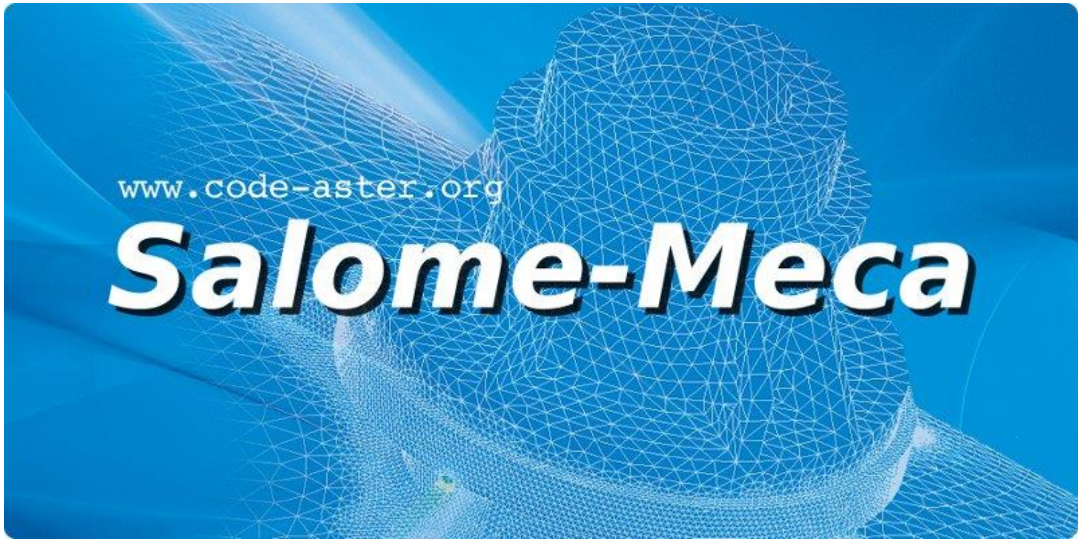
Thermal analysis

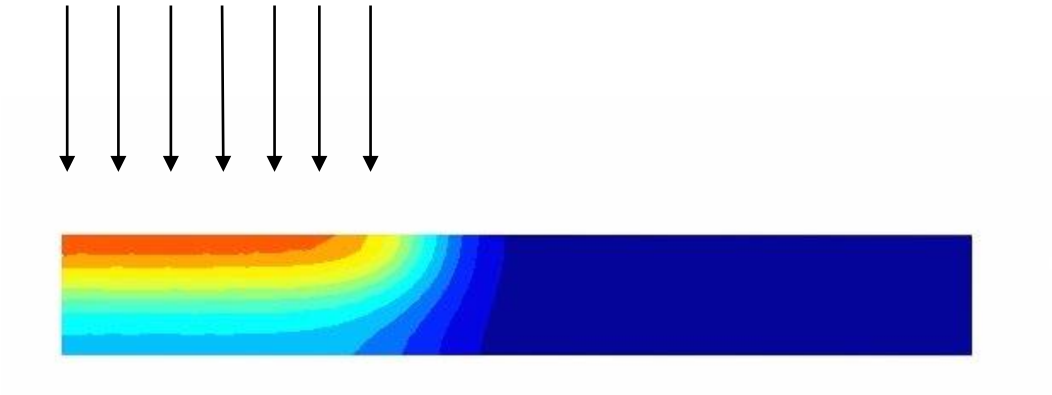


Code_Aster, Salome-Meca course material

Code_Aster : main features for thermics

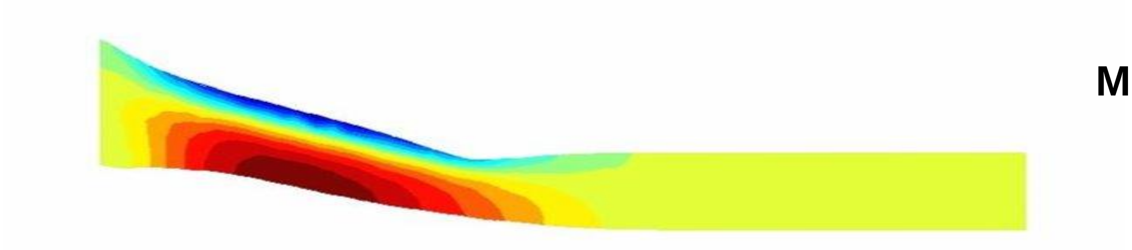
Frequently used as a prerequisite to a mechanical calculation

