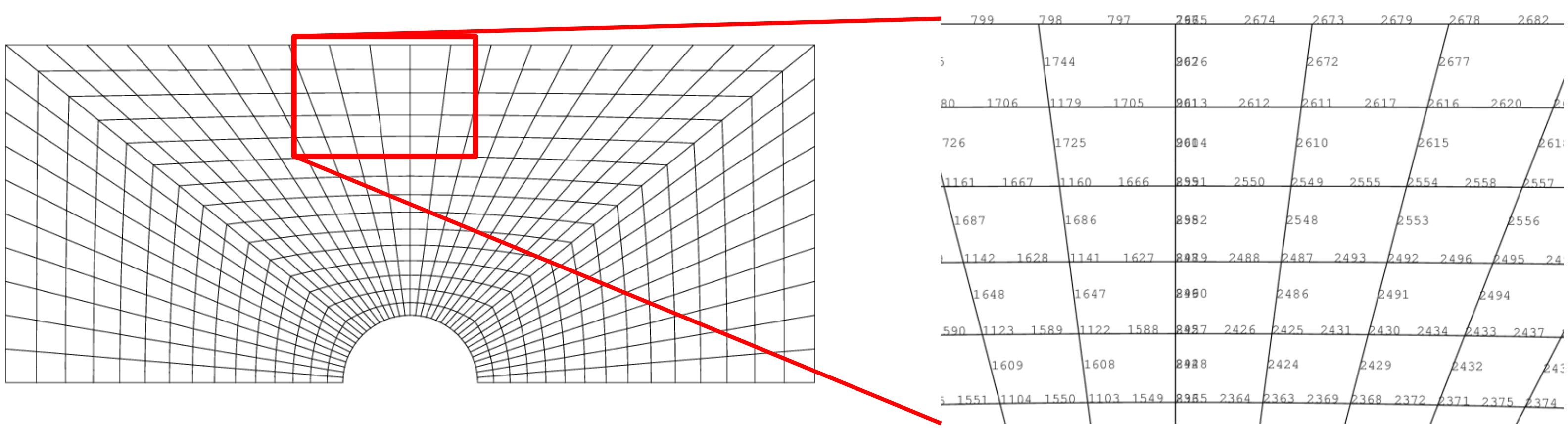


- **Surface meshing (**regular mesh surface relative to two lines**)**

```
NHAUT    = 10 ;
NQCG     = 20 ;
PH       = (locate point PH                         → POIN)
LIHG     = (define line HG                          → DROI)
LIGA     = (define line GA                          → DROI)
CE1      = (define quarter-circle CE1               → CERC)
SU1      = (mesh the surface SU1                    → REGL)
SU2      = (using symmetry mesh the surface SU2     → SYME)
SU       = (group the two meshes SU1 and SU2        → ET)
```

Surface meshing (regular mesh surface relative to two lines)

```
NHAUT     = 10 ;
NQCG      = 20 ;
PH        = (0.5 * LONG) HAUT ;
LIHG      = DROI (NQCG - NHAUT) PH PG ;
LIGA      = DROI NHAUT PG PA ;
CE1       = CERC NQCG PB PCEN PC ;
SU1       = REGL  'DINI' 0.05 'DFIN' 0.1 (INVE CE1) (LIHG ET LIGA) ;
SU2       = SU1 SYME 'DROI' PCEN PH ;
SU        = SU1 ET SU2 ;

TRAC SU ;
```



- Elimination of double nodes, directive **ELIM**

```
ELIM SU 1.E-9 ;
TRAC SU ;
```

➔ all SU nodes distant less than 10^{-9} m are merged in a single one

➔ does not change the element type

➔ directive to use carefully !

- See also **REGE** operator
➔ regeneration of elements with merged nodes

➔ change the element type if necessary

Meshed zones recovery

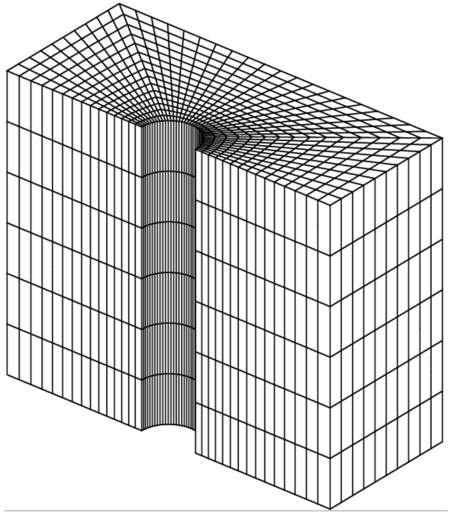
```
* HALF-CIRCLE MESH, UPPER AND LINES MESHES RETRIEVAL WITH 'POIN' ET 'ELEM'

CSU = CONT SU ;                          (Contour of SU mesh recovery)

PCE = SU POIN 'SPHE' PCEN PC 1.E-9 ;     (Points of SU located on sphere
                                         recovery)
CE  = CSU ELEM 'APPU' 'STRI' PCE ;       (Elements containing points of CE
                                         recovery)
PLHAUT = (It’s up to you);               (Points of line PF PG recovery)

LHAUT  = (It’s up to you);               (Elements containing points of
                                         PLHAUT recovery)
PLBAS = (It’s up to you);                (Points of line PA PE recovery)

LBAS  = (It’s up to you);                (Elements containing points of
                                         PLBAS recovery)
```

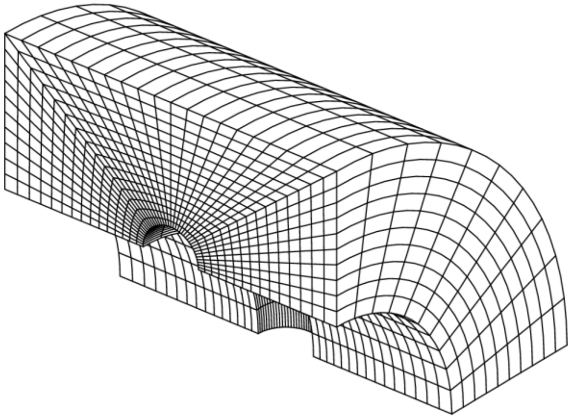
Meshed zones recovery

```
* HALF-CIRCLE MESH, UPPER AND LINES MESHES RETRIEVAL WITH 'POIN' ET 'ELEM'

CSU = CONT SU ;

PCE = SU POIN 'SPHE' PCEN PC 1.E-9 ;

CE  = CSU ELEM 'APPU' 'STRI' PCE ;

PLHAUT = SU POIN 'DROI' PF PG 1.E-9 ;

LHAUT  = CSU ELEM 'APPU' 'STRI' PLHAUT ;

PLBAS = SU POIN 'DROI' PE PCEN 1.E-9 ;

LBAS  = CSU ELEM 'APPU' 'STRI' PLBAS ;
```

Volume meshing (complement)

```
* CHANGE TO DIMENSION 3
OPTI 'DIME' 3 'ELEM' 'CU20' ;

* VOLUME FROM EXTRUSION
VO         = SU VOLU 6  'TRAN' (0. 0. 2.) ;

* VOLUME FROM ROTATION
VO         = SU VOLU 10 'ROTA' 90. (0. -1. 0.) (1. -1. 0.) ;

* DISPLAYING
TRAC VO ;
TRAC 'CACH' VO ;
OPTI 'TRAC' 'OPEN' ;
TRAC 'CACH' VO ;

* BACK TO DIMENSON 2
OPTI 'DIME' 2 ;
```





- **Saving data and end of program**

```
* NAME OF THE OUTPUT FILE
OPTI 'SAUV' 'formation_debutant_1_maillage.sauv' ;

* WRITTING THE FILE
SAUV ;

* END OF THE GIBI PROGRAM
FIN ;
```

- **All of the objects are saved in the output file**
- **The output file is binary (XDR format)**
- **Other formats available (text, …), see the manual page of** `OPTI 'SAUV'` **and** `SAUV`

THERMAL ANALYSIS: REMINDER

Heat equation

$$\rho c_p \frac{\partial T}{\partial t} + \mathrm{div}\left(-\lambda \overrightarrow{\mathrm{grad}}(T)\right) - q = 0 \qquad \text{on } V$$

with:

- T : temperature
- q : volume heat flux source
- λ : thermal conductivity
- ρ : volumetric mass density
- c_p : specific heat capacity
- t : time

Boundary conditions

Prescribed temperatures $\quad T = T_{imp} \qquad$ on ∂V^T

Prescribed surface heat flux

$$\vec{n}\,.\left(\lambda\,\overrightarrow{\mathrm{grad}}(T)\right) = \varphi_{imp} + \underbrace{h\left(T_f - T\right)}_{\text{convection}} + \underbrace{\varepsilon\sigma\left(T_\infty^4 - T^4\right)}_{\text{radiation}} \qquad \text{on } \partial V^{\varphi}$$

THERMAL ANALYSIS: REMINDER

Discrete form (finite elements)

FE discretization: $T(x) = [N(x)]\{T\}$ $\overrightarrow{\text{grad}}(T) = [B(x)]\{T\}$

Weak formulation + EF:

$$[C]\{\dot{T}\} + [K]\{T\} = \{F\}$$

Matrices

$[C] = \int_V \rho c_p [N]^T [N] dV$ capacity matrix (J.K^{-1})

$[K] = \int_V [B]^T [\lambda][B] dV + \int_{\partial V\varphi} h[N]^T [N] dS$ conductivity matrix (W.K^{-1})

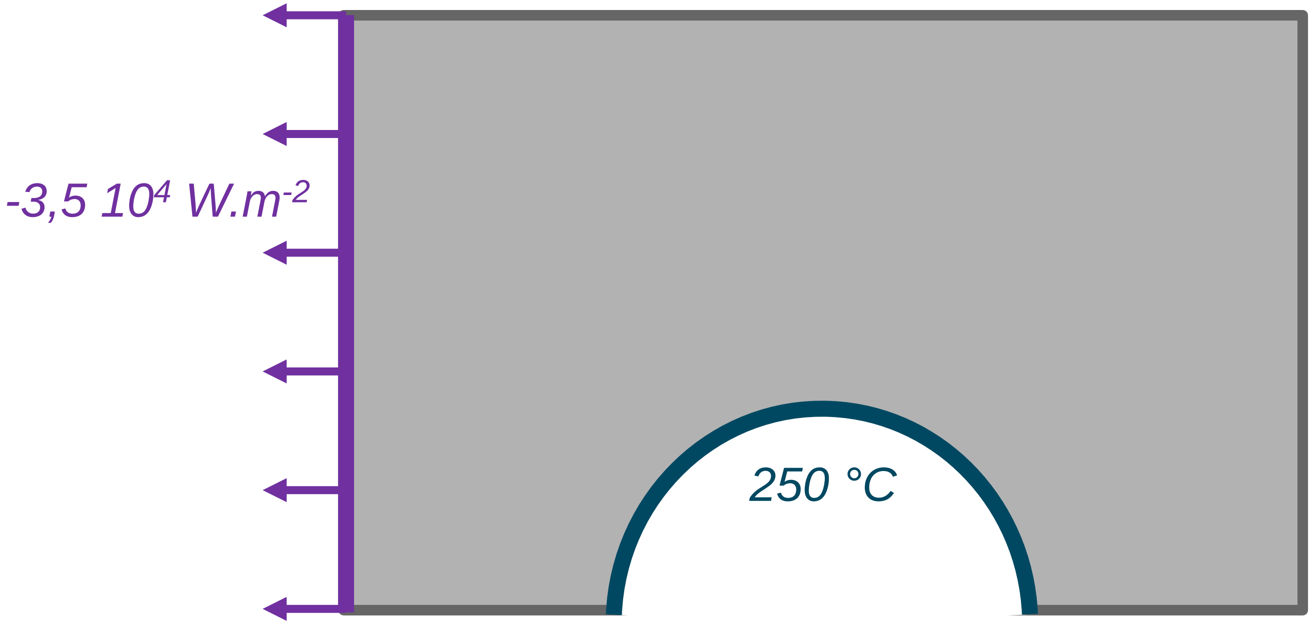
Equivalent nodal flux vectors (W)

$$\{F\} = \int_V [N]^T q \, dV + \int_{\partial V\varphi} [N]^T \left(\varphi_{imp} + hT_f + \varepsilon\sigma(T_\infty^4 - T^4)\right) dS$$

Objective: *stationary thermal calculation with prescribed temperatures (time independent)*

$$\sout{[C]\{\dot{T}\}} + [K]\{T\} = \{F\} \quad \rightarrow \text{Linear system}$$

1. *conductivity matrix calculation* — $[K]$ *(1st member)*
2. *fixed nodal heat fluxes calculation* — $\{F\}$ *(2nd member)*
3. *solving with* `RESO` ➔ *temperature* — $\{T\}$ *(unknown)*



Input data from previous computation (mesh, parameters, …)

```
* NAME OF THE INPUT FILE (RESTITUTION)
OPTI 'REST' 'formation_debutant_1_maillage.sauv' ;

* LOADING INTO MEMORY
REST ;
```

→ **All of the objects are loaded into memory**
→ **They can be called now**

```
* MATERIAL PROPERTIES
CONDUMAT = 210. ;
CAPAMAT  = 900. ;
RHOMAT   = 2700. ;

* INITIAL AND REFERENCE TEMPERATURE
T0       = 25. ;
```

Mathematical formulation

```
* THERMAL MODEL (CONDUCTION) UNIFORM AND STATIONARY MATERIAL PROPERTIES
MOT       = MODE SU 'THERMIQUE' ;
MAT       = MATE MOT 'K' CONDUMAT 'C' CAPAMAT 'RHO' RHOMAT 'TINI' T0 ;

* CONDUCTIVITY MATRIX CALCULATION (FIRST MEMBER)
CON       = COND MOT MAT ;
```

$$[K] = \int_V [B]^T[\lambda][B]dV + \int_{\partial V^{\varphi}} h[N]^T[N]dS$$

new objects MMODEL, MCHAML, RIGIDITE

Boundary conditions

```
* BOUNDARY CONDITIONS: PRESCRIBED TEMPERATURE ON THE HOLE
BLT       = BLOQ CE   'T' ;

* NODAL HEAT FLUX (LOADINGS) FOR CONSTRAINTS
TMAX      = 250. ;
FLT1      = DEPI BLT TMAX ;

* PRESCRIBED HEAT FLUX ON THE LEFT SIDE
FLT2      = FLUX MOT LIGA -3.5E4 ;
```

link

$$\{F\} = \int_V [N]^T q \, dV + \int_{\partial V \varphi} [N]^T \left(\varphi_{imp} + hT_f + \varepsilon\sigma(T_\infty^4 - T^4) \right) dS$$

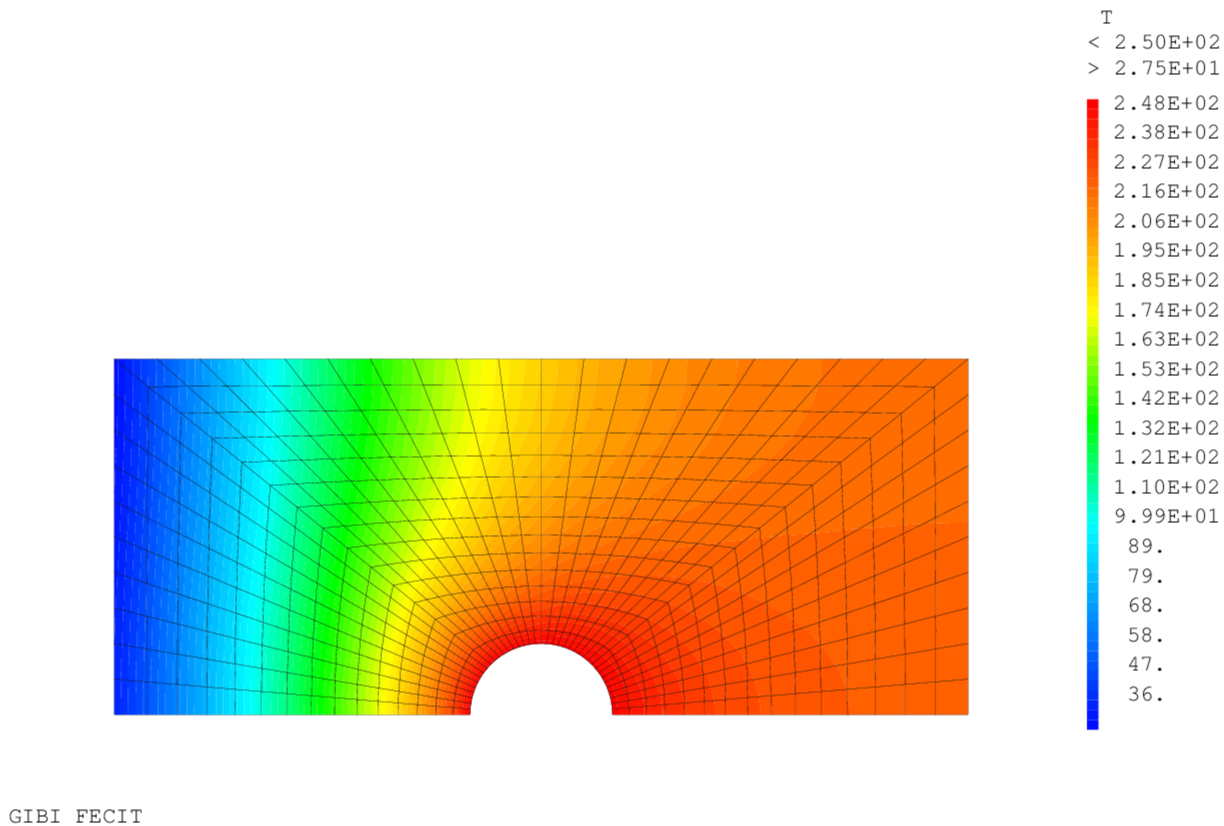
Linear system solving

```
* TEMPERATURE FIELD CALCULATION USING 'RESO'
TCON1     = RESO (CON ET BLT) (FLT1 ET FLT2) ;
```

new object CHPOINT

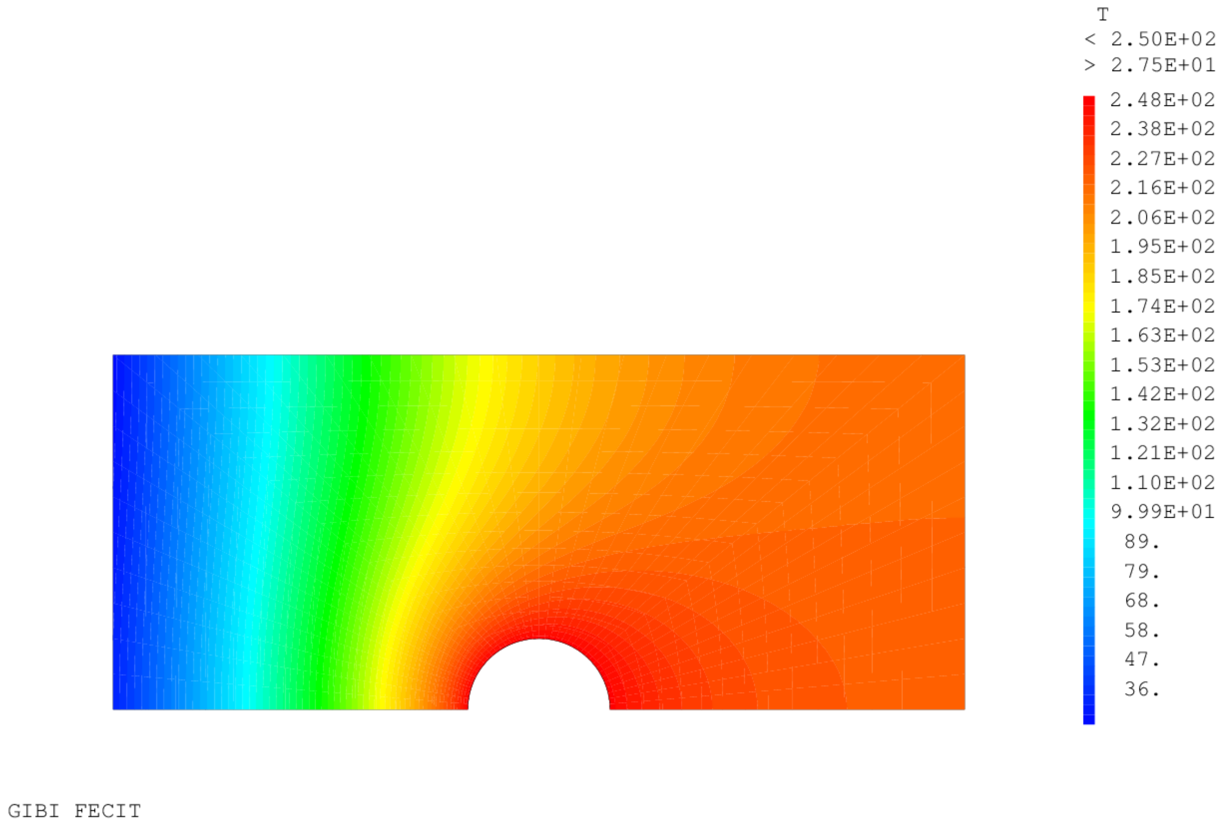
Results display

```
* DISPLAY THE TEMPERATURE FIELD

TRAC TCON1 SU ;
```