Example : laser heating of a metal disk



Thermic calculation

Laser heating of a disk : isothermal values

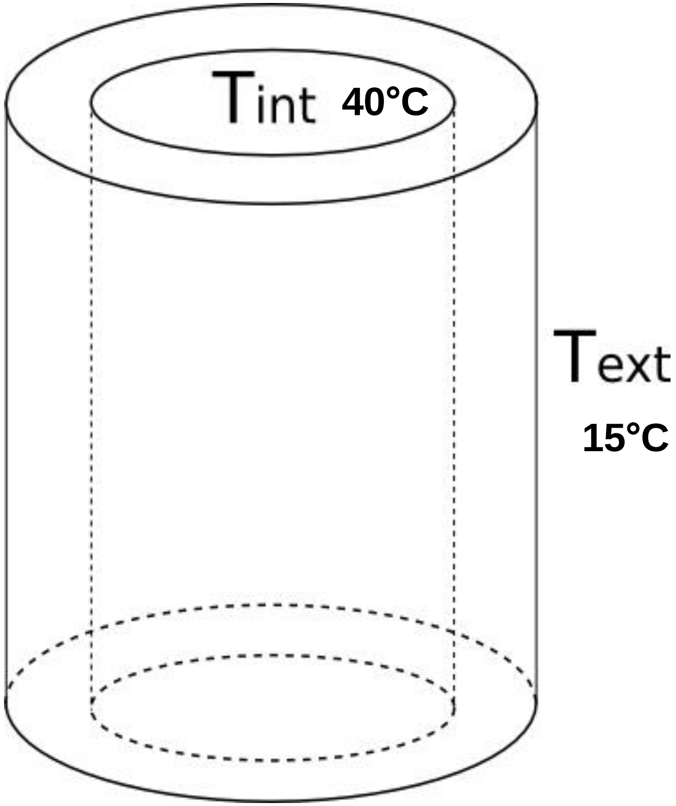


Mechanical calculation

Final circumferential isostresses on the deformed geometry

A simple example : pipe in thermal equilibrium

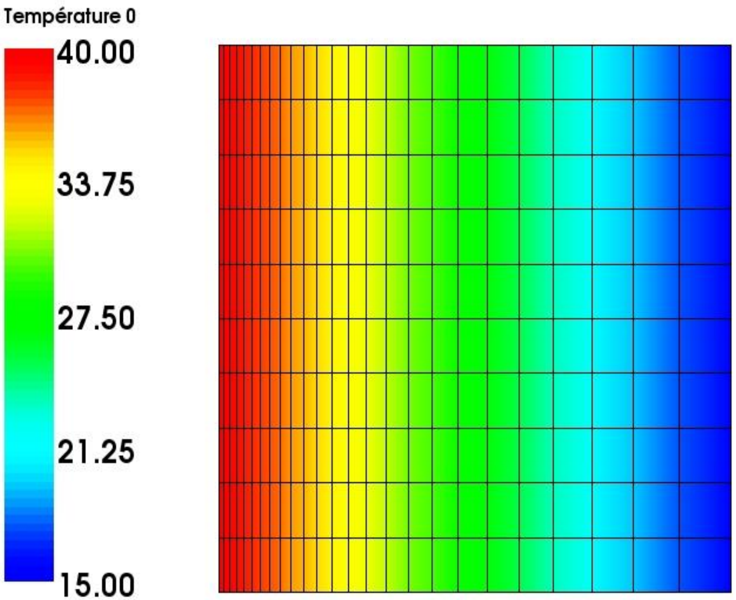
- Infinite pipe
- Load : a temperature field on each side



We want to know the temperature field in the thickness of the pipe

A simple example : pipe in thermal equilibrium

```
DEBUT();

mymesh=LIRE_MAILLAGE();

mymodl=AFFE_MODELE(     MAILLAGE= mymesh,
          AFFE=_F(      TOUT='OUI',
                        PHENOMENE='THERMIQUE',
                        MODELISATION='AXIS',),);

thmat =DEFI_MATERIAU(THER=_F(LAMBDA=6,),);

mymat=AFFE_MATERIAU(    MAILLAGE= mymesh,
          AFFE=_F(      TOUT='OUI',
                        MATER=thmat,),);

myload=AFFE_CHAR_THER( MODELE= mymodl,
          TEMP_IMPO=(_F(GROUP_NO='SEXTERI',
                                   TEMP=15,),
                                 _F(GROUP_NO='SINTERI',
                                   TEMP=40,),),);

resuth=THER_LINEAIRE(   MODELE= mymodl,
          CHAM_MATER= mymat,
          EXCIT=_F(CHARGE= myload,),);

IMPR_RESU(              FORMAT='MED',
          RESU=_F(RESULTAT= resuth,),);

FIN();
```



What is the problem to be solved?

```
AFFE_MODELE ( ...PHENOMENE = 'THERMIQUE'... )
```

Heat equation : $$\rho C\dot{T} - \lambda \Delta T - s = 0$$

Linear (`THER_LINEAIRE`)

OR

Non-linear (`THER_NON_LINE`)

If : Material parameters depend on T

AND/OR

Non linear boundary conditions (flux(T))

Temperature dependence?

AND

Stationnary

OR

Transient

Time dependence?