Results display

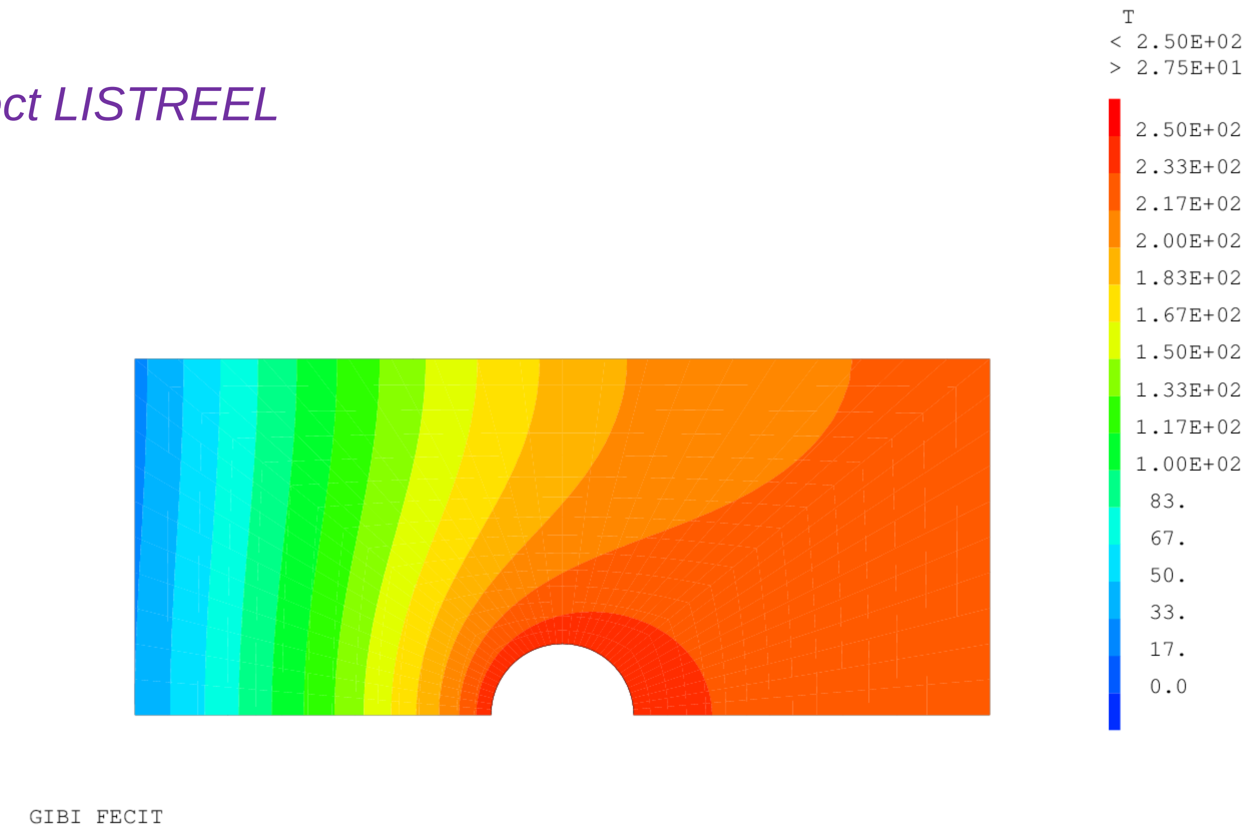
```
* DISPLAY THE TEMPERATURE FIELD

TRAC TCON1 SU CSU ;
```



Results display

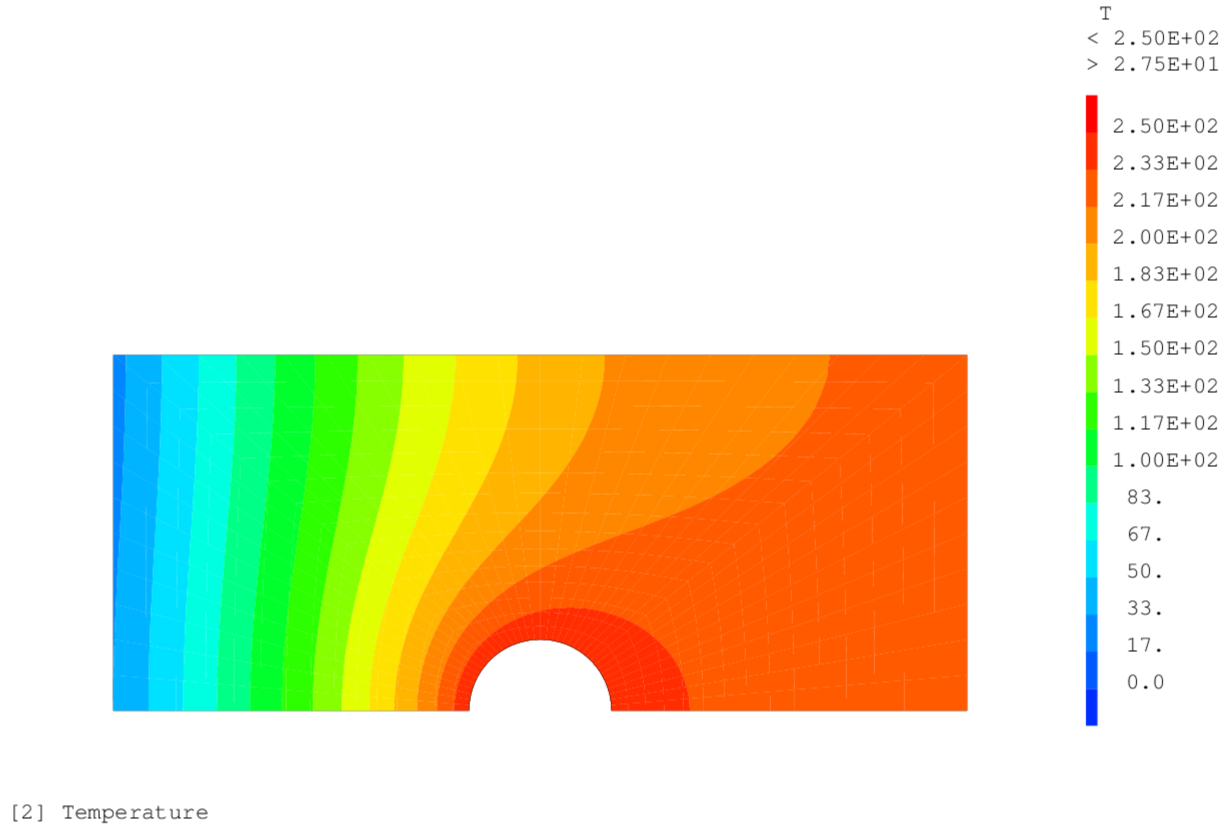
```
* DISPLAY THE TEMPERATURE FIELD
LISO1     = PROG 0. 'PAS' (TMAX / 15.) TMAX ;
TRAC TCON1 SU CSU LISO1 ;
```

new object LISTREEL



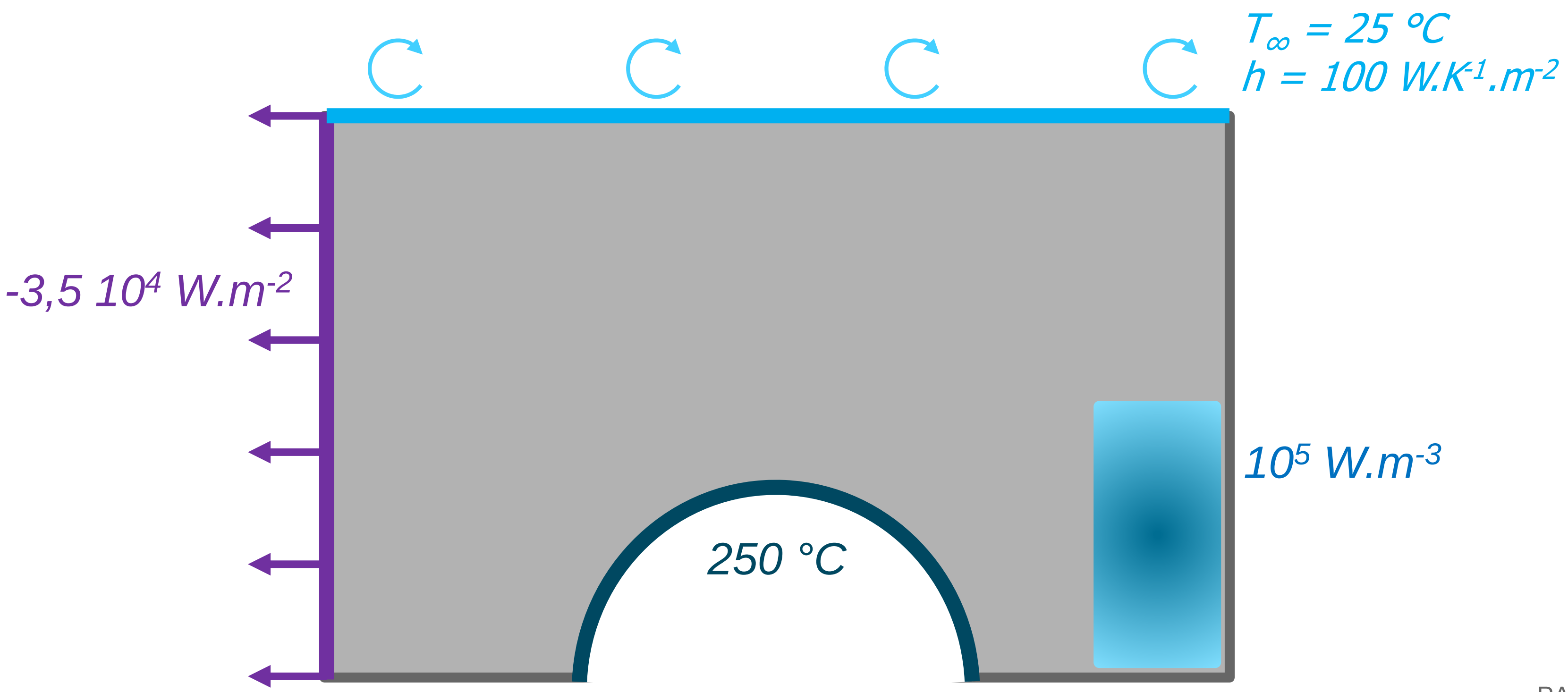
Results display

```
* DISPLAY THE TEMPERATURE FIELD

TRAC TCON1 SU CSU LISO1 'TITR' '[2] Temperature' ;
```



Objective: *previous thermal calculation*
+ convection
+ volume heat source

1. *convection model*
2. *volume source load*



Mathematical formulation

```
* CONVECTION MODEL
MOC       = MODE LHAUT 'THERMIQUE' 'CONVECTION' ;
MAC       = MATE MOC 'H' 100. ;
```

Conductivity matrix (but for convection !)

```
* FIRST MEMBER FOR CONVECTION
CONH      = COND MOC MAC ;
```

$$[K] = \int_V [B]^T[\lambda][B]dV + \int_{\partial V^{\varphi}} h[N]^T[N]dS$$

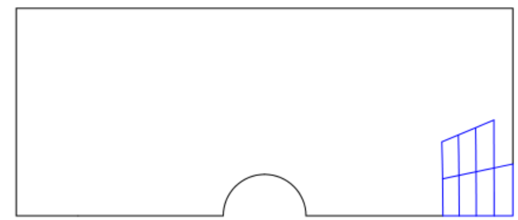
Equivalent nodal heat flux vector (convection)

```
* SECOND MEMBER FOR CONVECTION
CHTC      = MANU 'CHPO' LHAUT 'T' T0 ;
FLH       = CONV MOC MAC CHTC ;
```

$$\{F\} = \int_V [N]^T q \, dV + \int_{\partial V^{\varphi}} [N]^T \left(\varphi_{imp} + hT_f + \varepsilon\sigma(T_\infty^4 - T^4)\right) dS$$

Equivalent nodal heat flux vector (volume source)

```
* SECOND MEMBER FOR THE VOLUME SOURCE
X Y        = COOR SU ;
PT1        = X POIN 'SUPERIEUR' 20.E-1 ;
PT2        = (REDU Y PT1) POIN 'INFERIEUR' 5.E-1 ;
ELSOU      = SU ELEM 'APPUYE' 'STRICTEMENT' PT2 ;
FLS        = SOUR (REDU MOT ELSOU) 1.E5 ELSOU ;
```



$$\{F\} = \int_V [N]^T q \, dV + \int_{\partial V \varphi} [N]^T \left(\varphi_{imp} + hT_f + \varepsilon\sigma(T_\infty^4 - T^4)\right) dS$$

Linear system solving

```
* TEMPERATURE FIELD CALCULATION USING 'RESO'
TCON2      = RESO (CON ET CONH ET BLT) (FLT1 ET FLT2 ET FLH ET FLS) ;


* DISPLAY THE TEMPERATURE FIELD
TRAC TCON2 SU CSU LISO1 ;
```

NOTE: THE NODAL FIELD (CHPOINT)

The CHPOINT type object

Values are **located on POINTS (nodes)**
Examples:

- scalar temperature field
- vector displacement field (3 components)
- vector coordinates fields
- second member of a linear problem $[K]\{x\} = \{F\}$, for instance:
  - nodal forces
  - nodal heat flux
- and others…

Some characteristics:

- only one value per node
- do not depends on the mesh, only on nodes!
- when plotted, the field is continuous on the mesh

NOTE: THE ELEMENT FIELD (MCHAML)

The MCHAML type object

Values are **<u>located into ELEMENTS</u>** of a mesh
Examples:

- material properties
- stress, strains fields
- internal variables field
- gradient of a CHPOINT object
- and others…

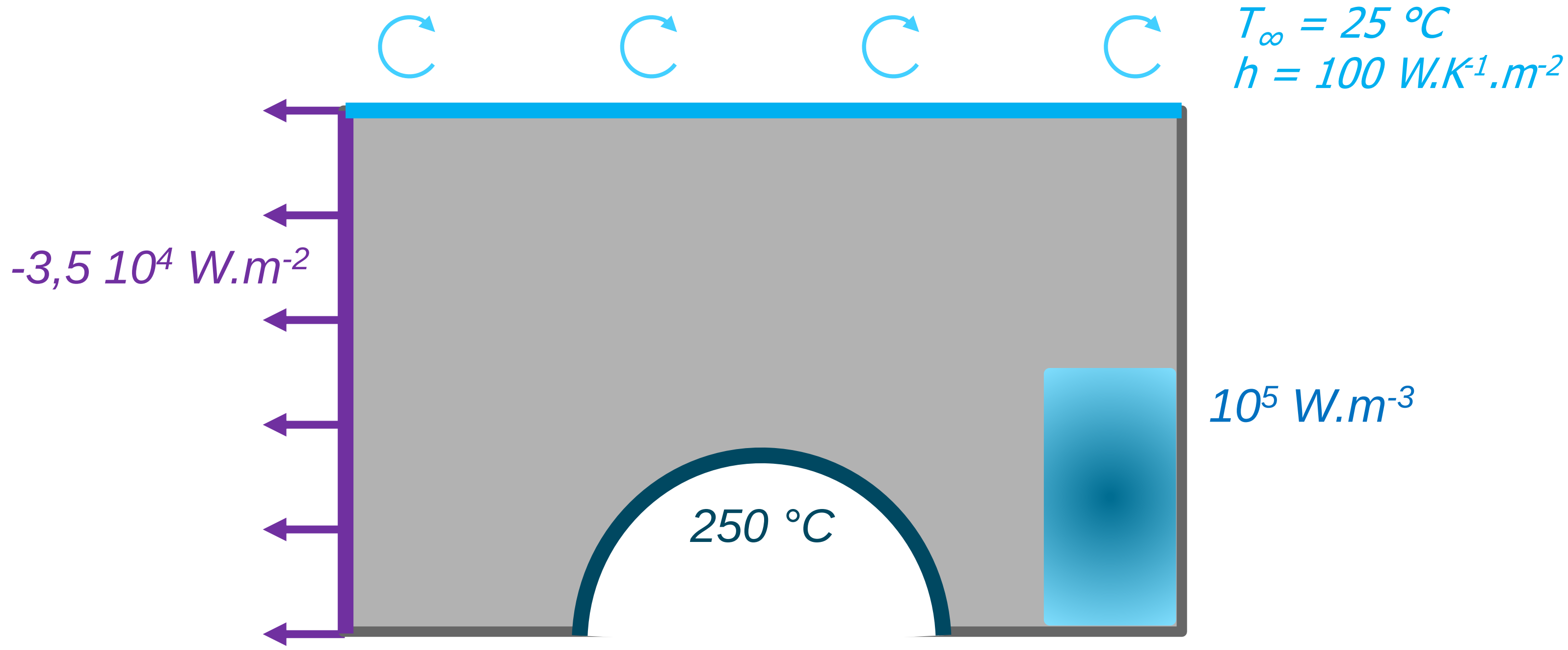
Some characteristics:

- several support points available:
    - integration points for stresses
    - integration points for stiffness
    - integration points for mass
    - center of gravity
    - nodes
- interpolation functions depends on the model
- non continuous between elements

Objective: *previous thermal calculation*
+ transient (with an initial temperature = 25 °C)

$$[C]\{\dot{T}\} + [K]\{T\} = \{F\}$$

1. *time description of loading*
2. *initial conditions*
3. *using the PASAPAS solving procedure*



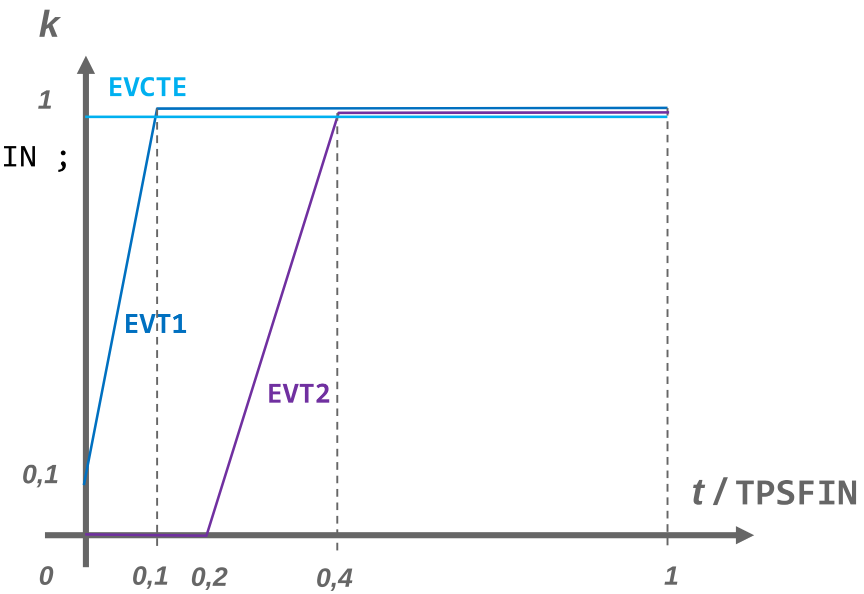
Initial conditions

```
* FINAL INSTANT
TPSFIN    = 5.E4 ;
```

Loading definition (BC as function of space and time)

```
* TEMPERATURE LOAD
LIST1     = PROG 0. (0.1 * TPSFIN) TPSFIN ;
LIST2     = PROG (T0 / TMAX) 1. 1. ;
EVT1      = EVOL 'MANU' LIST1 LIST2 ;
CHATIMP   = CHAR 'TIMP' FLT1 EVT1 ;

* HEAT FLUX LOAD
LIST3     = PROG 0. (0.2 * TPSFIN) (0.4 * TPSFIN) TPSFIN ;
LIST4     = PROG 0. 0. 1. 1. ;
EVT2      = EVOL 'MANU' LIST3 LIST4 ;
CHAFIMP   = CHAR 'Q'     FLT2 EVT2 ;

* CONVECTION TEMPERATURE LOAD
LIST5     = PROG 0. TPSFIN ;
LIST6     = PROG 1. 1. ;
EVCTE     = EVOL 'MANU' LIST5 LIST6 ;
CHACONV   = CHAR 'TECO' CHTC EVCTE ;

* HEAT SOURCE LOAD
CHASOUR   = CHAR 'Q'     FLS EVT2 ;


CHAT       = CHATIMP ET CHAFIMP ET CHACONV ET CHASOUR ;
```

Loading $\{F\}(t) = k(t).\{F\}$



new objects EVOLUTIOn and CHARGEMEnt

Construction of TAB1: table for procedure PASAPAS

```
* TABLE OF ARGUMENTS NECESSARY AS ENTRIES FOR THE PASAPAS PROCEDURE

TAB1                              = TABL ;
TAB1 . 'MODELE'                   = MOT ET MOC ;
TAB1 . 'CARACTERISTIQUES'         = MAT ET MAC ;
TAB1 . 'BLOCAGES_THERMIQUES' = BLT ;
TAB1 . 'CHARGEMENT'               = CHT ;
TAB1 . 'TEMPS_CALCULES'           = PROG 0. 'PAS' (0.02 * TPSFIN) TPSFIN ;
```

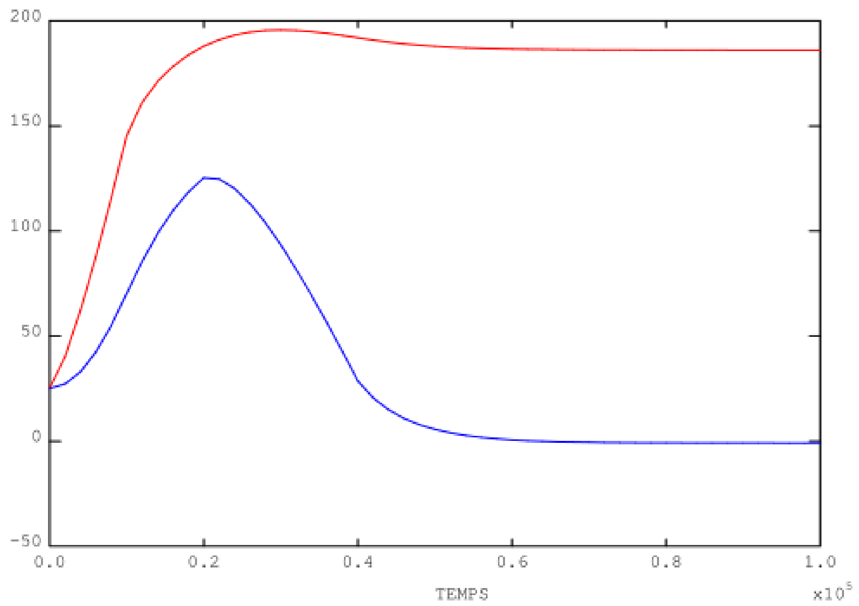
Solving with PASAPAS procedure

```
* PASAPAS CALL
PASAPAS TAB1 ;
```

new object TABLE

Post processing: evolution curves, field plot

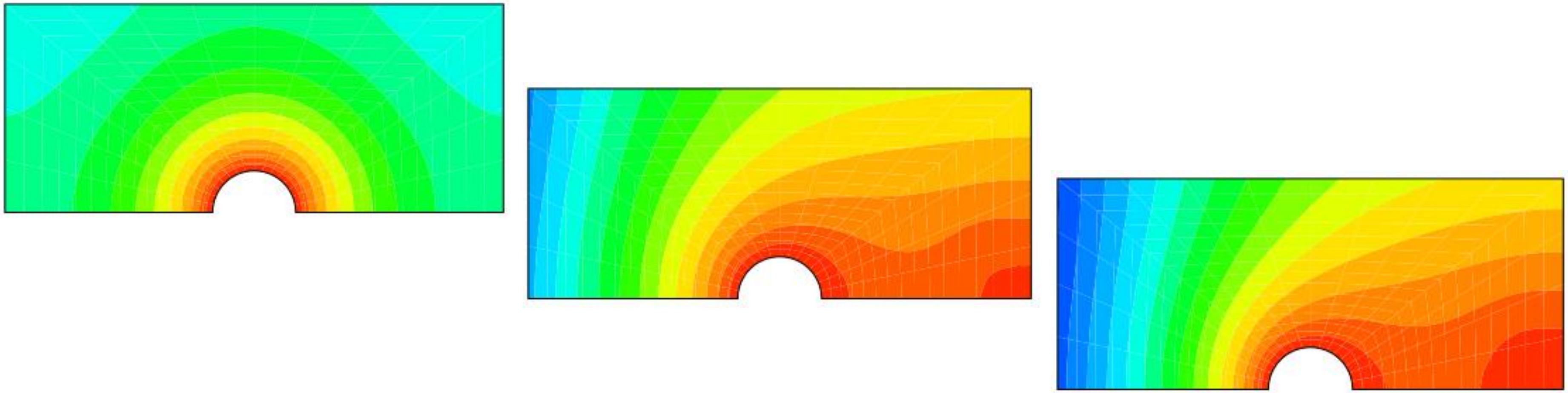
```
* VARIATION OF TEMPERATURE AT TWO POINTS AS A FUNCTION OF TIME
* IN ORDER TO VERIFY STEADY STATE IS REACHED
PMIL      = SU POIN 'PROC' ((0.5 * LONG) (0.5 * HAUT)) ;
EV1       = EVOL 'ROUG' 'TEMP' TAB1 'TEMPERATURES' 'T' PMIL ;
EV2       = EVOL 'BLEU' 'TEMP' TAB1 'TEMPERATURES' 'T' PG ;

DESS (EV1 ET EV2) ;
```



Post processing: iterative loop for plotting

```
* LOOP ON EACH TIME STEP AND PLOT OF TEMPERATURE FIELD
N1        = DIME (TAB1 . 'TEMPERATURES') ;
REPE B1 N1 ;
  T_I       = TAB1 . 'TEMPERATURES' . (&B1 - 1) ;
  TPS_I     = TAB1 . 'TEMPS' . (&B1 - 1) ;
  PRC_I     = ENTI (100. * TPS_I / TPSFIN) ;
  MOT_I     = CHAI '[3] Temperatures at time ' TPS_I '  (' PRC_I ' %)' ;
  TRAC T_I SU CSU 'TITR' MOT_I LISO1 ;
FIN B1 ;
```

Loop counter
1, 2, 3, … , N1



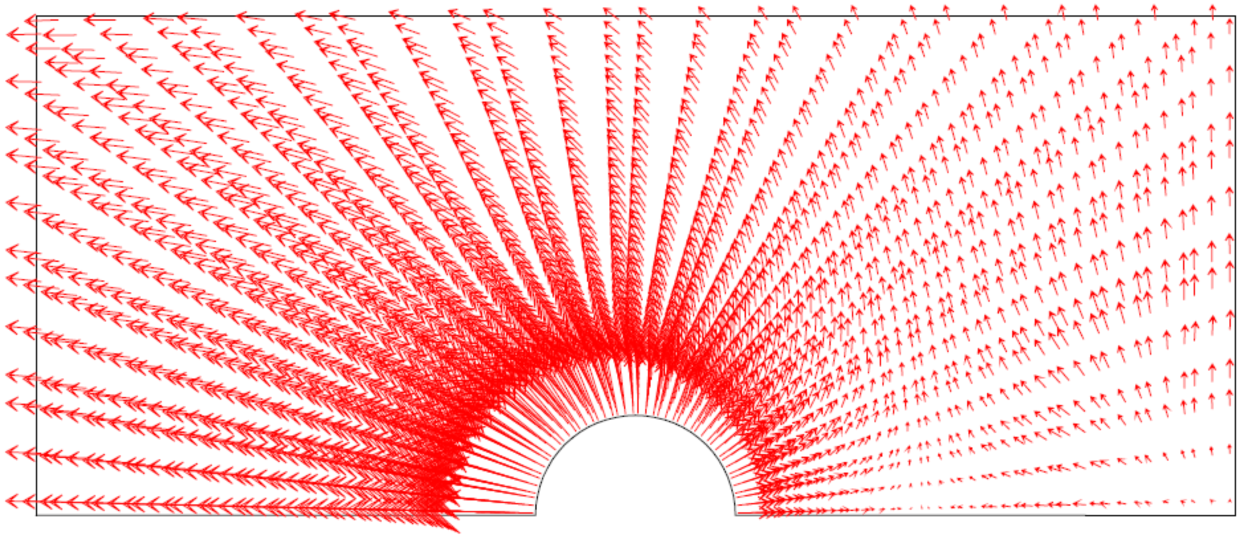
Procedure writing (heat flux vector calculation)

```
* HEAT FLUX VECTOR ==> PROCEDURE WRITING
DEBP VECFLU  CHP1*'CHPOINT'  MOD1*'MMODEL'  MAT1*'MCHAML' ;
* GRADIENT CALCULATION AND CHANGE OF ITS TYPE
  G1        = GRAD CHP1 MOD1 ;
  G2        = CHAN 'TYPE' G1 'CARACTERISTIQUES' ;
* MULTIPLICATION OF THE ELEMENT FIELDS
  Q         = MAT1 * G2 (MOTS 'K' 'K') (MOTS 'T,X' 'T,Y') (MOTS 'QX' 'QY') ;
  Q         = -1. * Q ;
* VECTOR OBJECT CREATION
  VEC1      = VECT Q MOD1 (MOTS 'QX' 'QY') 2.E-6 ;
FINP VEC1 ;
```

new objects PROCEDURe, LISTMOTS, VECTEUR

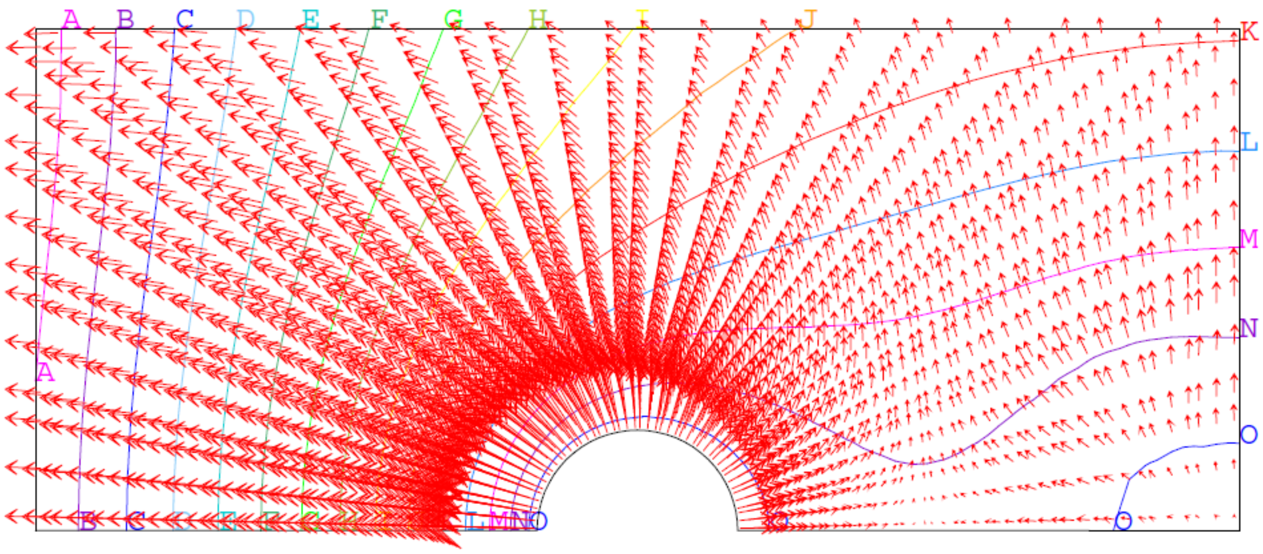
Post processing: heat flux vector plot

```
* LOOP ON EACH TIME STEP AND PLOT OF HEAT FLUX VECTOR
REPE B1 N1 ;
  T_I      = TAB1 . 'TEMPERATURES' . (&B1 - 1) ;
  VF_I     = VECFLU T_I MOT MAT ;
  TRAC VF_I CSU ;
FIN B1 ;
```



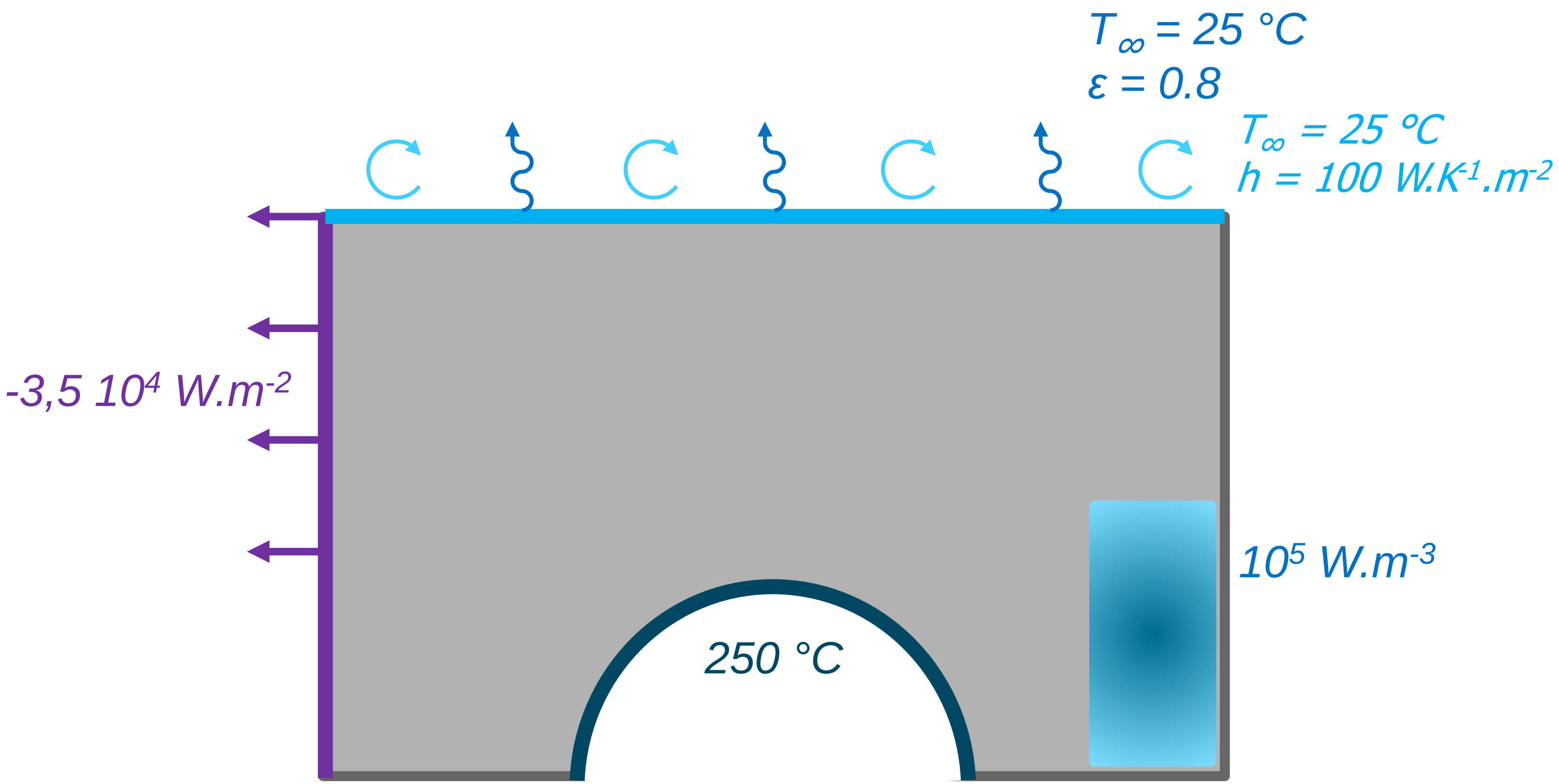
- **Post processing: plot of heat flux vector and temperature isovalue lines**

```
* FLUX VECTOR AND TEMPERATURE FIELD AS ISOVALUE LINES
OPTI 'ISOV' 'LIGN' ;
TRAC VF_I T_I SU CSU 15 ;
OPTI 'ISOV' 'SURF' ;
```



Objective: *previous thermal calculations*
+ radiation

1. *addition of a model and loading for radiation*



Radiation heat transfer model

```
* RADIATION MODEL ON THE UPPER LINE
MOR      = MODE LHAUT 'THERMIQUE' 'RAYONNEMENT' 'INFINI' ;
MAR      = MATE MOR 'EMIS' 0.8 ;
```

Loading with the surroundings temperature

```
* RADIATION LOADING IS THE SURROUNDINGS TEMPERATURE
CHTR     = MANU 'CHPO' LHAUT 1 'T' T0 ;
CHARAYE  = CHAR 'TERA' CHTR EVCTE ;
CHAT     = CHAT ET CHARAY ;
```

PASAPAS table building

```
* RE-DEFINITION OF TABLE TAB1 FOR PASAPAS PROCEDURE
TAB1                                = TABL ;
TAB1 . 'MODELE'                     = (It's up to you) ;
TAB1 . 'CARACTERISTIQUES'           = (It's up to you) ;
TAB1 . 'BLOCAGES_THERMIQUES'        = (It's up to you) ;
TAB1 . 'CHARGEMENT'                 = (It's up to you) ;
TAB1 . 'TEMPS_CALCULES'             = (It's up to you) ;
     Other indexes ?? (It's up to you)
```

Solving with PASAPAS procedure

```
* PASAPAS CALL
(It's up to you)
```

Post processing

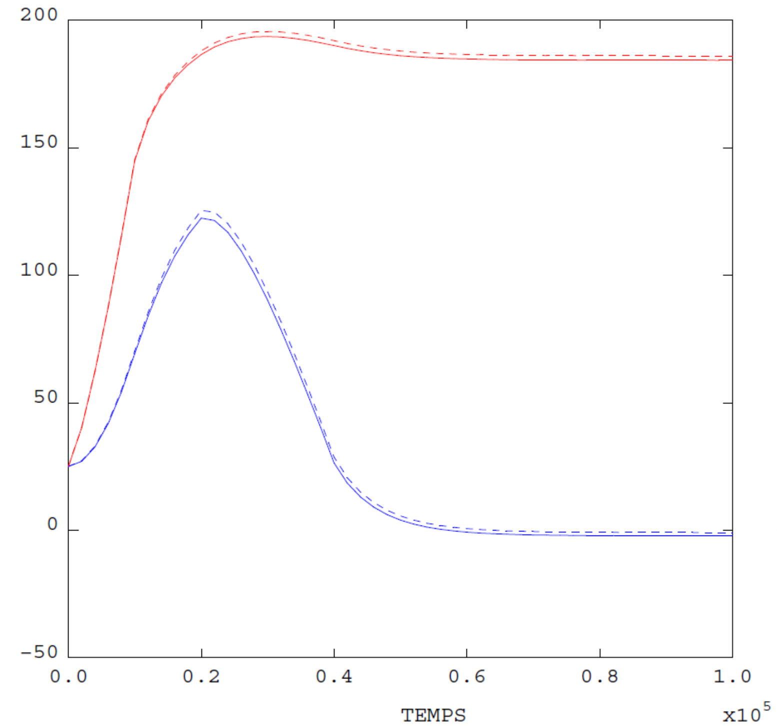
```
* RE-DEFINITION OF TABLE TAB1 FOR PASAPAS PROCEDURE
TAB1                          = TABL ;
TAB1 . 'MODELE'               = MOT ET MOC ET MOR ;
TAB1 . 'CARACTERISTIQUES'     = MAT ET MAC ET MAR ;
TAB1 . 'BLOCAGES_THERMIQUES' = BLT ;
TAB1 . 'CHARGEMENT'           = CHAT ;
TAB1 . 'TEMPS_CALCULES'       = PROG 0. 'PAS' (0.02 * TPSFIN) TPSFIN ;
TAB1 . 'CELSIUS'              = VRAI ;
```

Solving with PASAPAS procedure

```
* PASAPAS CALL
PASAPAS TAB1 ;
```

Post processing

```
* TIME EVOLUTION OF TEMPERATURE ON 2 POINTS
EV11     = EVOL 'ROUG' 'TEMP' TAB1 'TEMPERATURES' 'T' PMIL ;
EV22     = EVOL 'BLEU' 'TEMP' TAB1 'TEMPERATURES' 'T' PG ;
TL       = TABL ;
TL . 1   = 'TIRR' ;
TL . 2   = 'TIRR' ;
TL . 'TITRE' = TABL ;
TL . 'TITRE' . 1 = 'PMIL' ;
TL . 'TITRE' . 2 = 'PG  ' ;
TL . 'TITRE' . 3 = 'PMIL with rad' ;
TL . 'TITRE' . 4 = 'PG   with rad' ;
DESS (EV1 ET EV2 ET EV11 ET EV22) 'LEGE' TL ;
```



Saving data and end of program

```
OPTI 'SAUV' 'formation_debutant_2_thermique.sauv' ;
SAUV ;

FIN ;
```

MECHANICS: REMINDER

Equilibrium equation (statics)

$$\mathrm{div}(\boldsymbol{\sigma}) + \vec{f} = \vec{0} \quad \text{on } V$$

Boundary conditions

Prescribed displacements: $\vec{u} = \vec{d}$ on ∂V^d

Prescribed surface forces: $\boldsymbol{\sigma}.\vec{n} = \vec{t}$ on ∂V^t

with:

- $\vec{u}$: displacement vector
- $\boldsymbol{\sigma}$: stress tensor
- $\vec{f}$: volume forces vector
- $\vec{n}$: normal vector to a surface

MECHANICS: REMINDER

- **FE discretization**

$$\vec{u}(x) = [N(x)]\{u\}$$

- **Weak formulation of equilibrium + FE discretization**

$$\{F\} - \int_V [B]^T\{\sigma\}dV = \{0\}$$

$$\underbrace{\int_{\partial V^t} [N]^T\{t\}\,dS}_{\{F\}^S} + \underbrace{\int_{\partial V^d} [N]^T\{\boldsymbol{\sigma}.n\}dS}_{\{F\}^R} + \underbrace{\int_V [N]^T\{f\}\,dV}_{\{F\}^V} - \underbrace{\int_V [B]^T\{\sigma\}dV}_{[B]\{\sigma\}} = \{0\}$$

Equivalent nodal forces vector (N) to:

$\{F\}^S$ prescribed surface forces $\vec{t}$

$\{F\}^R$ surface forces in reaction to prescribed displacements $\vec{d}$

$\{F\}^V$ prescribed volume forces f

$[B]\{\sigma\}$ internal volume forces

Matrices

$[N]$ matrix of shape functions (.)