Available finite elements

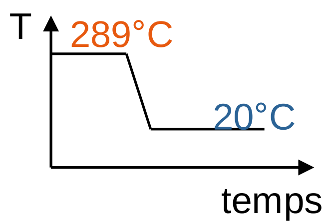
```
AFFE_MODELE (     … PHENOMENE    = 'THERMIQUE'
                  … MODELISATION = 'XXXX' ...) )
```

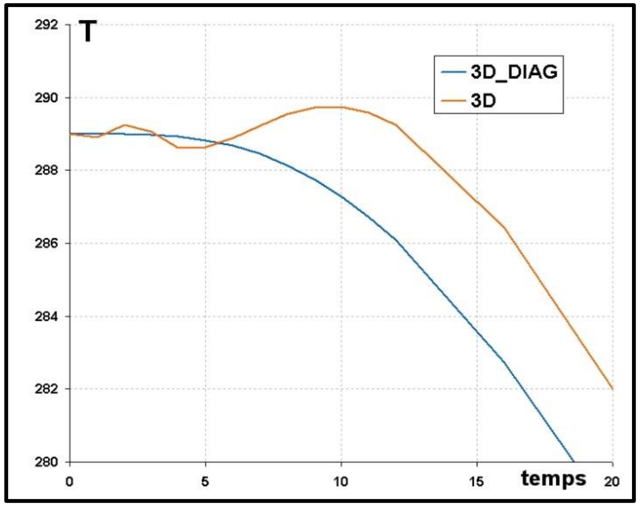
3D / AXIS / PLAN

3D_DIAG / AXIS_DIAG / PLAN_DIAG

- regularization during heat shock
- splitting of quadratic elements in 2D
- splitting not avalaible in 3D

Example of a sudden cooling :





COQUE / COQUE_PLAN / COQUE_AXIS (linear)

- thermal shell "thin" structural elements
- temperature field in the thickness : 3 degrees of freedom (Tsup, Tinf and Tmiddle)

Material behavior

DEFI_MATERIAU

At least these two characteristics:

Thermal conductivity	λ
Heat capacity	ρC_P

3 main materials :

THER	Linear isotropic
THER_ORTH	Linear orthotropic (definition of λ in 3 directions)
THER_NL	Non-linear behaviour : one has to define $\rho C_P(T)$ or $\beta(T)$ and $\lambda(T)$

+ other specific materials

- For concrete : drying, hydratation ... etc
- For metals : metallurgical transformations, hardness ... etc

Boundary conditions and loadings (1)

- **AFFE_CHAR_THER (...)**
- **AFFE_CHAR_THER_F (...)**
- Boundary conditions (Dirichlet)

Imposed temperatures function of time and space	**TEMP_IMPO**
Linear relationships between the nodal temperatures	**LIAISON_DDL** **LIAISON_GROUP** **LIAISON_MAIL**

Boundary conditions and loadings (2)

- Loadings (Neumann)

Natural convection (Fourier law)	**ECHANGE**	$\lambda(T)\dfrac{dT}{dn} = h(t)\cdot(Text - T)$
Heat exchange between walls	**ECHANGE_PAROI**	$\lambda_1 \dfrac{dT_1}{dn_1} = h(T_2 - T_1)$
Normal imposed flux : constant or function of time and space	**FLUX_REP**	$\lambda(T)\dfrac{dT}{dn} = f(t,x)$
Non linear normal flux : function of the temperature Non-linear only	**FLUX_NL** **RAYONNEMENT**	$\lambda(T)\dfrac{dT}{dn} = f(T)$
Heat source	**SOURCE**	$s(x,t)$

Is the problem well defined?

- For a stationnary calculation : **YES**

- For a transient calculation : **NO**
  - **A condition is missing : the temperature at the initial time**

- Defined in **THER_LINEAIRE** or **THER_NON_LINE** with **ETAT_INIT**

4 possibilities :

Résults of a stationnary calculation	**STATIONNAIRE ='OUI'**
A constant temperature	**VALE = T**$_0$
A known temperature field	**CHAM_NO = …** for example, created with **CREA_CHAMP**
The result of another thermal calculation	A result concept (**EVOL_THER = resu**) + chosen time (**NUME_ORDRE** or **INST**)