$[B]$ matrix of derivative of shape functions (m^{-1})

MECHANICS: REMINDER

- **FE discretization**

$$\vec{u}(x) = [N(x)]\{u\}$$

- **Weak formulation of equilibrium + FE discretization**

$$\{F\} - \int_V [B]^T\{\sigma\}dV = \{0\}$$

- **Small strain hypothesis** $\boldsymbol{\varepsilon}(x) = [B(x)]\{u\}$ **+ elasticity**

$$\int_V [B]^T\{\sigma\}dV = \underbrace{\int_V [B]^T[C][B]dV}_{[K]}.\{u\}$$

$$[K]\{u\} = \{F\}$$

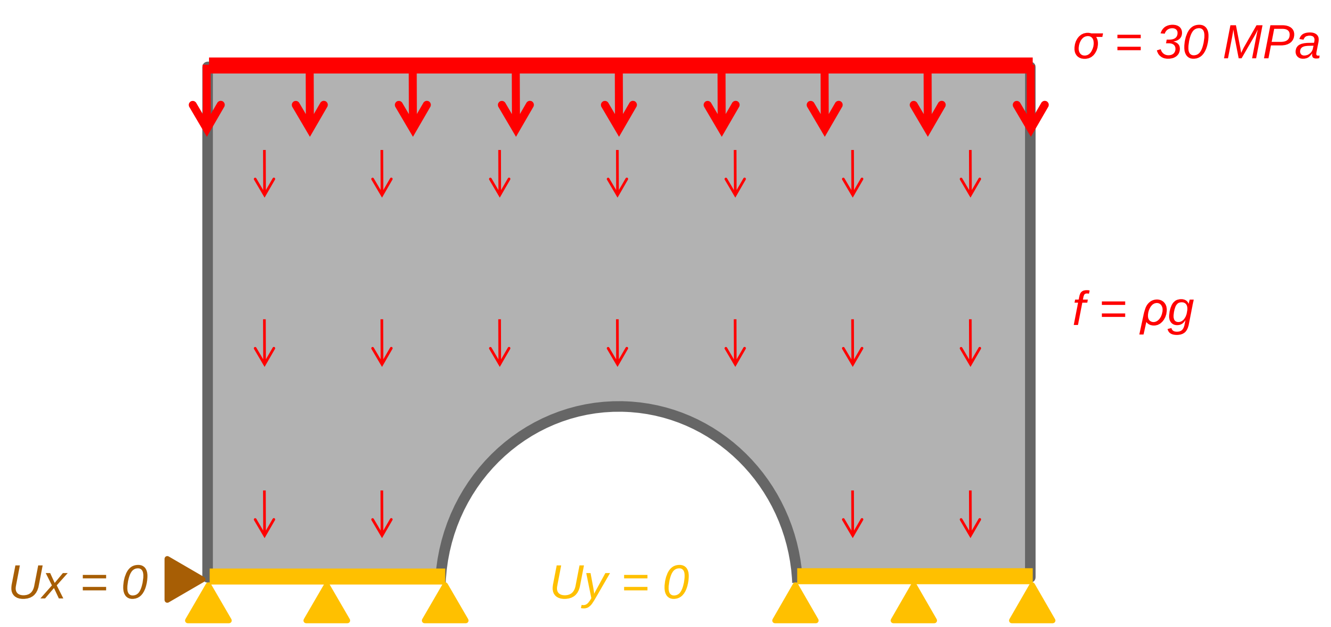
Matrices

$[K]$ stiffness matrix (N. m^{-1})

Objective: elastic calculations
with prescribed displacements
and surface and volume forces

$$[K]\{u\} = \{F\}$$ → *Linear system*

1. *stiffness matrix calculation* — $[K]$ *(1st member)*
2. *nodal forces calculation* — $\{F\}$ *(2nd member)*
3. *solving with* `RESO` ➔ displacements — $\{u\}$ *(unknown)*



Input data from previous computation (mesh, parameters, …)

```
OPTI 'REST' 'formation_debutant_2_thermique.sauv' ;
REST ;

* MECHANICAL MATERIAL PARAMETERS
YOUNGMAT = 30.E9 ;
NUMAT    = 0.2 ;
ALPHAMAT = 12.E-6 ;
SIGYMAT  = 120.E6 ;
```

Plane strain hypothesis

```
OPTI 'MODE' 'PLAN' 'DEFO' ;
```

Linear elastic isotropic mechanical model

```
MOM1     = MODE SU 'MECANIQUE' 'ELASTIQUE' ;
MAM1     = MATE MOM1 'YOUN' YOUNGMAT 'NU' NUMAT 'ALPH' ALPHAMAT 'TREF' T0 'TALP' T0 ;
```

- **Stiffness matrix (1st member)**

```
RI        = RIGI MOM1 MAM1 ;
```

$$[K] = \int_V [B]^T [C][B] dV$$

- **Constraints on displacements (1st member)**

```
BLMX      = BLOQ PA   'UX' ;          (point A locked along x)
BLMY      = BLOQ (It's up to you);    (lower line locked along y)
RITOT     = (It's up to you);         (assembly of stiffness)
```

- **Equivalent nodal forces to the pressure (2nd member)**

```
FS        = PRES (It's up to you);
```

- **Equivalent nodal forces to the volume forces (2nd member)**

```
FV        = CNEQ (It's up to you);
```

- **Linear system solving**

```
U5        = (It's up to you)
```

Stiffness matrix (1st member)

```
RI         = RIGI MOM1 MAM1 ;
```

$$[K] = \int_V [B]^T [C][B] dV$$

Constraints on displacements (1st member)

```
BLMX       = BLOQ PA   'UX' ;         (point A locked along x)
BLMY       = BLOQ LBAS 'UY' ;
RITOT      = RI ET BLMX ET BLMY ;
```

Equivalent nodal forces to the pressure (2nd member)

```
FS         = PRES 'MASS' MOM1 LHAUT 30.E6 ;
```

$$\{F\}^S = \int_{\partial V^t} [N]^T \{t\}\, dS$$

Equivalent nodal forces to the volume forces (2nd member)

```
ROG        = MANU 'CHPO' SU 2 'FX' 0. 'FY' (-9.81 * RHOMAT) ;
FV         = CNEQ MOM1 ROG ;
```

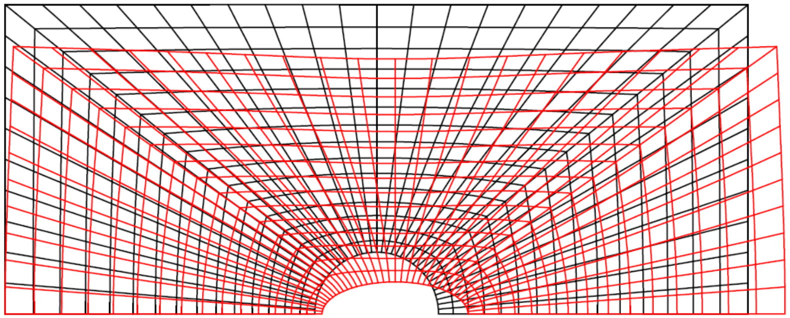
$$\{F\}^V = \int_V [N]^T \{f\}\, dV$$

Linear system solving

```
U5         = RESO RITOT (FS ET FV) ;
```

Post processing: strains, stresses, deformed mesh

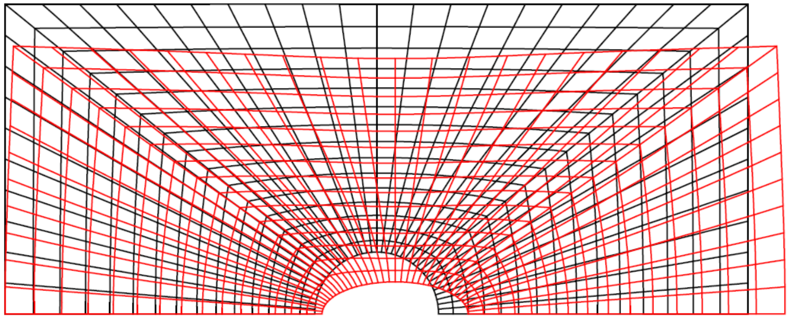
```
* DEFORMED MESH PLOT
DEF_5    = DEFO SU U5 150. 'ROUG' ;
DEF_INI  = DEFO SU U5 0. ;
TRAC (DEF_INI ET DEF_5) ;

* PLOT OF THE DEFORMED MESH BORDER
(It's up to you)
```

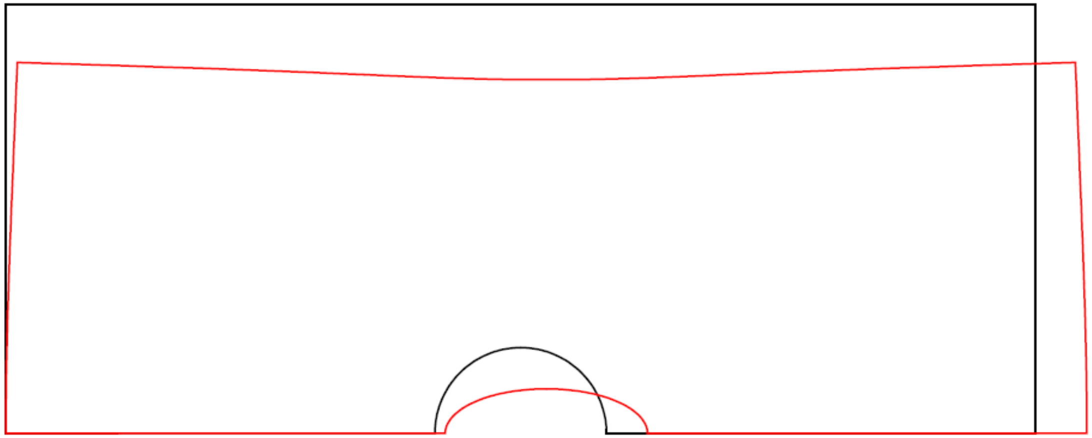


new object DEFORMEE

Post processing: strains, stresses, deformed mesh

```
* DEFORMED MESH PLOT
DEF_5     = DEFO SU U5 150. 'ROUG' ;
DEF_INI  = DEFO SU U5 0. ;
TRAC (DEF_INI ET DEF_5) ;

* PLOT OF THE DEFORMED MESH BORDER
DEF_5C    = DEFO CSU U5 150. 'ROUG' ;
DEF_INIC = DEFO CSU U5 0. ;
TRAC (DEF_INIC ET DEF_5C) ;
```





new object DEFORMEE

Post processing: strains, stresses, deformed mesh

With two statements:

```
* STRAINS CALCULATION
DEF       = EPSI U5 MOM1 'LINE' ;
```

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\mathbf{grad}(u) + \mathbf{grad}^{\boldsymbol{T}}(u)\right)$$

$$\{\varepsilon\} = [B]\{u\}$$

```
* STRESSES CALCULATION FROM STRAINS
SIG       = ELAS DEF MOM1 MAM1 ;
```

$$\{\sigma\} = [C]\{\varepsilon\}$$

With one statement:

```
* STRESSES CALCULATION FROM DISPLACEMENTS
SIG       = SIGM U5 MOM1 MAM1 'LINE' ;
```

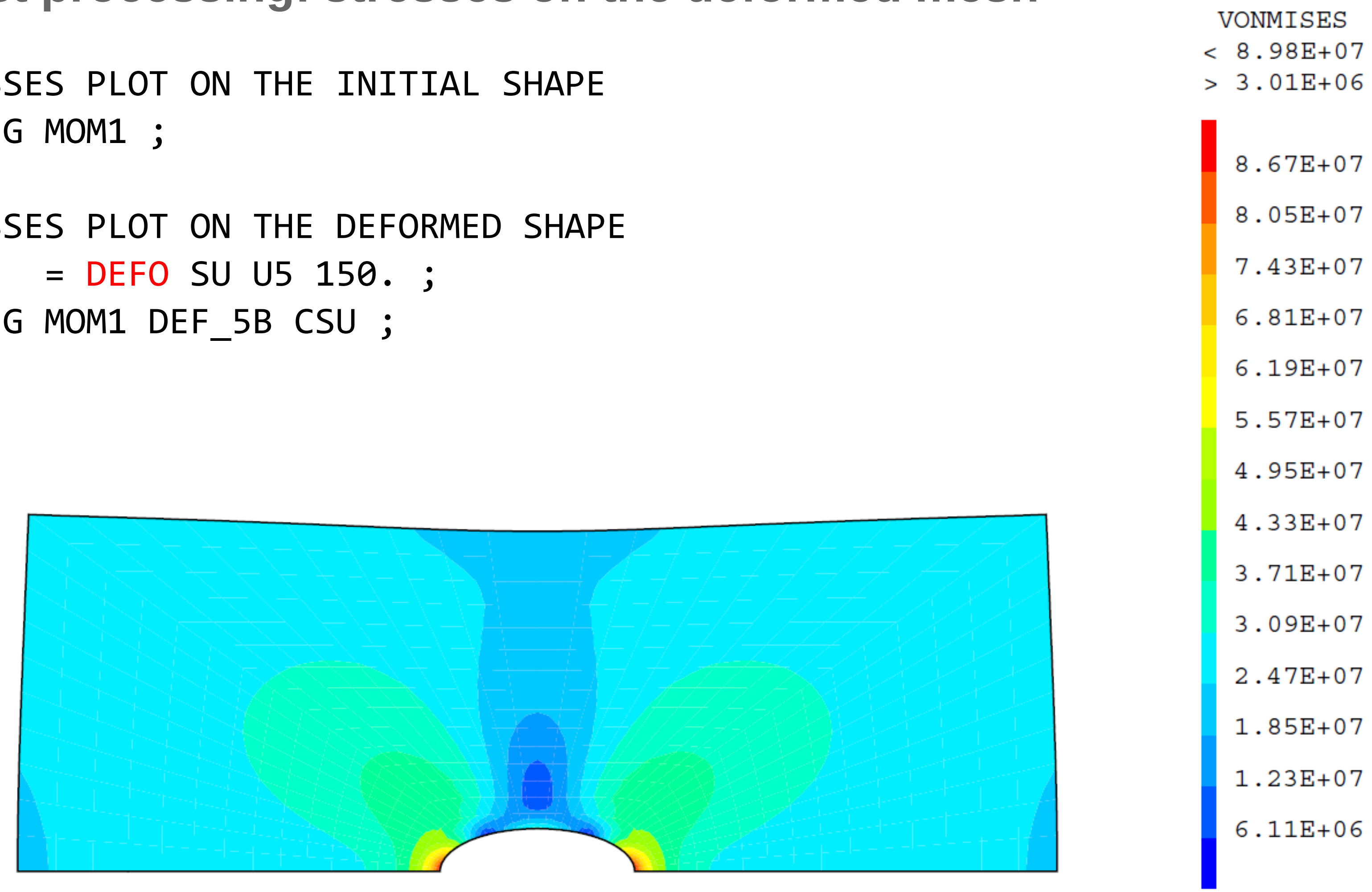
$$\{\sigma\} = [C][B]\{u\}$$

Post processing: stresses on the deformed mesh

```
* STRESSES PLOT ON THE INITIAL SHAPE
TRAC (It's up to you) ;
```

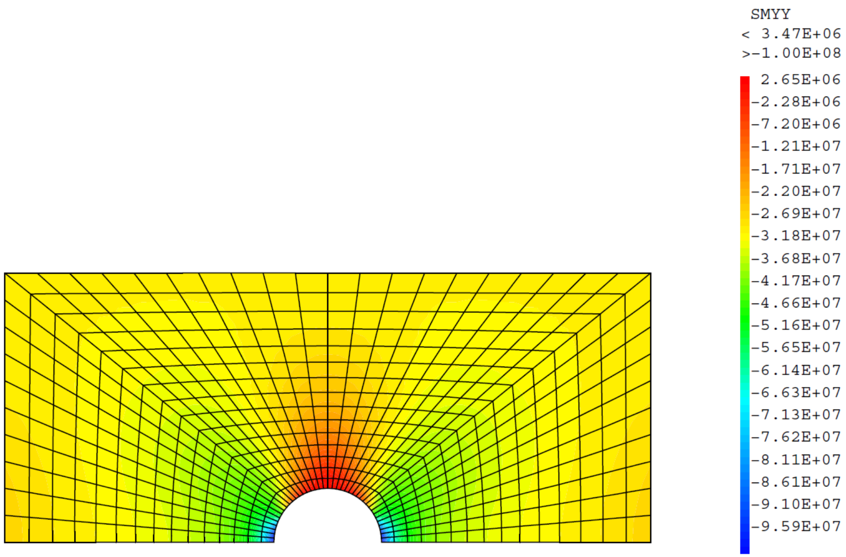
Post processing: stresses on the deformed mesh

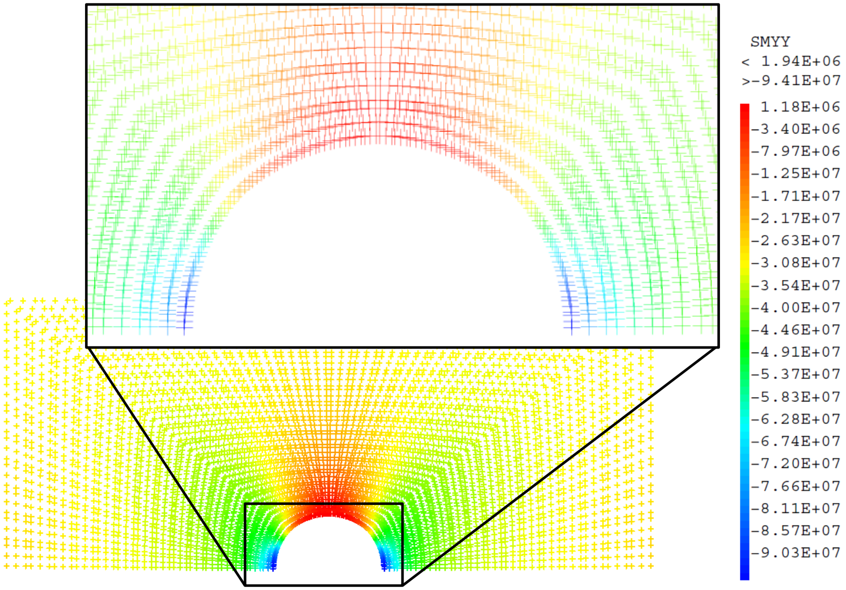
```
* STRESSES PLOT ON THE INITIAL SHAPE
TRAC SIG MOM1 ;

* STRESSES PLOT ON THE DEFORMED SHAPE
DEF_5B   = DEFO SU U5 150. ;
TRAC SIG MOM1 DEF_5B CSU ;
```