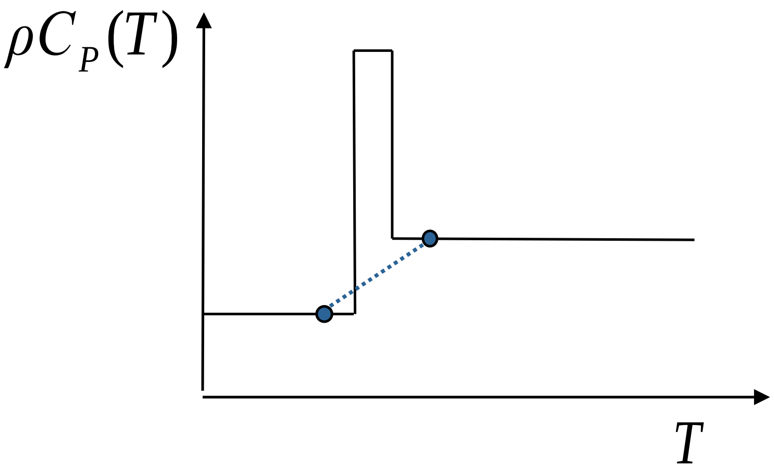
Solving methods

- Linear thermics : **THER_LINEAIRE**

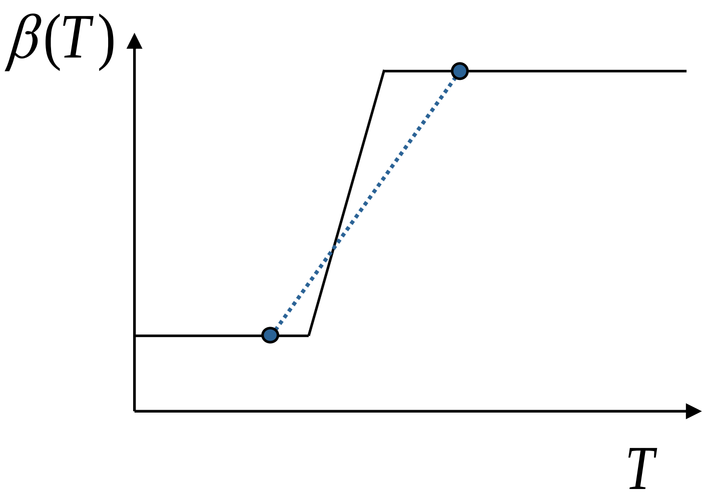
- Non-linear thermics : **THER_NON_LINE**

  Newton method for non-linéarities

  Enthalpy formulation $\beta(T)$



Numerically easier to handle phase changes



Precautions for use in thermics

- Time discretization with the θ-méthode.

$$\frac{\rho C}{\lambda . \Delta t}(T_{n+1} - T_n) = \theta \Delta T_{n+1} + (1-\theta)\Delta T_n$$

  - θ = 0.57, good compromise and the default value
  - Ability to change θ with **PARM_THETA**

- Beware to inconsistent refinements in time and space.
  - Respect the time and distance characteristics of the material

$$\Delta t \leq \frac{\rho C (\Delta x)^2}{(6\theta)\lambda}$$

- Drive the non-linear convergence, if necessary.
  - Frequency of updating the tangent matrix
  - Linear search criteria

  } **NEWTON**

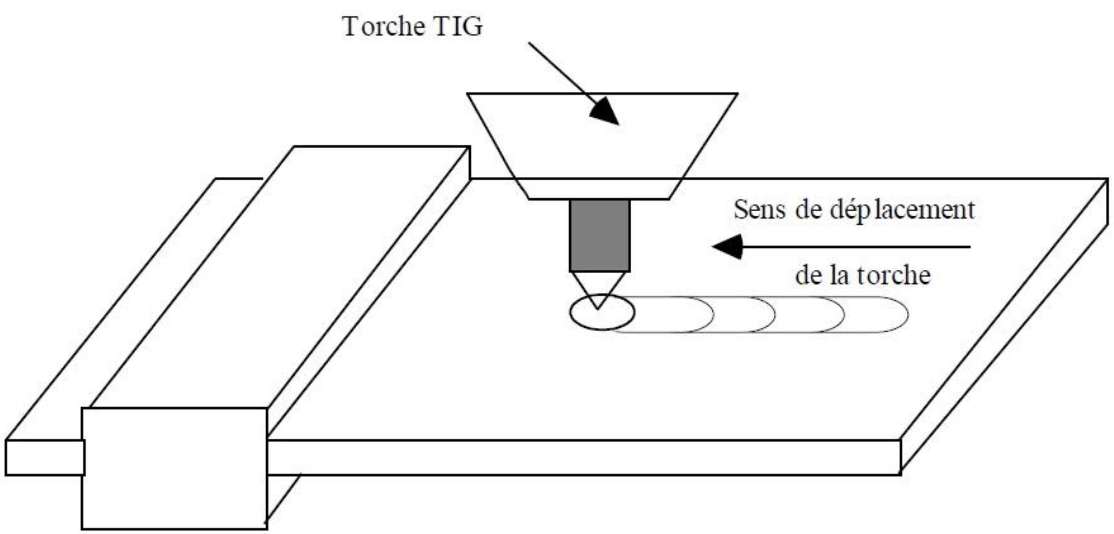
  - Convergence criteria (number of iterations, residues ...) **CONVERGENCE**

Post-processing options

- What's the result of **THER_LINEAIRE** / **THER_NON_LINE** ?
  - By default : only the nodal temperatures
- How to get the thermal fluxes ?
  - With **CALC_CHAMP**
    - **FLUX_ELGA**
    - **FLUX_ELNO**
    - **FLUX_NOEU**

Metallurgy

- Possibilities :
  - Compute in moving frame (welding) : **THER_NON_LINE_MO**
  - With forced convection (welding) **AFFE_CHAR_THER (...CONVECTION...)**
  - Calculate the hardness, the metallurgical transformations **CALC_META**

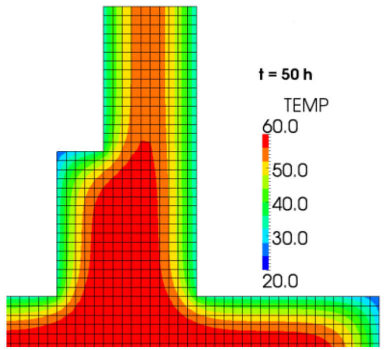


Thermo-hydration and drying

- Common use : chained calculation :