Post processing: stresses at Gauss points

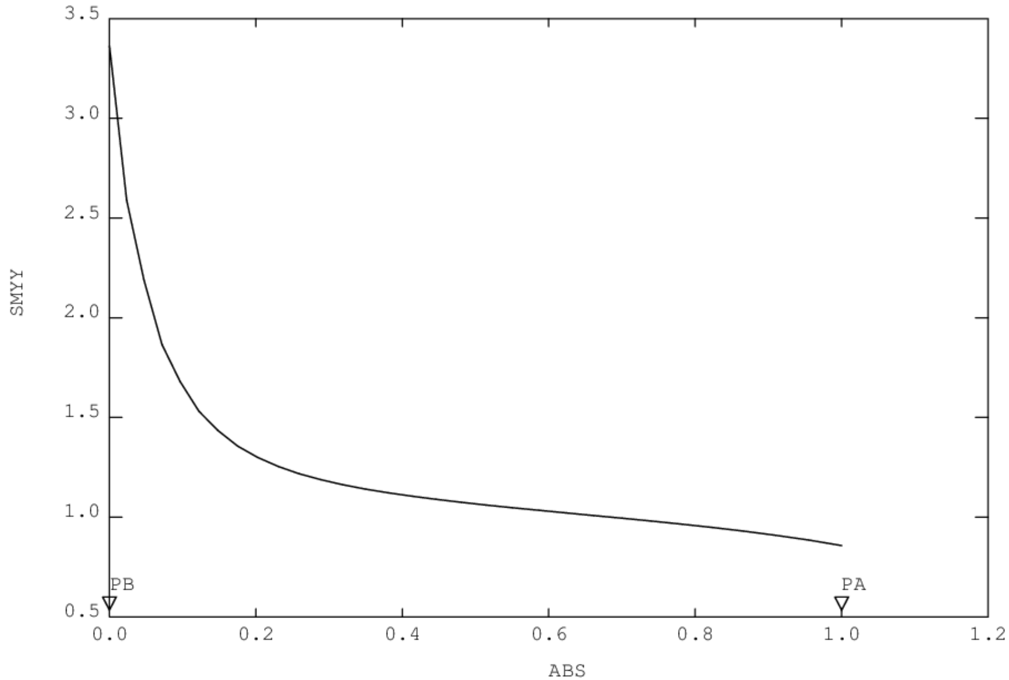
```
* STRESS PLOT ON THE MESH
SIGYY     = EXCO SIG 'SMYY' ;
MESS 'Mini and Maxi for Sigma YY : ' (MINI SIGYY)  (MAXI SIGYY) ;
TRAC SIGYY MOM1 ;

SIGYYG    = CHAN 'CHPO' SIGYY MOM1 'SUPP'  ;
MPGAUSS   = EXTR SIGYYG 'MAIL' ;
TRAC SIGYYG MPGAUSS ;
```





Post processing: stress concentration evolution curve (along the lower side)

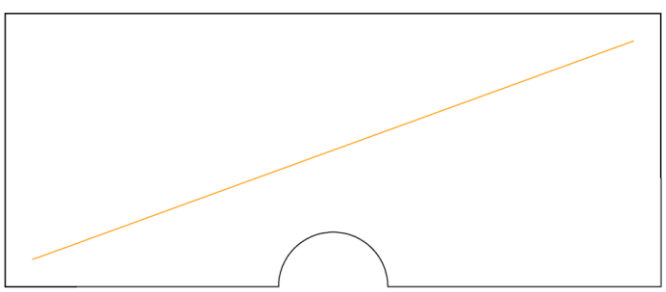
```
SIGB      = CHAN 'CHPO' SIG MOM1 ;
LBASG     = LBAS ELEM 'COMP' PB PA ;
EVSIG     = EVOL 'CHPO' SIGB 'SMYY' LBASG ;
EVK       = (ABS EVSIG) / 30.E6 ;
DESS EVK ;
```

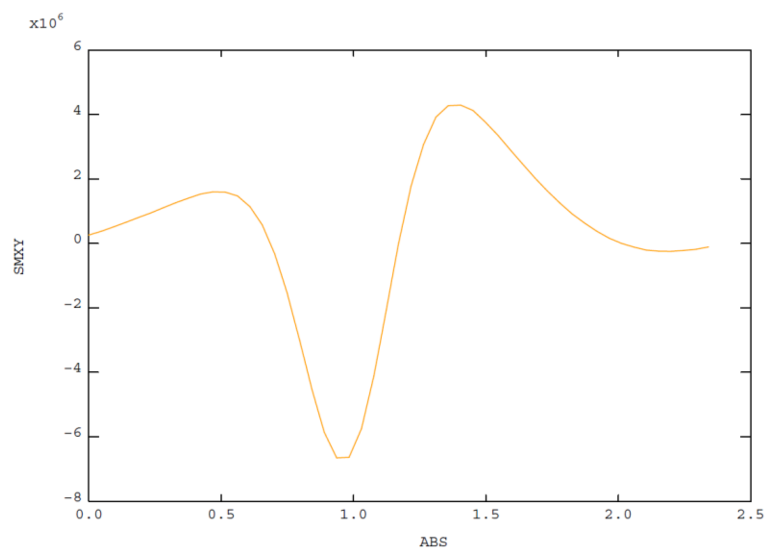


[5] Concentration de contrainte le long de LBAS

Post processing: evolution along any line

```
OPTI 'ELEM' 'SEG2' ;
LIG1      = DROI 50 (0.1 0.1) (2.3 0.9) COUL 'ORAN' ;
TRAC (CSU ET LIG1) ;

SIGT      = CHAN 'NOEUD' MOM1 SIG ;
SIGLBAS   = PROI SIGT LIG1 ;
EVSIG2    = EVOL 'ORAN' 'CHPO' SIGLBAS 'SMXY' LIG1 ;
DESS EVSIG2 ;
```



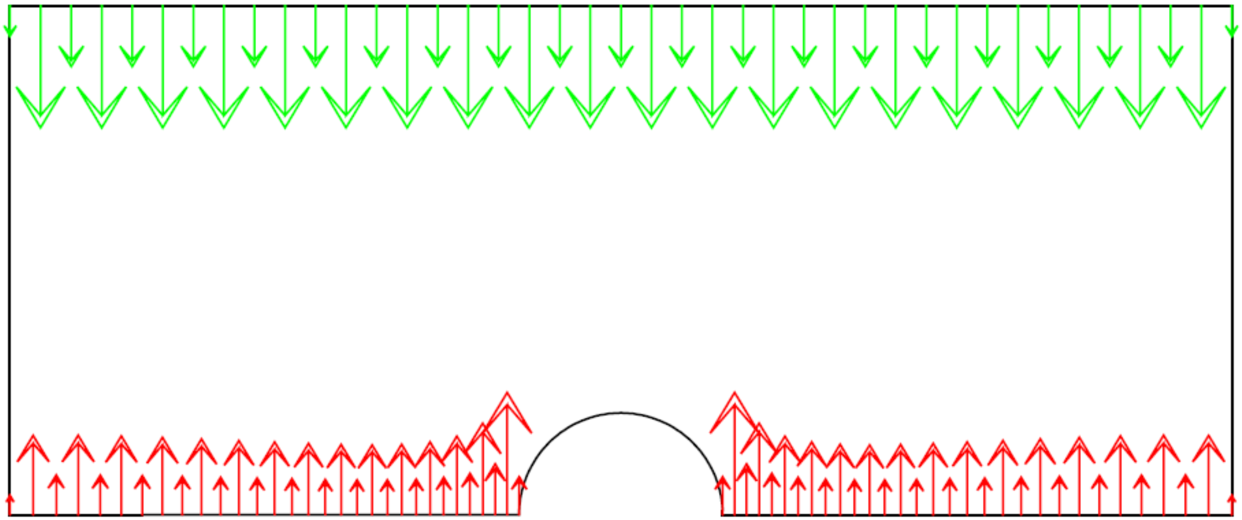


Post processing: force reaction

```
* SUPPORTS REACTIONS PLOT AS VECTORS
REAC1     = REAC U5 (BLMX ET BLMY) ;          (reaction due to a constraint)
VREAC     = VECT REAC1 'FORC' 'ROUG' ;        (arrows for plotting)

* IDEM FOR APPLIED FORCES
VFIMP     = (It's up to you)
TRAC (It's up to you) ;
```

Post processing: force reaction

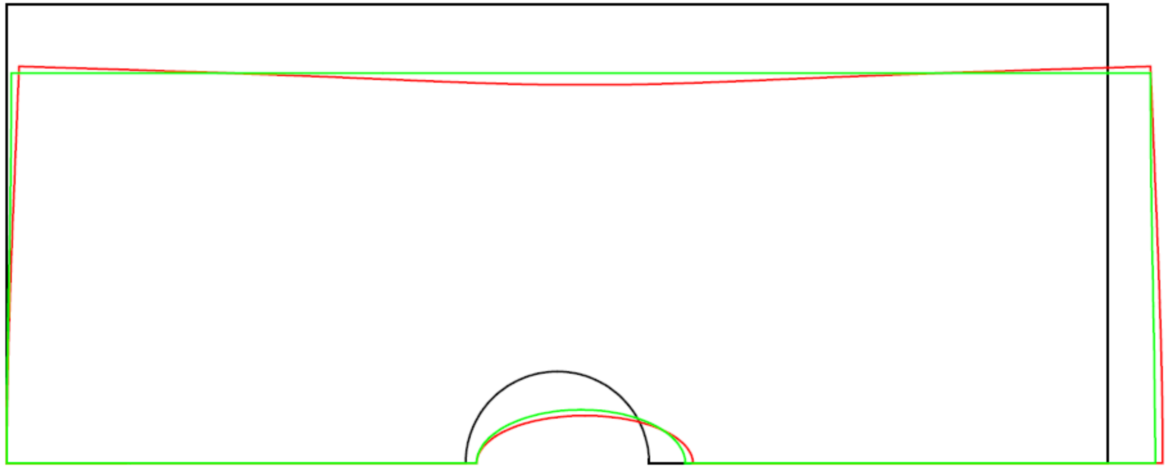
```
* SUPPORTS REACTIONS PLOT AS VECTORS
REAC1    = REAC U5 (BLMX ET BLMY) ;          (reaction due to a constraint)
VREAC    = VECT REAC1 'FORC' 'ROUG' ;        (arrows for plotting)

* IDEM FOR APPLIED FORCES
VFIMP    = VECT FS 'FORC' 'VERT' ;
TRAC (VFIMP ET VREAC) CSU ;
```



Bonus: case with prescribed displacements

link

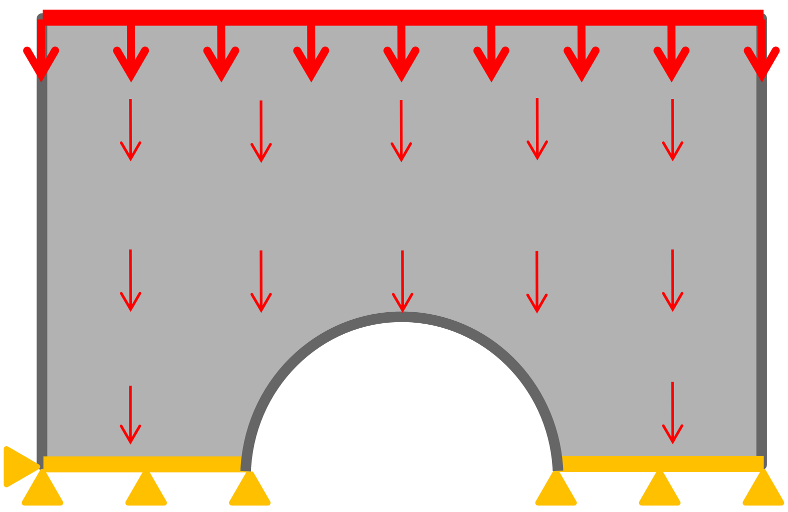
```
* STIFFNESS MATRICES (1ST MEMBER)
BLMY2     = BLOQ LHAUT 'UY' ;
RITOT2    = RI ET BLMX ET BLMY ET BLMY2 ;

* SECOND MEMBER ASSOCIETED TO THE STIFFNESS MATRIX
FU        = DEPI BLMY2 (8.E-3 * RAYON) ;

* RESOLUTION
U52       = RESO RITOT2 (FU ET FV) ;

* POST PROCESSING
DEF_5C2   = DEFO CSU U52 150. 'VERT' ;
TRAC (DEF_INIC ET DEF_5C ET DEF_5C2) ;
```



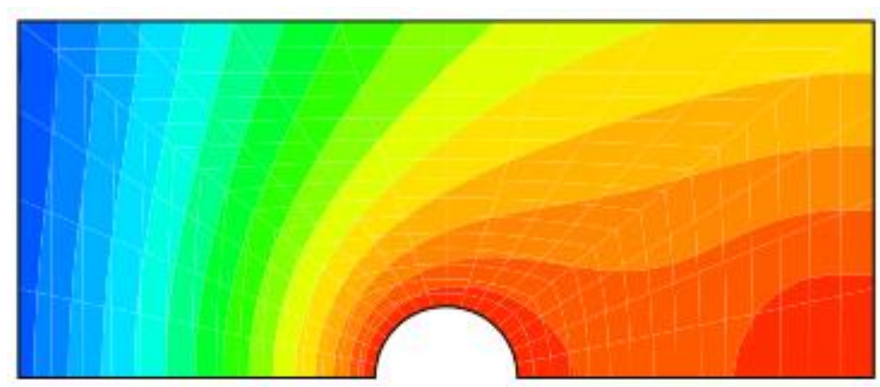
CHAP. 6: LINEAR ELASTICITY
THERMAL LOADING

***Objective*:** *previous mechanical calculation*
+ *thermal loading*

1. *thermal strain calculation*
2. *add the nodal forces equivalent to the thermal strain* **(link)**



+



Nodal forces resulting from thermal strain (2nd member)

```
* THERMAL STRAINS FROM TEMPERATURE FIELD CALCULATED
* TAKING INTO ACCOUNT CONVECTION + RADIATION
TFINAL    = TAB1 . 'TEMPERATURES' . (N1 - 1) ;
EPT       = EPTH TFINAL MOM1 MAM1 ;
```

$$\{\varepsilon\}^{th} = [\alpha]\{T - \text{TALP}\}$$

```
* PSEUDO STRESS FROM THERMAL STRAINS
SIT       = ELAS EPT MOM1 MAM1 ;
```

$$\{\sigma\}^{th} = [C]\{\varepsilon\}^{th}$$

```
* NODAL FORCES FROM THIS STRESS
FT        = BSIG SIT MOM1 ;
```

$$\{F\}^{th} = \int_V [B]^T\{\sigma\}^{th}\, dV$$

Linear system solving (additional term to the 2nd member)

```
* DISPLACEMENTS CALCULATION BY CALLING SOLVER WITH THERMAL PSEUDO FORCES
* ADDED TO THE MECHANICAL ONES
U6        = (It's up to you);
```

Nodal forces resulting from thermal strain (2nd member)

```
* THERMAL STRAINS FROM TEMPERATURE FIELD CALCULATED
* TAKING INTO ACCOUNT CONVECTION + RADIATION
TFINAL    = TAB1 . 'TEMPERATURES' . (N1 - 1) ;
EPT       = EPTH TFINAL MOM1 MAM1 ;
```

$$\{\varepsilon\}^{th} = [\alpha]\{T - \text{TALP}\}$$

```
* PSEUDO STRESS FROM THERMAL STRAINS
SIT       = ELAS EPT MOM1 MAM1 ;
```

$$\{\sigma\}^{th} = [C]\{\varepsilon\}^{th}$$

```
* NODAL FORCES FROM THIS STRESS
FT        = BSIG SIT MOM1 ;
```

$$\{F\}^{th} = \int_V [B]^T \{\sigma\}^{th}\, dV$$

Linear system solving (additional term to the 2nd member)

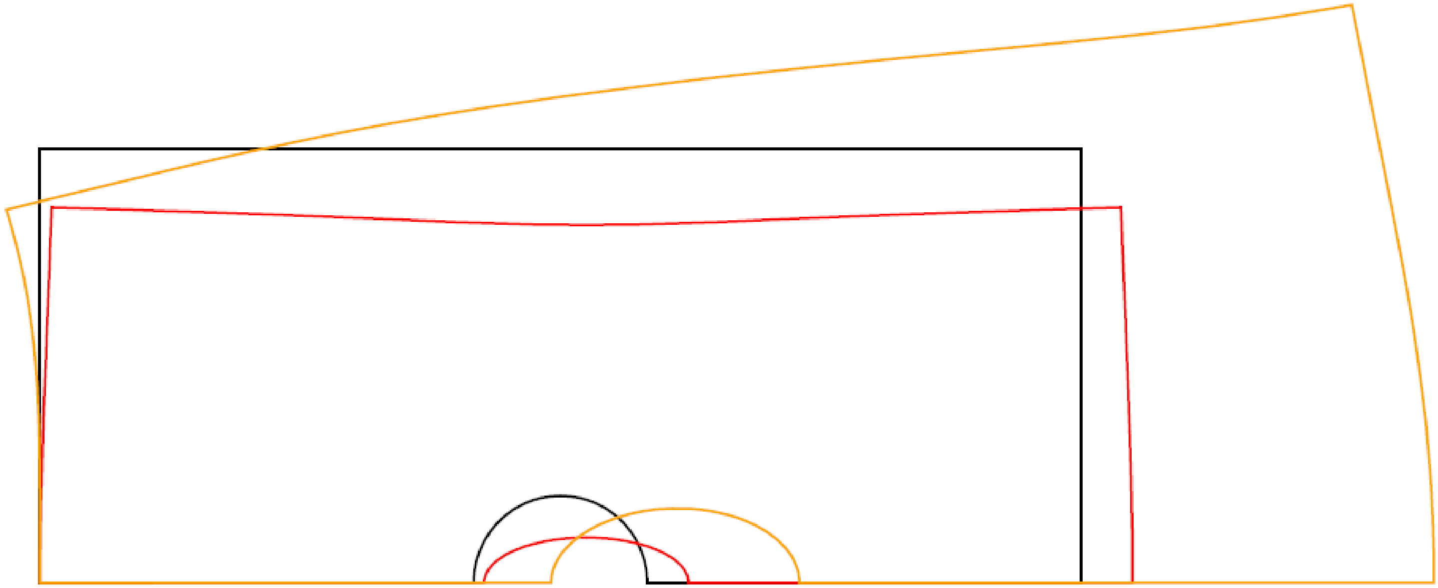
```
* DISPLACEMENTS CALCULATION BY CALLING SOLVER WITH THERMAL PSEUDO FORCES
* ADDED TO THE MECHANICAL ONES
U6        = RESO RITOT (FS ET FV ET FT) ;
```

Post processing: deformed mesh, strains, stresses

```
* THERMOMECANICAL DEFORMED MESH
DEF_6     = (It's up to you)
DEF_6C    = (It's up to you)
TRAC (It's up to you) ;
```

Post processing: deformed mesh, strains, stresses

```
* THERMOMECANICAL DEFORMED MESH
DEF_6    = DEFO SU  U6 150. ;
DEF_6C   = DEFO CSU U6 150. 'ORAN' ;
TRAC (DEF_INIC ET DEF_5C ET DEF_6C) ;
```

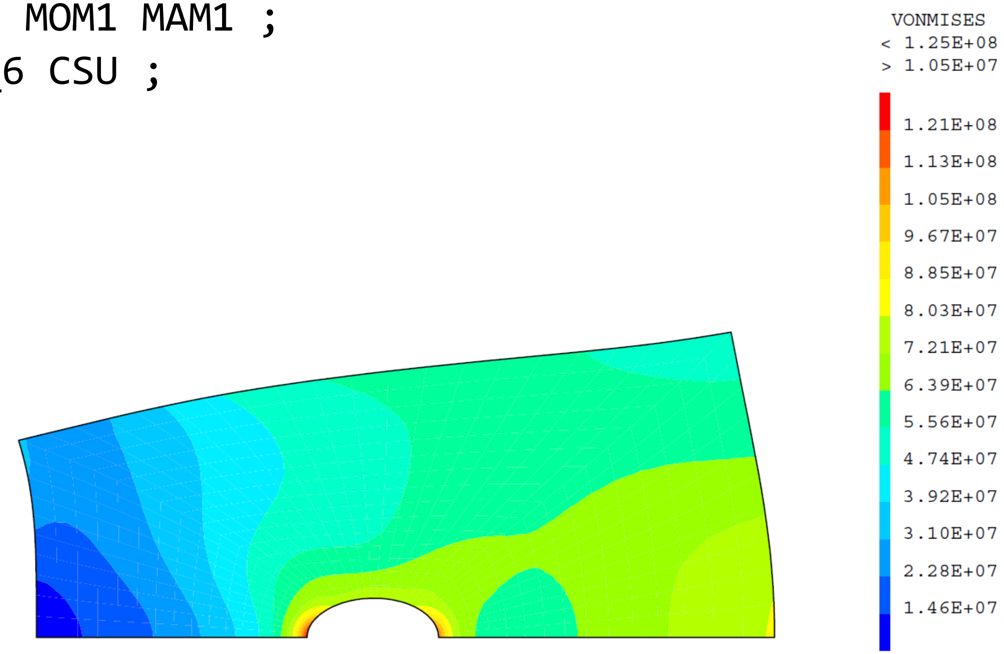


Post processing: deformed mesh, strains, stresses

```
* TOTAL STRAINS
EP        = (It's up to you, EPSI operator)

* ELASTIC STRAINS
EPE       = (It's up to you)

* STRESSES CALCULATED FROM ELASTIC STRAINS
SIGT      = (It's up to you)
TRAC SIGT MOM1 DEF_6 CSU ;
```

Post processing: deformed mesh, strains, stresses

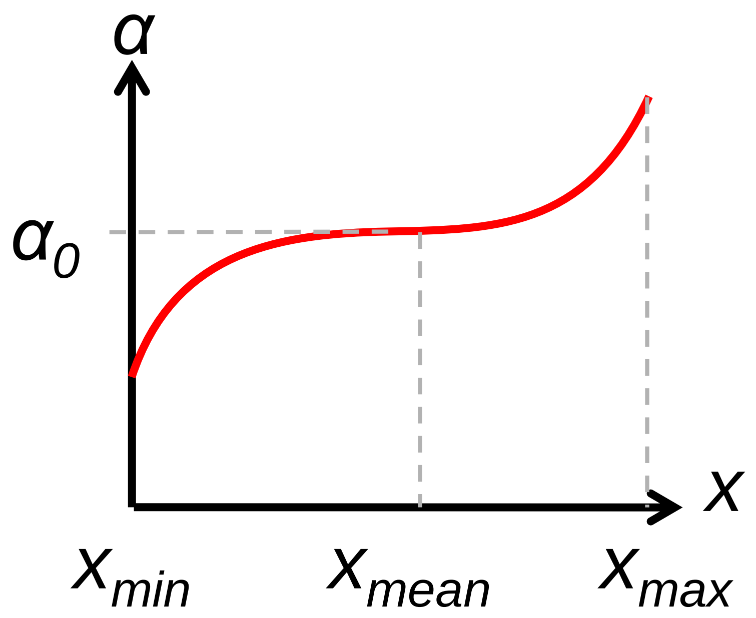
```
* TOTAL STRAINS
EP        = EPSI MOM1 U6 'LINE' ;

* ELASTIC STRAINS
EPE       = EP - EPT ;

* STRESSES CALCULATED FROM ELASTIC STRAINS
SIGT      = ELAS EPE MOM1 MAM1 ;
TRAC SIGT MOM1 DEF_6 CSU ;
```



Objective: *previous thermo-mechanical calculation*
+ thermal expansion α variable in space

1. *calculation of the field α(x) (from coordinates)*
2. *material property described by α(x)*
3. *updating of the nodal thermal pseudo forces*

$$\alpha(x) = \alpha_0 \left(1 + \beta \left(\frac{x - x_{\text{mean}}}{x_{\text{max}} - x_{\text{min}}}\right)^3\right)$$



Non uniform thermal expansion coefficient

```
XX        = COOR SU 1 ;
XMAX      = MAXI XX ;
XMIN      = MINI XX ;
XMOY      = 0.5 * (XMAX + XMIN) ;

BETA      = 7. ;
CHP_ALPH = ALPHAMAT * (1. + (BETA * (((XX - XMOY) / (XMAX - XMIN)) ** 3))) ;
```

Material properties updating

```
* CONVERSION FROM CHPOINT -> TO MCHAML
CHM_ALPH = CHAN 'CHAM' CHP_ALPH MOM1 ;

* UPDATING OF THE MATERIAL MCHAML WITH THE NON UNIFORM ALPHA
MAM1B     = MATE MOM1 'YOUN' YOUNGMAT 'NU' NUMAT
                      'ALPH' CHM_ALPH 'TREF' T0 'TALP' T0 ;
```

Updating the nodal forces due to thermal strains (2^{nd} member)

```
* UPDATING THE PURE THERMAL STRAINS WITH ALPHA VARIABLE
EPT       = (It's up to you, EPTH operator)

* UPDATING THE THERMAL PSEUDO STRESSES WITH ALPHA VARIABLE
SIT       = (It's up to you, ELAS operator)

* UPDATING THE NODAL FORCES FOR THESE THERMAL STRAINS
FT        = (It's up to you, BSIG operator)
```

Linear system solving

```
* DISPLACEMENTS WITH ALPHA VARIABLE
U7        = RESO RITOT (FS ET FV ET FT) ;
```

Updating the nodal forces due to thermal strains (2nd member)

```
* UPDATING THE PURE THERMAL STRAINS WITH ALPHA VARIABLE
EPT       = EPTH TFINAL MOM1 MAM1B ;

* UPDATING THE THERMAL PSEUDO STRESSES WITH ALPHA VARIABLE
SIT       = ELAS EPT MOM1 MAM1B ;

* UPDATING THE NODAL FORCES FOR THESE THERMAL STRAINS
FT        = BSIG SIT MOM1 ;
```

Linear system solving

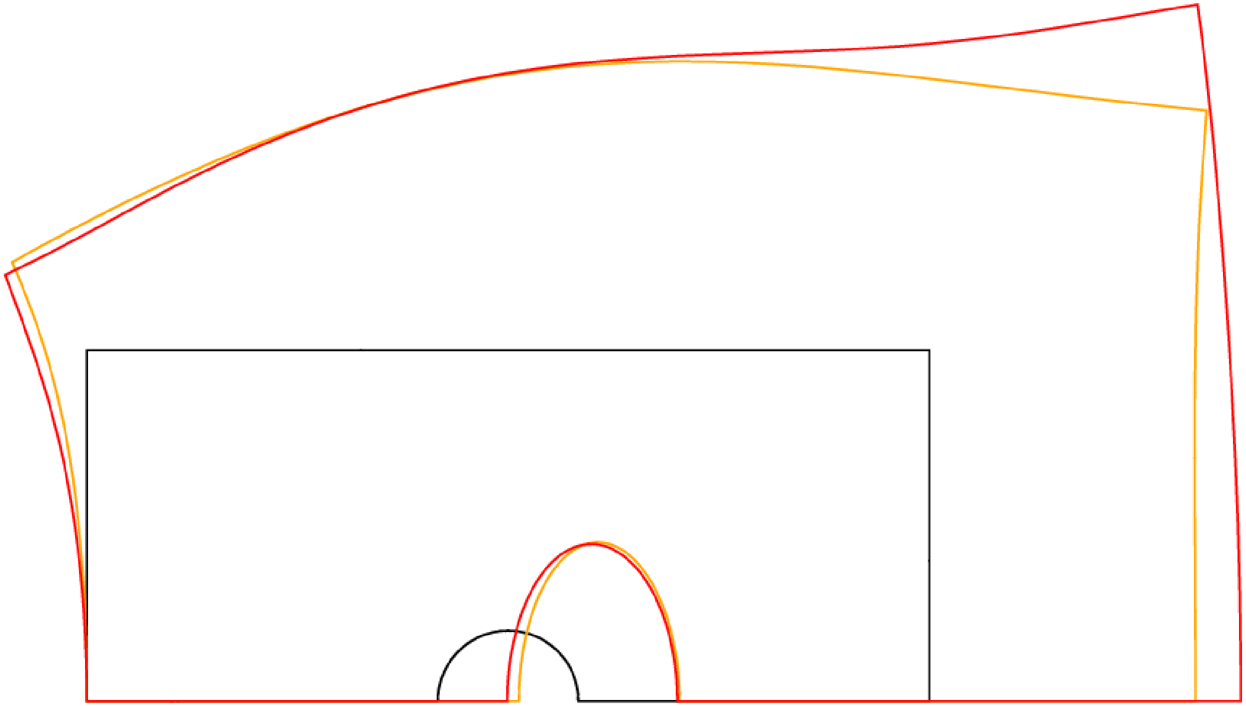
```
* DISPLACEMENTS WITH ALPHA VARIABLE
U7        = RESO RITOT (FS ET FV ET FT) ;
```

Post processing: deformed mesh, strains, stresses

```
* DEFORMED SHAPE WITH ALPHA VARIABLE
(It's up to you)
(It's up to you)
TRAC (It's up to you) ;
```

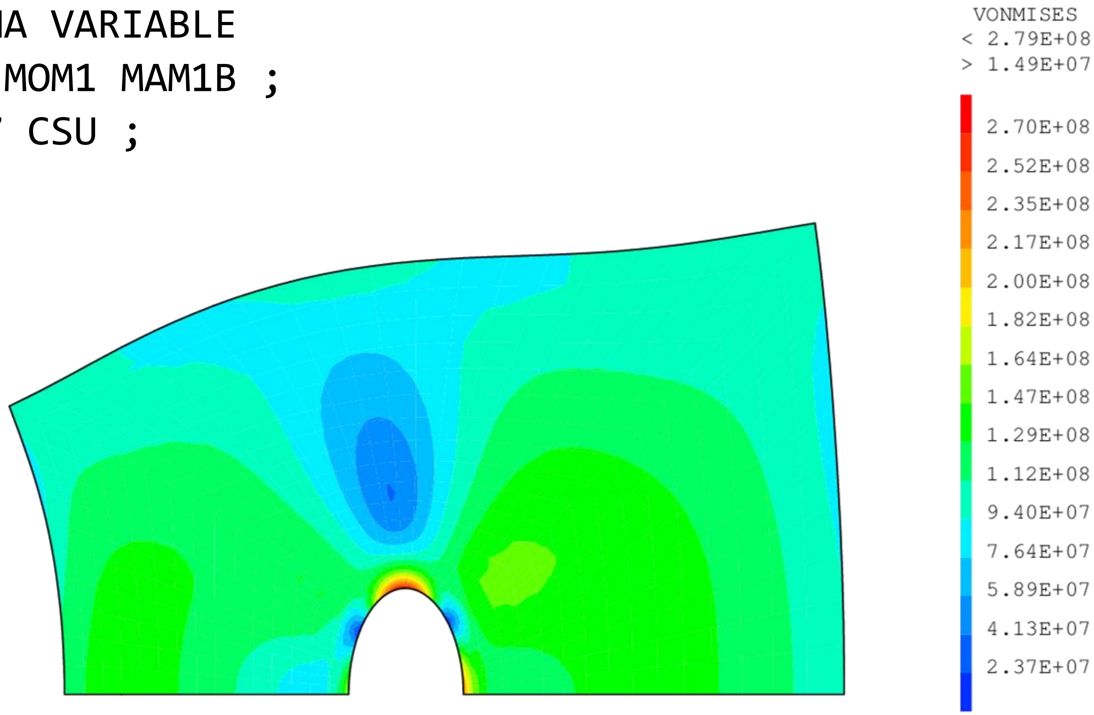
Post processing: deformed mesh, strains, stresses

```
* DEFORMED SHAPE WITH ALPHA VARIABLE
DEF_7     = DEFO SU  U7 150. ;
DEF_7C    = DEFO CSU U7 150. 'ROUG' ;
TRAC (DEF_INIC ET DEF_6C ET DEF_7C) ;
```



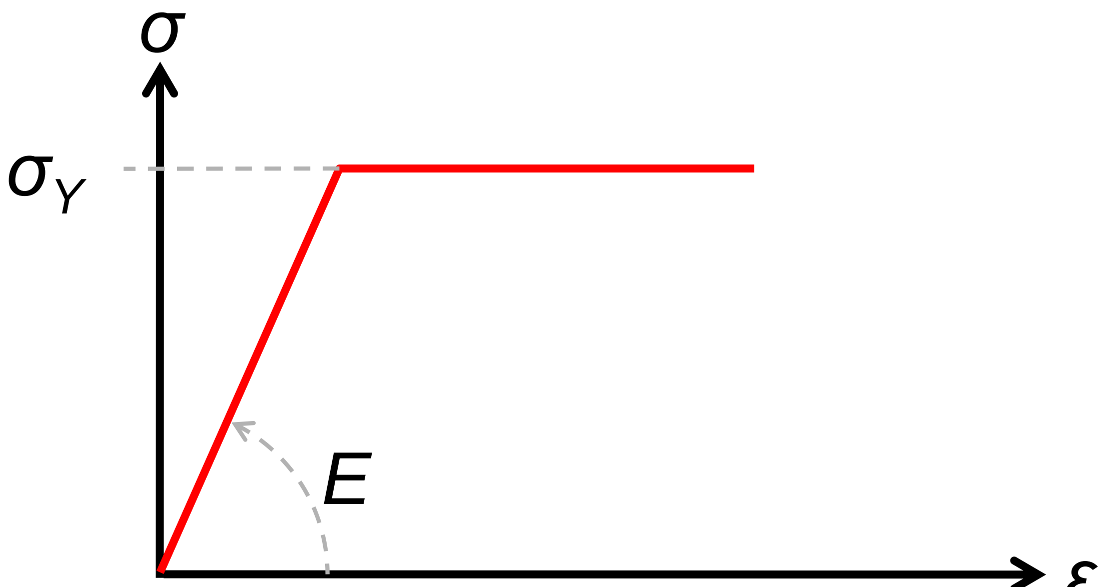
Post processing: deformed mesh, strains, stresses

```
* STRAINS WITH ALPHA VARIABLE
EP        = EPSI U7 MOM1 ;

* ELASTIC STRAINS WITH ALPHA VARIABLE
EPE       = EP - EPT ;

* STRESSES WITH ALPHA VARIABLE
SIGT      = ELAS EPE MOM1 MAM1B ;
TRAC SIGT MOM1 DEF_7 CSU ;
```



Objective: *previous thermo-mechanical calculation*
+ perfect elasto-plastic material

1. *starting from the previous transient thermal analysis*
2. *add a non-linear mechanical model*
3. *temporal description of mechanical loadings*
4. *solving with the* PASAPAS *procedure*



Model and material properties updating

```
* MECHANICAL MODEL UPDATING
MOM2      = MODE SU 'MECANIQUE' 'ELASTIQUE' 'PLASTIQUE' 'PARFAIT' ;

* MATERIAL PROPERTIES UPDATING (YIELD STRENGTH 'SIGY' ADDED TO PROPERTIES)
MAM2      = MATE MOM2 'YOUN' YOUNGMAT 'NU' NUMAT 'SIGY' SIGYMAT
                      'ALPH' CHM_ALPH 'TREF' T0  'TALP ' T0 ;
```

Incremental mechanical loading

```
* TEMPORAL MECHANICAL LOADS DESCRIPTION
EVTM      = EVOL 'MANU'  (PROG 0. (0.98 * TPSFIN) TPSFIN)
                         (PROG 0. 0.              1.) ;
CHAFS     = CHAR 'MECA' FS EVTM ;

CHAFV     = CHAR 'MECA' FV EVCTE ;

CHAM      = CHAFS ET CHAFV ;
```

Building of the table for PASAPAS procedure

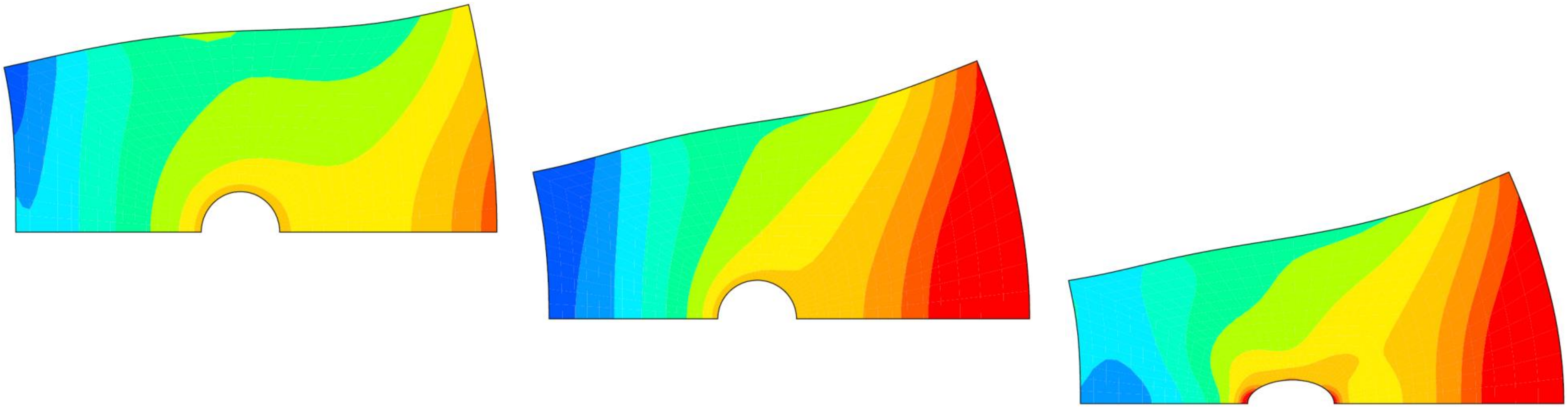
```
* PASAPAS TABLE
TAB2                          = TABL ;

.........
.........
.........
        (It's up to you)
.........
.........
.........
PASAPAS TAB2 ;
```

Building of the table for PASAPAS procedure

```
* PASAPAS TABLE
TAB2                           = TABL ;
TAB2 . 'MODELE'                = MOT ET MOC ET MOR ET MOM2 ;
TAB2 . 'CARACTERISTIQUES'      = MAT ET MAC ET MAR ET MAM2 ;
TAB2 . 'BLOCAGES_MECANIQUES'   = BLMX ET BLMY ;
TAB2 . 'BLOCAGES_THERMIQUES'   = BLT ;
TAB2 . 'CHARGEMENT'            = CHAT ET CHAM ;
TAB2 . 'TEMPS_CALCULES'        = PROG 0. 'PAS' (0.02 * TPSFIN)   (0.98 * TPSFIN)
                                          'PAS' (0.001 * TPSFIN)  TPSFIN ;
TAB2 . 'CELSIUS'               = VRAI ;
PASAPAS TAB2 ;
```

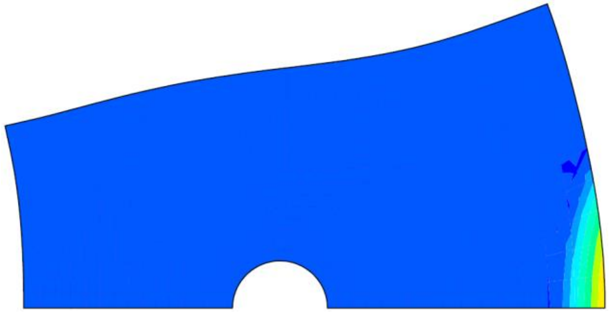
Post-processing: stresses

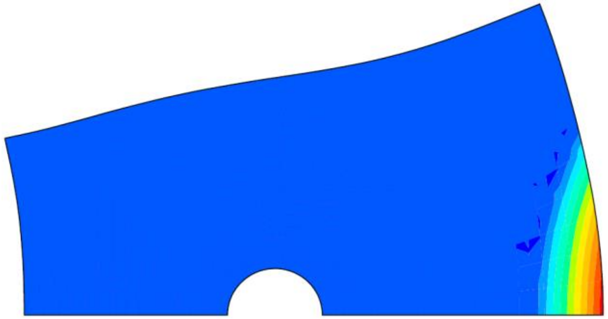
```
* LOOP ON EACH TIME STEP
LISO2    = PROG 0. 'PAS' 10.E6 100.E6 ;
N2       = DIME (TAB2 . 'DEPLACEMENTS') ;
REPE B1 N2 ;
  U_I      = TAB2 . 'DEPLACEMENTS' . (&B1 - 1) ;
  DEF_I    = DEFO SU U_I 150. ;
  S_I      = TAB2 . 'CONTRAINTES' . (&B1 - 1) ;
  TRAC S_I MOM2 DEF_I CSU LISO2 ;
FIN B1 ;
DEF_8C   = DEFO CSU U_I 150. 'VIOL' ;
```

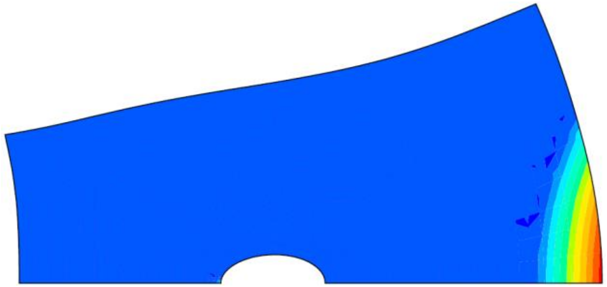


Post-processing: cumulated plastic strains (link)

```
* LOOP ON EACH TIME STEP
VI        = TAB2 . 'VARIABLES_INTERNES' . (N2 - 1) ;
EQ_MAX    = MAXI (EXCO 'EPSE' VI) ;
LISO3     = PROG 0. 'PAS' (EQ_MAX / 10.) EQ_MAX ;
REPE B1 N2 ;
  U_I       = TAB2 . 'DEPLACEMENTS' . (&B1 - 1) ;
  DEF_I     = DEFO SU U_I 150. ;
  V_I       = TAB2 . 'VARIABLES_INTERNES' . (&B1 - 1) ;
  TRAC V_I MOM2 DEF_I CSU LISO3 ;
FIN B1 ;
```