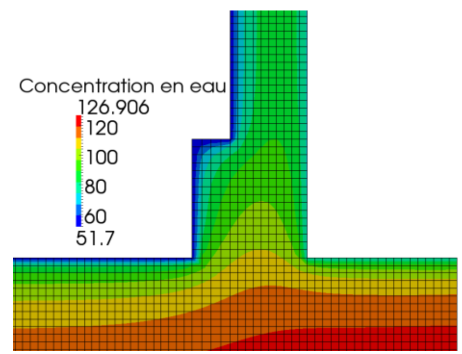
Thermic and hydration calculation of concrete

Temperature field



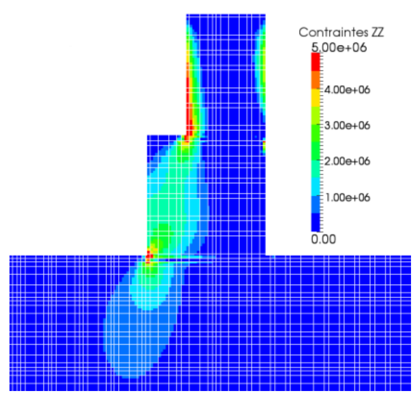
Hydration field

Calculation of concrete drying



Water concentration field

Mechanical calculation of concrete shrinkage



Displacements, strains …

Stresses field

And mechanics?

- How to take into account the temperature field in the mechanical calculation?

In **AFFE_MATERIAU** (...

```
                    AFFE_VARC=_F( NOM_VARC   = 'TEMP',
                          EVOL   = EVOTH,
                          VALE_REF       = 20. )
                                    ...)
```

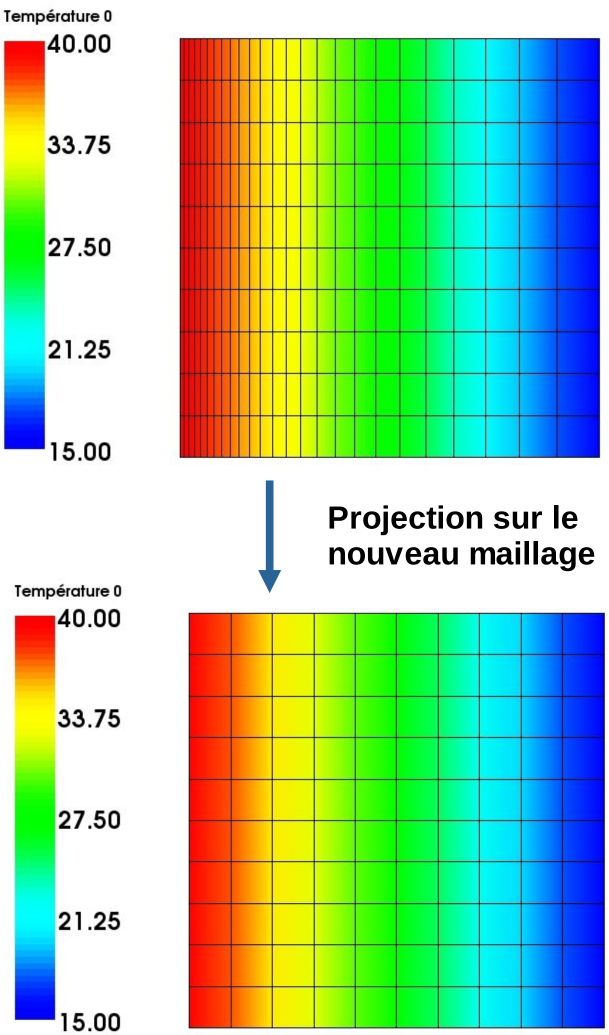
- The mesh to be used in mechanics must be different from that used for thermics. For two main reasons:
  - A linear mesh is better in thermics, while rather quadratic in mechanics
  - Areas of interest are different => the refined areas are different

- One must project the temperature field on the mesh used for mechanics
  - **PROJ_CHAMP (…)**

And mechanics?

Example of the cylinder

```
DEBUT()
thmesh=LIRE_MAILLAGE()
thmodl=AFFE_MODELE(...)
thmater=DEFI_MATERIAU(...)
fthmater=AFFE_MATERIAU(...)
thload=AFFE_CHAR_THER(...)
thresu=THER_LINEAIRE(...)
```



We want to make a mechanical calculation with a different mesh

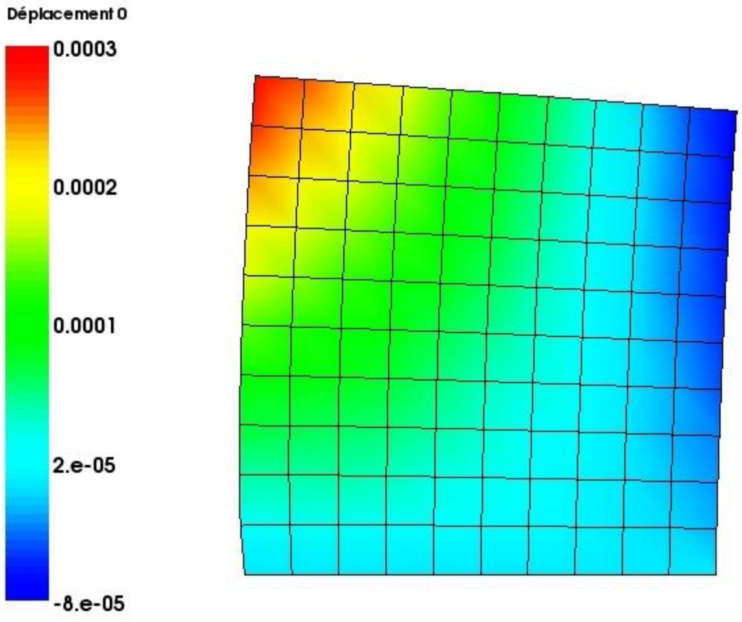
```
memesh=LIRE_MAILLAGE()
thmodl2=AFFE_MODELE(MAILLAGE= memesh,
          AFFE=_F(  TOUT='OUI',
                       PHENOMENE='THERMIQUE',
                       MODELISATION='AXIS',),)
resproj=PROJ_CHAMP( RESULTAT= thresu,
          MODELE_1= thmodl,
          MODELE_2= thmodl2,)
...
```