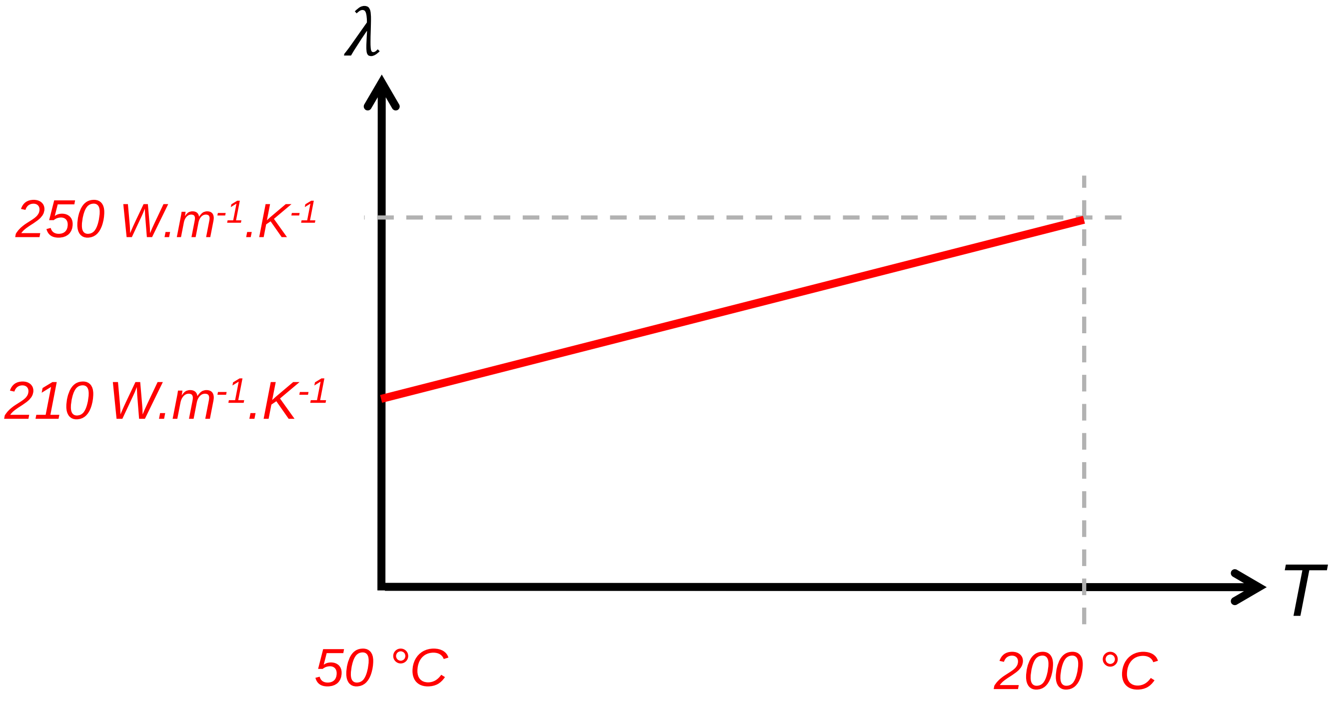






Objective: *previous thermo-mechanical calculation*
+ conductivity depending on temperature

1. *variable material property depending on the thermal unknown*



Updating of material properties

```
* VARIATION OF CONDUCTIVITY AS A FUNCTION OF T
EVLAM    = EVOL 'MANU' 'T' (PROG 50. 200.) 'K' (PROG 210. 250.) ;

* UPDATING OF THE MATERIAL
MAT2     = MATE MOT 'K' EVLAM 'C' CAPAMAT 'RHO' RHOMAT ;
```

PASAPAS call

(It's up to you)

Updating of material properties

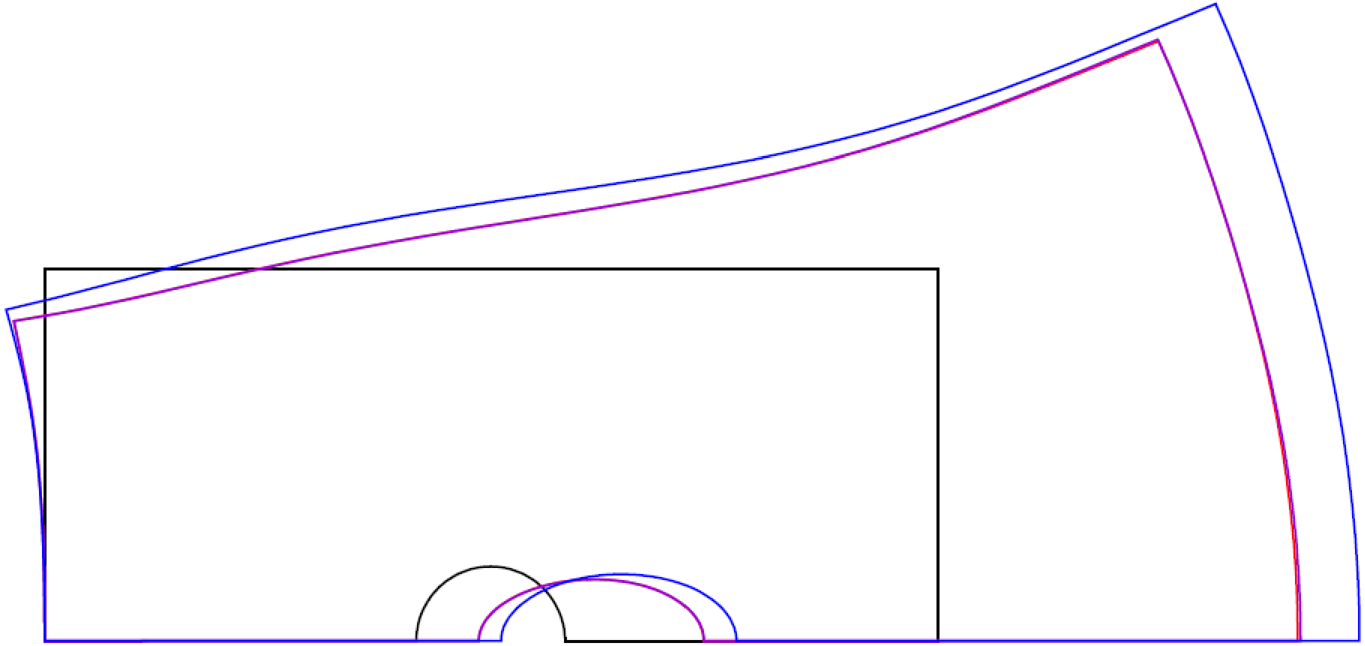
```
* VARIATION OF CONDUCTIVITY AS A FUNCTION OF T
EVLAM    = EVOL 'MANU' 'T' (PROG 50. 200.) 'K' (PROG 210. 250.) ;

* UPDATING OF THE MATERIAL
MAT2     = MATE MOT 'K' EVLAM 'C' CAPAMAT 'RHO' RHOMAT ;
```

PASAPAS call

```
TAB2                            = TABL ;
TAB2 . 'MODELE'                 = MOT  ET MOC ET MOR ET MOM2 ;
TAB2 . 'CARACTERISTIQUES'       = MAT2 ET MAC ET MAR ET MAM2 ;
TAB2 . 'BLOCAGES_MECANIQUES'    = BLMX ET BLMY ;
TAB2 . 'BLOCAGES_THERMIQUES'    = BLT ;
TAB2 . 'CHARGEMENT'             = CHAT ET CHAM ;
TAB2 . 'TEMPS_CALCULES'         = PROG 0. 'PAS' (0.02 * TPSFIN)  (0.98 * TPSFIN)
                                          'PAS' (0.001 * TPSFIN) TPSFIN ;
TAB2 . 'CELSIUS'                = VRAI ;
PASAPAS TAB2 ;
```

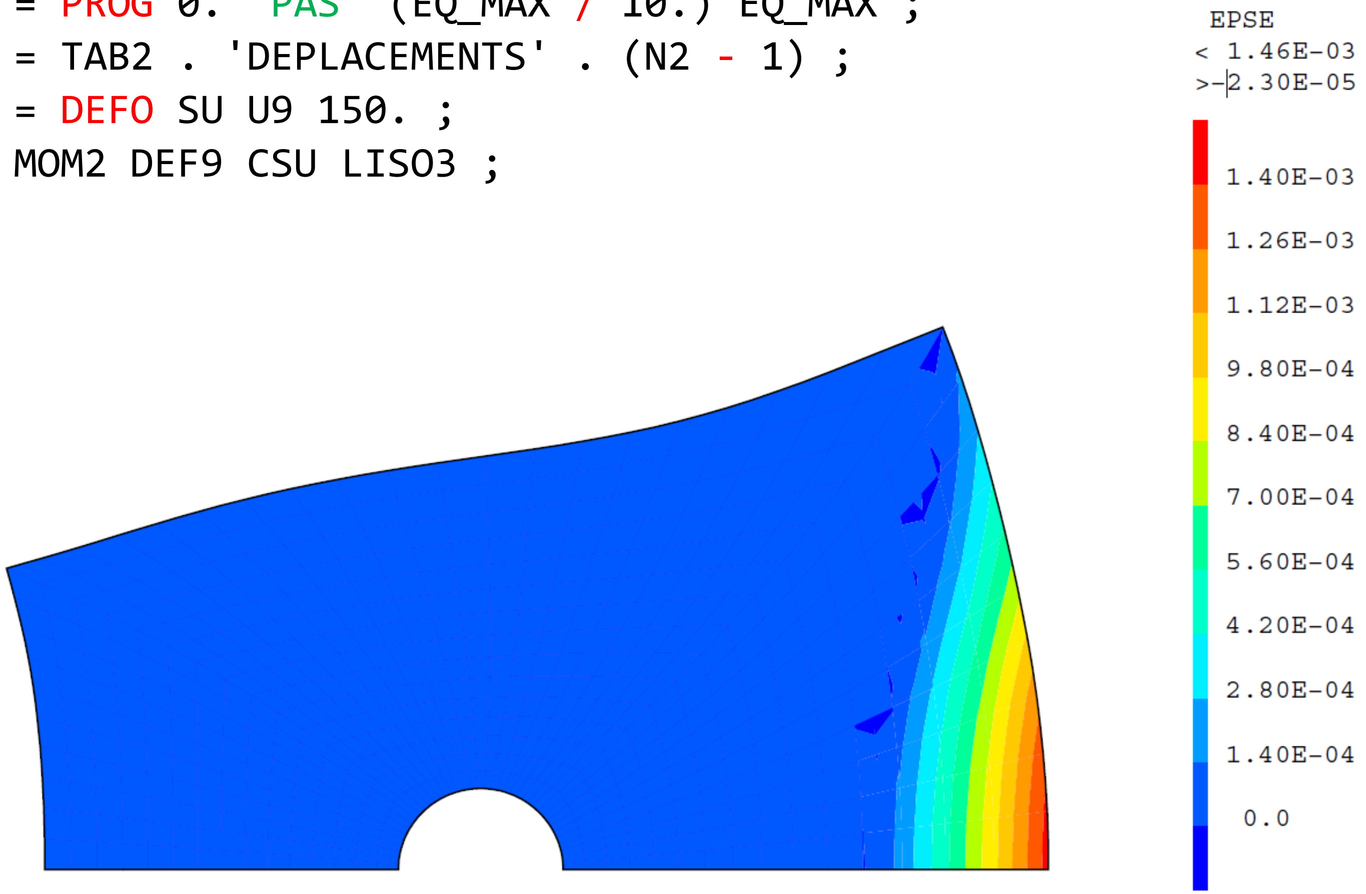
Post processing: deformed mesh

```
* COMPARISON OF THE FINAL SHAPE WITH THOSE OF THE PREVIOUS CALCULATIONS
N2        = DIME (TAB2 . 'DEPLACEMENTS') ;
U9        = TAB2 . 'DEPLACEMENTS' . (N2 - 1) ;
DEF_9     = DEFO SU  U9 150. ;
DEF_9C    = DEFO CSU U9 150. 'BLEU' ;
TRAC (DEF_INIC ET DEF_7C ET DEF_8C ET DEF_9C) ;
```



Post processing: cumulated plastic strains

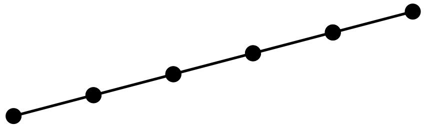
```
VI9       = TAB2 . 'VARIABLES_INTERNES' . (N2 - 1) ;
EQ_MAX    = MAXI (EXCO 'EPSE' VI9) ;
LISO3     = PROG 0. 'PAS' (EQ_MAX / 10.) EQ_MAX ;
U9        = TAB2 . 'DEPLACEMENTS' . (N2 - 1) ;
DEF9      = DEFO SU U9 150. ;
TRAC VI9 MOM2 DEF9 CSU LISO3 ;
```



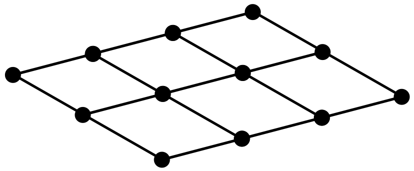
ADDENDA

- The choice of the finite elements is made in MODE

```
MODP1 = MODE LIG1 'MECANIQUE' 'ELASTIQUE' | 'POUT' | ;
                                          | 'TIMO' |
```



```
MODC1 = MODE SUR1 'MECANIQUE' 'ELASTIQUE' | 'DKT'  | ;
                                          | 'COQ8' |
```



- You should give the geometrical characteristics

```
MATP1 = MATE MODP1 'YOUN' 210.E9 'NU   ' 0.3
                   'SECT' 1.E-2  'INRY' 1.E-4 'INRZ' 2.E-4 'TORS' 3.E-4 ;

MATC1 = MATE MODC1 'YOUN' 210.E9 'NU   ' 0.3
                   'EPAI' 1.E-2 ;
```

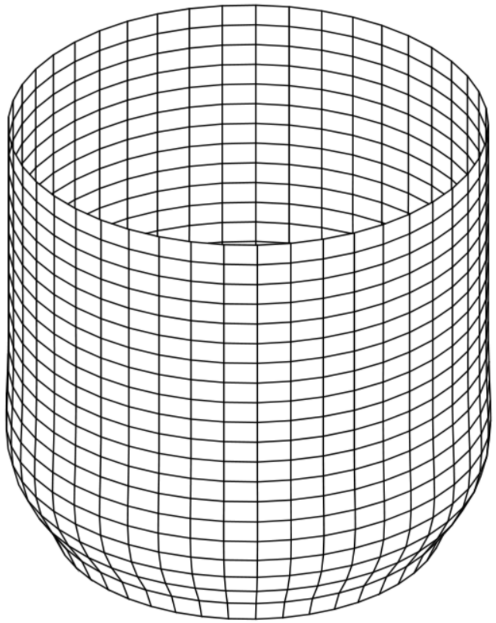
- Then you can prescribe d.o.f. for displacements and rotation

```
BL1 = BLOQ P1  'UX' 'UY' 'UZ'  'RX' 'RY' 'RZ' ;
```

SHELL ELEMENTS IN 3D

- Example of a cylindrical shell under pressure (in 3D)

```
OPTI 'DIME' 3 'ELEM' 'QUA4' ;
p1 = 4. 0. 0. ;
lig1 = CERC 45 'ROTA' 360. p1 (0. 0. 0.) (0. 0. 1.) ;
ELIM lig1 1.E-2 ;
sur1 = lig1 TRAN 20 (0. 0. 10.) ;

mo = MODE sur1 'MECANIQUE' 'COQ4' ;
ma = MATE mo 'YOUN' 210.E9 'NU' 0.3 'EPAI' 0.3 ;
rig = RIGI mo ma ;

bl1 = BLOQ 'UX' 'UY' 'UZ' 'RX' 'RY' 'RZ' lig1 ;
f1  = PRES 'COQU' mo 42.E5 'NORM' ma ;

u = RESO (rig ET bl1) f1 ;
def1 = DEFO sur1 u ;
TRAC 'CACH' def1 ;
```



SHELL ELEMENTS IN 2D

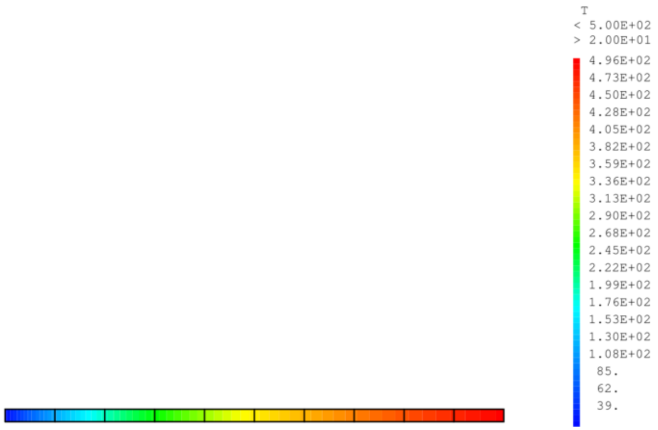
- The same case (in 2D axisymmetric)

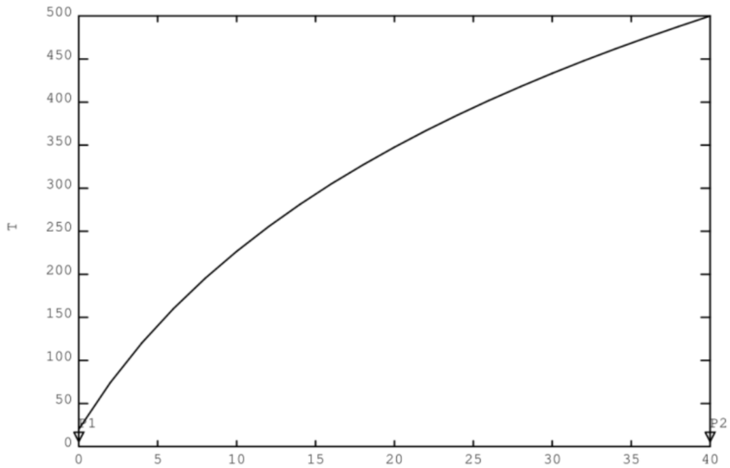
```
OPTI 'DIME' 2 'MODE' 'AXIS' 'ELEM' 'SEG2' ;
p1 = 4. 0. ;
lig1 = DROI 20 p1 (4. 10.) ;



mo = MODE lig1 'MECANIQUE' 'COQ2' ;
ma = MATE mo 'YOUN' 210.E9 'NU' 0.3 'EPAI' 0.3 ;
rig = RIGI mo ma ;

bl1 = BLOQ 'UR' 'UZ' 'RT' p1 ;
f1  = PRES 'COQU' mo -42.E5 'NORM' ;

u = RESO (rig ET bl1) f1 ;
def1 = DEFO lig1 u ;
TRAC def1 ;
```

AND SOME OTHER OPTIONS (1D, 2D AXIS, ETC…)

- Example of steady state 2D axisymmetric conduction problem

```
OPTI 'DIME' 2 'MODE' 'AXIS' 'ELEM' 'QUA8' ;

p1 = 10. 0. ;
p2 = 50. 0. ;
lig1 = DROI 10 p1 p2 ;
sur1 = lig1 TRAN 1 (0. 1.) ;
cot1 = sur1 COTE 4 ;
cot2 = sur1 COTE 2 ;

mo = MODE sur1 'THERMIQUE' ;
ma = MATE mo 'K' 42. ;
con = COND mo ma ;

bl1 = BLOQ 'T' cot1 ;
bl2 = BLOQ 'T' cot2 ;
f1  = DEPI bl1 20. ;
f2  = DEPI bl2 500. ;

t = RESO (con ET bl1 ET bl2) (f1 ET f2) ;
TRAC t sur1 ;
evt = EVOL 'CHPO' t 'T' lig1 ;
DESS evt ;
```

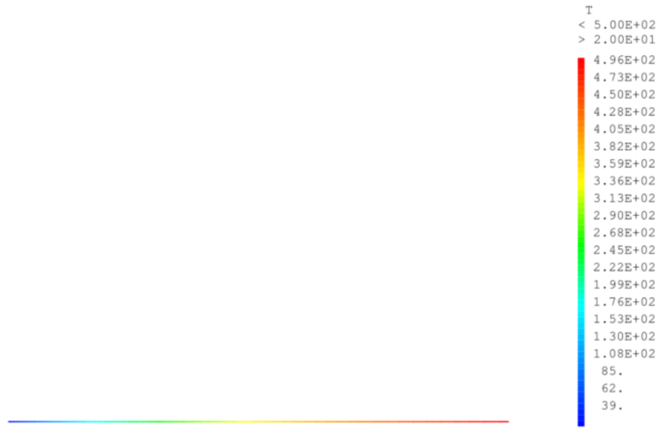


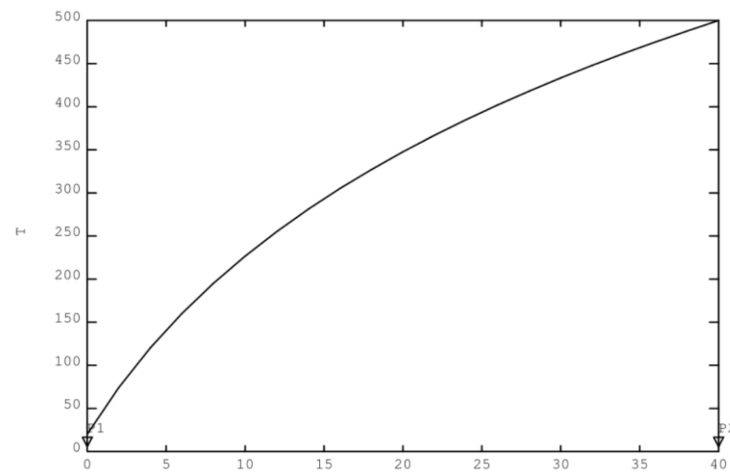


AND SOME OTHER OPTIONS (1D, 2D AXIS, ETC…)

- The same case in 1D (cylindrical)

```
OPTI 'DIME' 1 'MODE' 'UNID' 'AXIS' 'ELEM' 'SEG3' ;

p1 = POIN 10. ;
p2 = POIN 50. ;
lig1 = DROI 10 p1 p2 ;

mo = MODE lig1 'THERMIQUE' ;
ma = MATE mo 'K' 42. ;
con = COND mo ma ;

bl1 = BLOQ 'T' p1 ;
bl2 = BLOQ 'T' p2 ;
f1  = DEPI bl1 20. ;
f2  = DEPI bl2 500. ;

t = RESO (con ET bl1 ET bl2) (f1 ET f2) ;
TRAC t lig1 ;
evt = EVOL 'CHPO' t 'T' lig1 ;
DESS evt ;
```





READING / WRITING DATA

- **Binary file: backup / recovery**

```
OPTI 'SAUV' 'foo' ;   SAUV ;
OPTI 'REST' 'foo' ;   REST ;
```

  Also possible by a text file

- **Run an EXTErnal command**

```
TAB1 = EXTE 'grep -in mechanical foo.dgibi' 'RC' ;
```

  ➔ TAB1 contents the output of the `grep` command

- **The ACQUisition command**

  Read a texte file, line by line

```
OPTI 'ACQU' 'foo.txt' ;
ACQU N1*'ENTIER' A*'FLOTTANT' ;
ACQU N2*'ENTIER' L1*'LISTREEL' 4 ;
```

foo.txt

```
1  3.14  X     Y     Z
2  25.2  28.3  24.3  16.6
```

Writing in a text file
➔ using the SORT directive

```
OPTI 'SORT' 'my_file.txt'  ;

SORT 'CHAI' 'I am writing a text file!' ;
SORT 'CHAI' 4 8 15 16 23 42 ;
SORT 'CHAI' ' ' ;

SORT 'CHAI' '#iteration     Fibonacci' ;
FIBO1 = 0 ;
FIBO2 = 1 ;
SORT 'CHAI' 0   FIBO1 ;
SORT 'CHAI' 1   FIBO2 ;
REPE B1 15 ;
  FIBO2B = FIBO2 ;
  FIBO2 = FIBO1 + FIBO2 ;
  FIBO1 = FIBO2B ;
  SORT 'CHAI' (&B1 + 1)   FIBO2 ;
FIN  B1 ;
```

foo.txt

```
 I am writing in a text file!
     4         8        15        16        23        42

 #iteration         Fibonacci
     0                 0
     1                 1
     2                 1
     3                 2
     4                 3
     5                 5
     6                 8
     7                13
     8                21
     9                34
    10                55
    11                89
    12               144
    13               233
    14               377
    15               610
    16               987
```

READING / WRITING DATA

- **LIRE / SORT** **reading / writing different file formats**

Read/write tabular data (CSV or other)
concerned objects: lists, TABLE
used by text editors, spreadsheet (Excel)

```
OPTI 'SORT' 'foo.csv' ;
SORT 'EXCE' OBJECT1 ;

TAB1 = 'LIRE' 'CSV' 'foo.csv' ;
```

The column separator can be changed:
semicolon, comma, space, tab, slash

READING / WRITING DATA

- **LIRE / SORT** **reading / writing different file formats**

Read the UNV format (text file)
concerned objects: MAILLAGE (meshes)
used by Gmsh, Salomé, HyperMesh, …

```
TAB1 = LIRE 'UNV' 'foo.unv' ;
```

Read/wrtite the AVS format (text file)
concerned objects: MAILLAGE, CHPOINT, MCHAML
.inp extension used by Abaqus

```
OPTI 'SORT' 'foo.inp' ;
SORT 'AVS' OBJET1 'TEMPS' 12.3 ;

OPTI 'LECT' 'foo.inp' ;
TAB1 = LIRE 'AVS' ;
```

READING / WRITING DATA

- **LIRE / SORT** **reading / writing different file formats**

  Write the VTK format
  concerned objects: MAILLAGE, CHPOINT, MCHAML
  used by Paraview

```
OPTI 'SORT' 'foo.vtk' ;
SORT 'VTK' MAIL1 'SCREW'  MAIL2 'NUT'
           DEP1 'DISPLACEMENTS' SIG1 'STRESSES' ;
```

READING / WRITING DATA

- **LIRE / SORT reading / writing different file formats**

Read/write the MED format
concerned objects: MAILLAGE, CHPOINT, TABLE (from PASAPAS)
used by Salomé, Europlexus

```
OPTI 'SORT' 'foo.med' ;
SORT 'MED' OBJET1 ;

TAB1 = LIRE 'MED' 'foo.med' ;
```

Read the FEM fomat
concerned objetcs: MAILLAGE
used by HyperMesh (Altair)

```
TAB1 = LIRE 'FEM' 'foo.fem' ;
```

DEVELOPMENT: GIBIANE PROCEDURES

- **Concatenate all procedures in a text file with the .procedur extension**

- **Launch castem22 with the –u option**
  `castem22 –u`

  ➔ Cast3M creates a file named UTILPROC (direct access file)

- **When Cast3M will be launched all procedures will be available**
  The UTILPROC file must be present in the working directory

Idem for notices (manual pages) (.notice extension files)

DEVELOPMENT: ESOPE SOURCES

The user can modify/correct/add the source code of operators and directives

- **Compilation of Esope source files**

```
compilcast22  toto.eso
compilcast22  tata.eso
...
```

- **Linking**

```
essaicast22
```

➔ creation of a binary executable file : **cast_64_22**
➔ local version of Cast3M

- **Can be launched as usual**

```
castem22  mon_fichier.dgibi
```

AND TO COMPLETE

- **Peruse documentation regularly**
  ~70 instructions reviewed during this course
  Around 1400 available instructions!

- **Subscription to the Cast3M mailing list (see the Cast3M web site)**
  Send an e-mail at sympa@umontpellier.fr with in the message frame:
  SUB cast3m-util your_name your_firstname

  and nothing more! (no object, no signature, …)

- **Club Cast3M: annual users seminar**
  Each year in November in the south of Paris
  Presentation of studies performed with Cast3M, developments in the next release
  Free registration!

PLANE LINEAR THERMAL ELASTICITY
(1) EQUATIONS

Total strains (linear) $\boldsymbol{\varepsilon} = \frac{1}{2}\left(\mathbf{grad}(u) + \mathbf{grad}^{\boldsymbol{T}}(u)\right)$ $\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$

Strains partition $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^{th}$

Thermal strains $\boldsymbol{\varepsilon}^{th} = \alpha \Delta T\, \boldsymbol{\delta}$

Elasticity $\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon}^e = \mathbb{C} : \left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{th}\right)$

with:

u	displacement
$\boldsymbol{\varepsilon}$	total strain
$\boldsymbol{\varepsilon}^e$	elastic strain
$\boldsymbol{\delta}$	identity tensor
$\mathbb{C}$	stiffness tensor (4th order)
α	thermal expansion
ΔT	temperature increment
$\boldsymbol{\sigma}$	stress
$\boldsymbol{\varepsilon}^{th}$	thermal strain

PLANE LINEAR THERMAL ELASTICITY (2) PLANE STRESSES

Plane strains hypothesis:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & 0 \\ \sigma_{12} & \sigma_{22} & 0 \\ 0 & 0 & \sigma_{33} \end{bmatrix} \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & 0 \\ \varepsilon_{12} & \varepsilon_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Vector notation:

$$\{\sigma\} = \begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{Bmatrix} \qquad \{\varepsilon\} = \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{Bmatrix}$$

with:$\sigma_{33} = \nu(\sigma_{11} + \sigma_{22})$
3 independent components

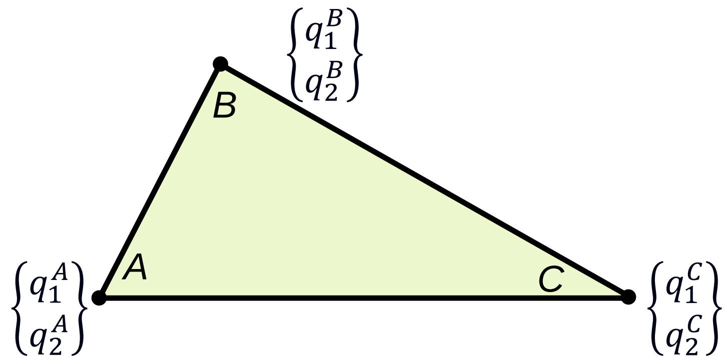
Hooke's law:

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{Bmatrix} = \underbrace{\frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}}_{[\boldsymbol{C}]} \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{Bmatrix}$$

$$\{\sigma\} = [\boldsymbol{C}]\{\varepsilon\}$$

$[\boldsymbol{C}]$ Hooke matrix

PLANE LINEAR THERMAL ELASTICITY (3) FINITE ELEMENTS



Interpolation of primal unknowns:

$$\{u(x)\} = [N(x)]\{q\}$$

at each point (x, y) in the element
$\{q\}$ are the displacements of the element nodes

Case of a TRI3 element (6 dof):

$$\begin{Bmatrix} u_x(x) \\ u_y(x) \end{Bmatrix} = \begin{bmatrix} N_A(x) & 0 & N_B(x) & 0 & N_C(x) & 0 \\ 0 & N_A(x) & 0 & N_B(x) & 0 & N_C(x) \end{bmatrix} \begin{Bmatrix} q_1^A \\ q_2^A \\ q_1^B \\ q_2^B \\ q_1^C \\ q_2^C \end{Bmatrix}$$

Strains (linearized):

$$\{\varepsilon(x)\} = \begin{bmatrix} N_{A,x_1} & 0 & N_{B,x_1} & 0 & N_{C,x_1} & 0 \\ 0 & N_{A,x_2} & 0 & N_{B,x_2} & 0 & N_{C,x_2} \\ N_{A,x_1} & N_{A,x_2} & N_{B,x_1} & N_{B,x_2} & N_{C,x_1} & N_{C,x_2} \end{bmatrix} \{q\}$$

with $N_{A,x_i} = \dfrac{\partial N_A(x)}{\partial x_i}$

$$\{\varepsilon(x)\} = [B(x)]\{q\}$$

$[B(x)]$ discrete gradient operator

PLANE LINEAR THERMAL ELASTICITY
(4) STIFFNESS MATRIX

Elementary elastic energy:

$$e_{def} = \frac{1}{2}\int_{V_e} \boldsymbol{\sigma} : \boldsymbol{\varepsilon}\, dV$$

$$= \frac{1}{2}\int_{V_e} \{\varepsilon\}^T\{\sigma\}\, dV$$

$$= \frac{1}{2}\int_{V_e} \{q\}^T[B]^T[C][B]\{q\}\, dV$$

$$= \frac{1}{2}\{q\}^T \underbrace{\left[\int_{V_e} [B]^T[C][B]\, dV\right]}_{[k_e]}\{q\}$$

V_e finite element "volume"
$[k_e]$ elementary stiffness matrix
(obtained by numerical computation)

Assembly:

$$E_{def} = \frac{1}{2}\{Q\}^T\sum_e [k_e]\{Q\} = \frac{1}{2}\{Q\}^T[K]\{Q\}$$

$\{Q\}$ nodes displacements (global mesh)
The $[k_e]$ matrices are extended on all dof with null items
$[K]$ assembly stiffness matrix

f_v prescribed volume forces
f_s prescribed surface forces

Principle of virtual work:

$$\int_{V_e} \boldsymbol{\sigma} : \boldsymbol{\delta\varepsilon}^* \, dV = \int_{V_e} f_v \delta u^* \, dV + \int_{S_e} f_s \delta u^* \, dS$$

$$\int_{V_e} \{\delta\varepsilon^*\}^T \{\sigma\} \, dV = \int_{V_e} \{\delta u^*\}^T \{f_v\} \, dV + \int_{S_e} \{\delta u^*\}^T \{f_s\} dS$$

$$\{\delta q^*\}^T \int_{V_e} [B]^T \{\sigma\} \, dV = \{\delta q^*\}^T \left(\int_{V_e} [N]^T \{f_v\} \, dV + \int_{S_e} [N]^T \{f_s\} dS \right)$$

$$\underbrace{\int_{V_e} [B]^T [C][B] \, dV}_{[k_e]} \{q\} = \underbrace{\int_{V_e} [N]^T \{f_v\} \, dV + \int_{S_e} [N]^T \{f_s\} dS}_{\{F_e\}}$$

$$[k_e]\{q\} = \{F_e\}$$

$\{F_e\}$ nodal equivalent forces

We add a second member item:

$$\int_{V_e} [B]^T \{\sigma\}\, dV = \{F_e\}$$

$$\int_{V_e} [B]^T [C] \{\varepsilon - \varepsilon^{th}\}\, dV = \{F_e\}$$

$$\int_{V_e} [B]^T [C] \{\varepsilon\}\, dV = \{F_e\} + \underbrace{\int_{V_e} [B]^T [C] \{\varepsilon^{th}\}\, dV}_{\{F_{th}\}}$$

$$[k_e]\{q\} = \{F_e\} + \{F_{th}\}$$

STIFFNESS AND LAGRANGE MULTIPLIERS

- Mechanics deals with the following constrained problem:

$$\begin{cases} [K]\{Q\} = \{F\} \\ [b]\{Q\} = \{Q_0\} \end{cases}$$

$[b]$ constraints matrix
$\{Q_0\}$ imposed values

- We have to minimize the following function:

$$f\{Q\} = \frac{1}{2}\{Q\}^T.[K].\{Q\} - \{Q\}^T.\{F\}$$

with the constrain: $[b]\{Q\} - \{Q_0\} = \{0\}$

- The Lagrange theorem introduces new unknowns $\{\lambda\}$ so that:

$$[K]\{Q\} - \{F\} + [b]^T\{\lambda\} = \{0\}$$

- We have to solve the following linear system:

$$\begin{bmatrix} K & b^T \\ b & 0 \end{bmatrix} \begin{Bmatrix} Q \\ \lambda \end{Bmatrix} = \begin{Bmatrix} F \\ Q_0 \end{Bmatrix}$$

STIFFNESS AND LAGRANGE MULTIPLIERS

Example: one bar element

```
OPTI 'DIME' 2 'ELEM' 'SEG2' ;

* Mesh
p1 = 0. 0. ;
p2 = 3. 0. ;
l1 = DROI 1 p1 p2 ;

* Model and properties
mo = MODE l1 'MECANIQUE' 'BARR' ;
ma = MATE mo 'YOUN' 210.E9 'NU' 0.3
             'SECT' 1.E-2 ;

* Main stiffness matrix
rig1 = RIGI mo ma ;
LIST rig1 ;

* Constraint matrix
bl1 = BLOQ 'UX' p1 ;
LIST bl1 ;

* Second member
f1 = DEPI bl1 1.2 ;
LIST f1 ;
```

matrices / primal unknowns / dual unknowns

$$\text{rig1} = \begin{bmatrix} k & 0 & -k & 0 \\ 0 & 0 & 0 & 0 \\ -k & 0 & k & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} UX_1 \\ UY_1 \\ UX_2 \\ UY_2 \end{bmatrix} \quad \begin{bmatrix} FX_1 \\ FY_1 \\ FX_2 \\ FY_2 \end{bmatrix}$$

$$\text{bl1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} LX_3 \\ UX_1 \end{bmatrix} \quad \begin{bmatrix} FLX_3 \\ FX_1 \end{bmatrix}$$

$$\text{rig1 ET bl1} = \left[\begin{array}{cccc:c} k+0 & 0 & -k & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -k & 0 & k & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \hdashline 1 & 0 & 0 & 0 & 0 \end{array}\right] \quad \left[\begin{array}{c} UX_1 \\ UY_1 \\ UX_2 \\ UY_2 \\ \hdashline LX_3 \end{array}\right] \quad \left[\begin{array}{c} FX_1 \\ FY_1 \\ FX_2 \\ FY_2 \\ \hdashline FLX_3 \end{array}\right]$$

$$FLX_3 = 1{,}2$$

lien 1 lien 2

ACCUMULATED PLASTIC STRAIN

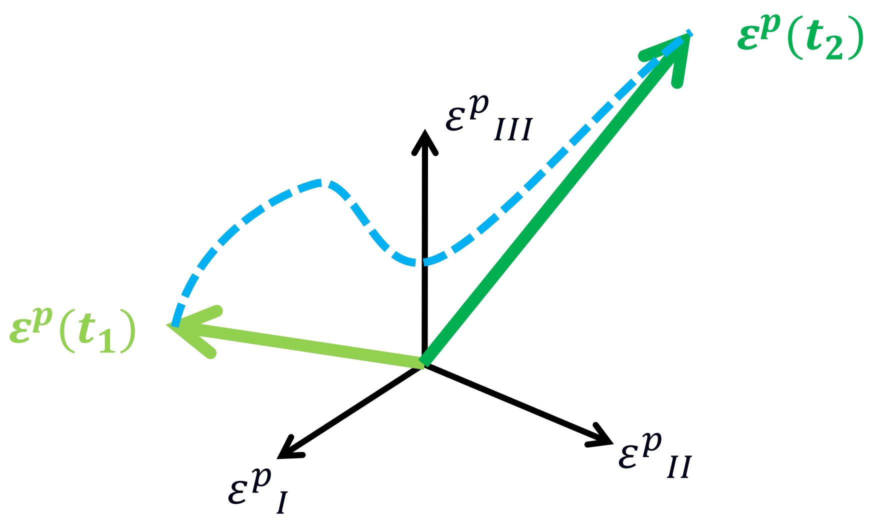
- Definition

$$p(t) = \int_0^t \dot{p}(\tau)d\tau \qquad \dot{p} = \sqrt{\frac{2}{3}\dot{\underline{\underline{\varepsilon}}}^p : \dot{\underline{\underline{\varepsilon}}}^p} = \sqrt{\frac{2}{3}\dot{\varepsilon}^p{}_{ij}\dot{\varepsilon}^p{}_{ij}}$$

- Meaning

Measure of the ***length of the flow trajectory*** *in the plastic strain space*

(link)



DESCRIPTION OF GIBIANE OBJECTS

OBJECTS DESCRIPTION

- General objects

ENTIER	Integer
FLOTTANT	Real
MOT	Characters string
LOGIQUE	Boolean (TRUE and FALSE)
LISTENTI	List of integers
LISTREEL	List of reals
LISTMOT	List of word (restricted to 4 digits)
TABLE	Set of objects of various types characterized by an index of various types
EVOLUTIO	Representation of a function, couple of lists (x ; f(x))

OBJECTS DESCRIPTION

- Objects for meshing

POINT	Coordinates of a point of the space + density
MAILLAGE	Region of the discretized space

- Objects for calculation

CHPOINT	Any data type defined at the mesh nodes (floating, Boolean, fields, …)
MMODEL	Association of a type of finite element and a material behavior with a mesh Defines the primal / dual unknowns *ex: displacements / forces* *temperature / heat flux*

OBJECTS DESCRIPTION

- Objects for calculation (continuation)

MCHAML	Any data type defined inside the mesh elements (floating, Boolean, fields, …). Various supports (Gauss points, center of gravity, nodes, …)
RIGIDITE	Stiffness, mass, conductivity matrix, … Coupling the physical unknowns
CHARGEME	Loading spatio-temporal description

- Objects for post processing

VECTEUR	To visualize multi component fields
DEFORMEE	To visualize a deformed mesh

François DI PAOLA
Commissariat à l'énergie atomique et aux énergies alternatives
Centre de Saclay | 91191 Gif-sur-Yvette Cedex

DEN/DANS
DM2S
SEMT