And mechanics?

Mechanical calculation of the expansion of the cylinder

```
mymodl=AFFE_MODELE(  MAILLAGE=mymesh,
          AFFE=_F(    TOUT='OUI',
                      PHENOMENE='MECANIQUE',
                      MODELISATION='AXIS',),)

steel=DEFI_MATERIAU( ELAS=_F( E=2.1e11,
                      NU=0.2,
                      ALPHA=12e-6,),)

mymat=AFFE_MATERIAU( MAILLAGE=mymesh,
          AFFE=_F(    TOUT='OUI',
                      MATER=steel,),
          AFFE_VARC=_F(TOUT='OUI',
                      NOM_VARC='TEMP',
                      EVOL=projres,
                      VALE_REF=20,),)

mecload=AFFE_CHAR_MECA(MODELE=MODME,
          DDL_IMPO=_F(GROUP_NO='FACEB',
                      DX=0,
                      DY=0,),)

mecres=MECA_STATIQUE(MODELE=mymodl,
          CHAM_MATER=mymat,
          EXCIT=_F(CHARGE= mecload,),)

IMPR_RESU(...)

FIN()
```

Temperature field
+ reference value



Analytical temperature field

Here we know the analytical formula of the temperature field in the thickness of the cylinder (X):

$$T_{inte} - \ln\left(\frac{X}{R_{inte}}\right)\frac{T_{exte} - T_{inte}}{\ln\left(\frac{R_{exte}}{R_{inte}}\right)}$$

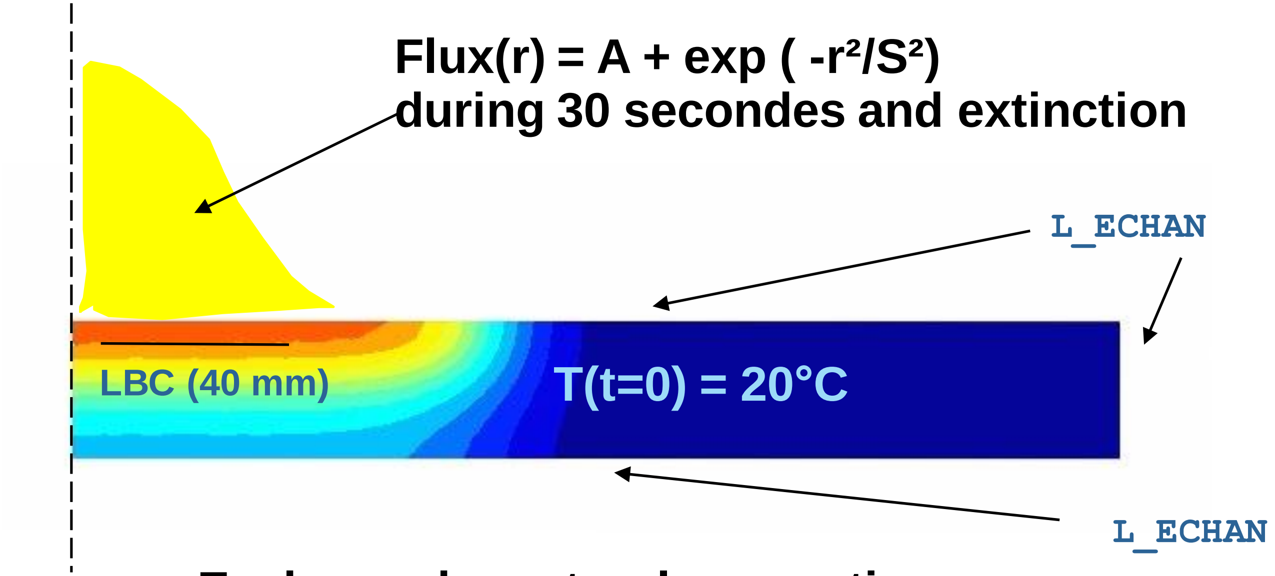
```
DEBUT();
// Thermic calculation
thmesh=LIRE_MAILLAGE()
thmodl=AFFE_MODELE(...)
thmater=DEFI_MATERIAU(...)
fthmater=AFFE_MATERIAU(...)
thload=AFFE_CHAR_THER(...)
thresu=THER_LINEAIRE(...)
// Projection
memesh=LIRE_MAILLAGE()
thmodl2=AFFE_MODELE(...)
resproj=PROJ_CHAMP(...)
```

```
memesh=LIRE_MAILLAGE()

T0 = FORMULE(VALE='40-log(X/20)*(40-15)/log(21/20)',
             NOM_PARA=('X',),)
fT0 =CREA_CHAMP(         TYPE_CHAM='NOEU_TEMP_F',
             OPERATION='AFFE',
             MAILLAGE= memesh,
             AFFE=_F(    TOUT='OUI',
                         NOM_CMP='TEMP',
                         VALE_F=T0,),)

thresu=CREA_RESU(OPERATION='AFFE',
TYPE_RESU='EVOL_THER',
NOM_CHAM='TEMP',
AFFE=_F(CHAM_GD=fT0,
INST=0,),)
```

```
// Mechanic calculation
memodl=AFFE_MODELE(...)
steel=DEFI_MATERIAU(...)
memat=AFFE_MATERIAU(  MAILLAGE=memesh,
           AFFE=_F(   TOUT='OUI',
                      MATER=steel,),
           AFFE_VARC=_F(TOUT='OUI',
                      NOM_VARC='TEMP',
                      EVOL= resproj,
                      VALE_REF=20,),)
meload=AFFE_CHAR_MECA(...)
meresu=MECA_STATIQUE(...)
IMPR_RESU(...)
FIN()
```

Example of use

Laser treatment of a disc in linear thermics



Exchange by natural convection :

Text = 20°C

h = 5 W/m²°C

Example of use (2)

Model and material definitions

```
DEBUT( )
mymesh  =  LIRE_MAILLAGE  ( )
```

 # Definition of the model

```
mymodl =  AFFE_MODELE(MAILLAGE = mymesh ,
                  AFFE = _F(       TOUT = 'OUI',
                      PHENOMENE = 'THERMIQUE',
                      MODELISATION = 'AXIS',  )   )
```

 # Definition of the thermal properties of the material

```
steel =  DEFI_MATERIAU(  THER = _F (   RHO_CP  = 3.6e6,
                                  LAMBDA  = 14,  )    )
```

 # ssignment of the material on the finite element model

```
mymat= AFFE_MATERIAU(MAILLAGE= mymesh,
                  AFFE=_F( TOUT = 'OUI',
                  MATER = steel ,   )   )
```

Example of use (3)

Definition of data needed for the loadings definition

 # "mathematical" definition of the *F(R)* formula

```
A = 5e6
S = 0.0138
F = FORMULE(NOM_PARA='X',VALE="(A + EXP(-X**2/S**2))")
```

 # Interpretation of the *F(R)* formula and transformation in a function

 # (tabulation X ; *F*(X))

 # Definition of the abscissa for tabulation

```
LISTR=DEFI_LIST_REEL(DEBUT  =  0.,
         INTERVALLE=_F( JUSQU_A = 0.04, NOMBRE = 100, ) ,)
```

 # Calculation (and tabulation) of the formula for the *'LISTR'* values of the radius

```
FF = CALC_FONC_INTERP(   FONCTION      = F,
                 LIST_PARA             = LISTR,
                 PROL_DROITE           ='LINEAIRE',
                 PROL_GAUCHE           ='LINEAIRE'       )
```

Creation of a flux function of X (=R) in order to represent the heat due to the laser

 # Temperature of the outside

```
T_EXT   =   DEFI_CONSTANTE ( VALE=20., )
```

 # Coefficient for exchange by convection

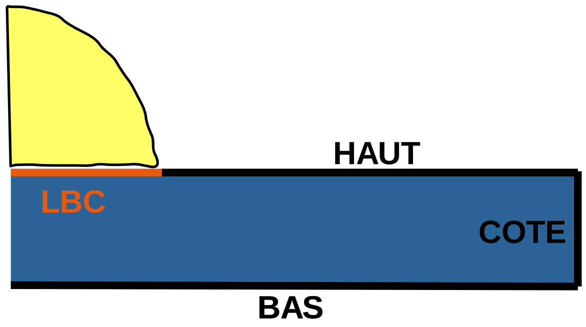
```
H   =   DEFI_CONSTANTE(VALE=5 ,)
```

Example of use (4)

Definition of two loadings

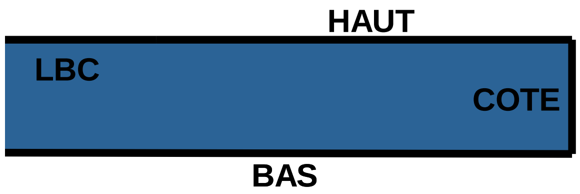
 # Exchange condition on the edge (during heating)

```
thload1  =  AFFE_CHAR_THER_F(  MODELE=mymodl,
FLUX_REP =_F( GROUP_MA = 'LBC', FLUN = FF,),
ECHANGE =_F( GROUP_MA = ('BAS','HAUT','COTE',),
COEF_H              = H,
TEMP_EXT  = T_EXT,   )    )
```



 # Exchange condition on the edge (during the free cooling) : convection everywhere

```
thload2 =  AFFE_CHAR_THER_F ( MODELE = mymodl,
ECHANGE=_F( GROUP_MA =('LBC','HAUT','COTE','BAS',),
COEF_H              = H ,
TEMP_EXT  = T_EXT , ),)
```



Example of use (5)

Calculation

 # Definition of time steps

```
LISTT  =  DEFI_LIST_REEL(  DEBUT=0.,  INTERVALLE=_F( JUSQU_A = 150., NOMBRE = 300, )  )
```

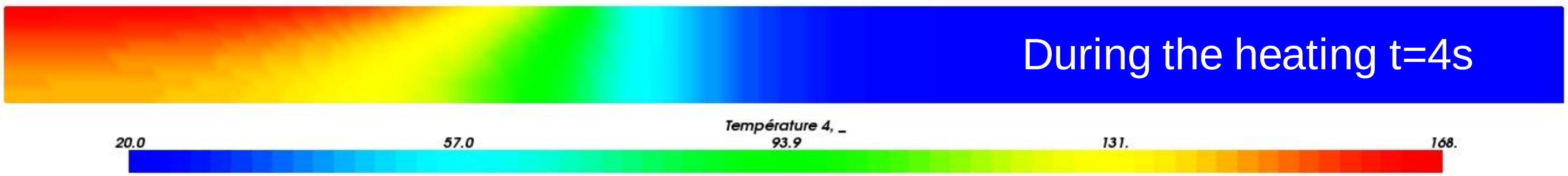
 # Calculation for the heating phase

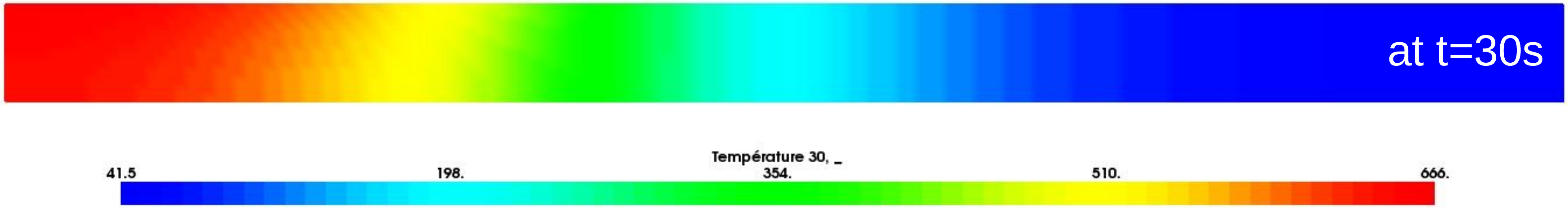
```
temper =THER_LINEAIRE(  MODELE= mymodl,
                     CHAM_MATER = mymat,
                     EXCIT           = _F( CHARGE = thload1, ),
                     INCREMENT  = _F(  LIST_INST = LISTT,
                                       INST_FIN  = 30,    ),      → Stop at t=30s
                     ETAT_INIT  = _F(  VALE = 20.0,   ) )   → at t=0s : T=20°C everywhere
```

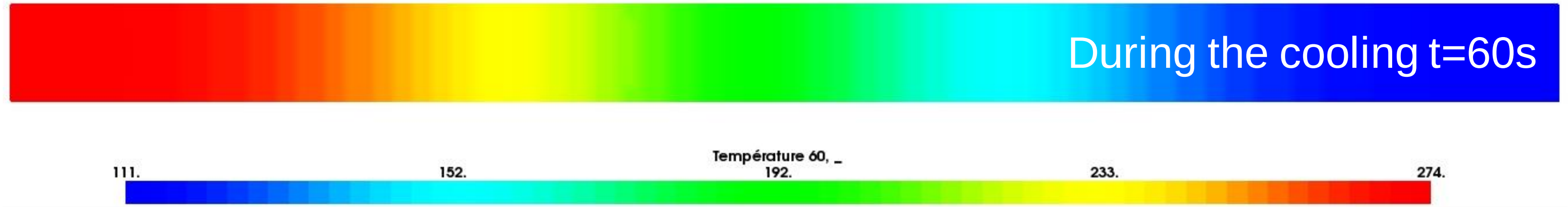
 # Calculation for the cooling phase

```
temper = THER_LINEAIRE( reuse         = temper ,
                     MODELE          = mymodl,
                     CHAM_MATER = mymat,
                     PARM_THETA = 0.60,
                     EXCIT           = _F(  CHARGE = thload2, ),
                     INCREMENT  = _F(LIST_INST = LISTT ,       → at t=30s : T= T° at the end
                                     INST_INIT = 30,),           of the previous calculation
                     ETAT_INIT  = _F(EVOL_THER = temper ,))
IMPR_RESU(  ...  )
FIN  ( )
```

Results


During the heating t=4s
Température 4, _
20.0
57.0
93.9
131.
168.


at t=30s
Température 30, _
41.5
198.
354.
510.
666.


During the cooling t=60s
Température 60, _
111.
152.
192.
233.
274.

End of presentation

Is something missing or unclear in this document?
Or feeling happy to have read such a clear tutorial?

Please, we welcome any feedbacks about Code_Aster training materials.
Do not hesitate to share with us your comments on the Code_Aster forum dedicated thread.