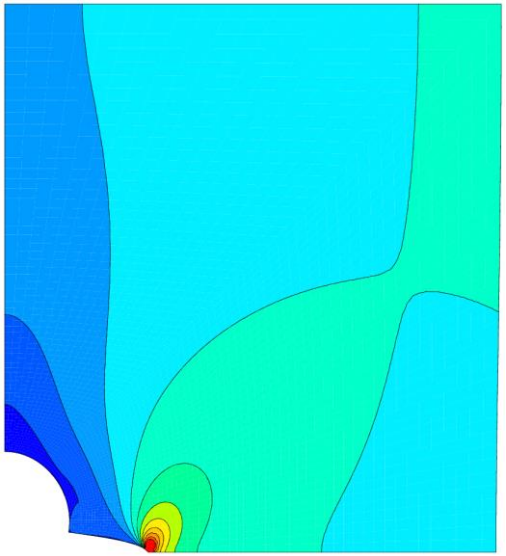
DE LA RECHERCHE À L'INDUSTRIE

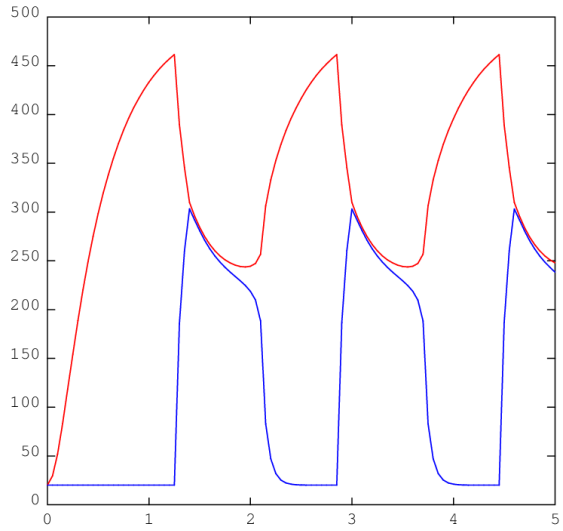
www.cea.fr

THE PASAPAS PROCEDURE

AND THE USERS PROCEDURES

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





François DI PAOLA, Caroline GUERIN

LAST CHANGES: NOVEMBER 30 2022

SUMMARY

- Introduction to Cast3M and reminders on PASAPAS
- PASAPAS operation
- Finding the mechanical equilibrium: UNPAS
  - Exercise 1: following load
  - Exercise 2: failure by elements removal
- Finding the thermal equilibrium: TRANSNON
  - Exercise 3: variable heat source
  - Exercise 4: thermo-mechanical contact

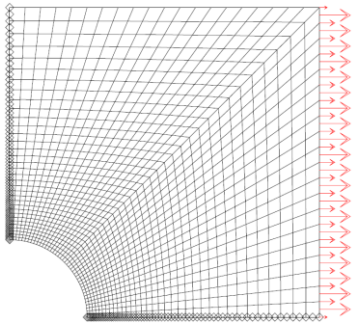
INTRODUCTION TO CAST3M

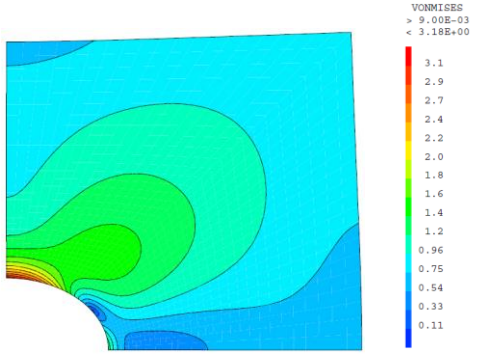
REMINDERS ON PASAPAS

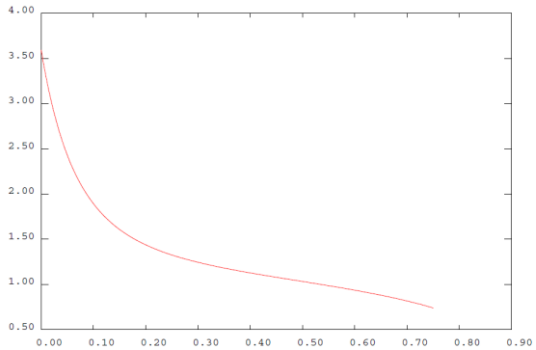
CAST3M, QUID ?

A **finite element** software for **structural** and **fluid mechanics**

- **Partial differential equations** solved with the finite element method
- **Complete software:** solver, pre-processing and post-processing, visualization



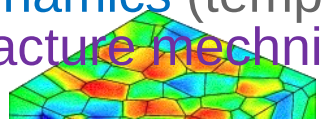




- Based on an objet-oriented programming language: **Gibiane**

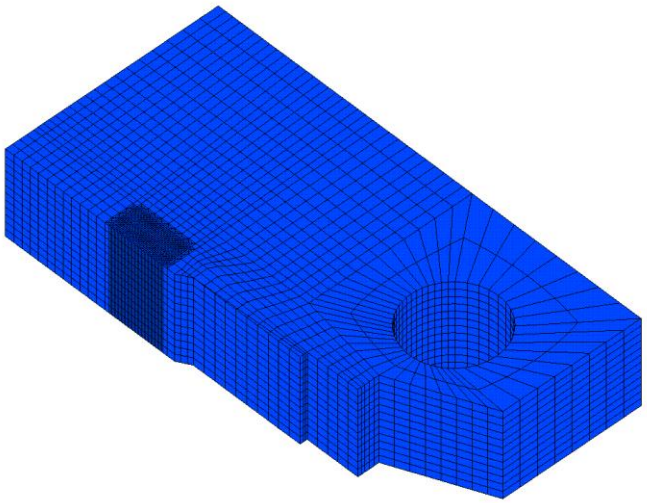
MODELING IN NUMEROUS AREAS

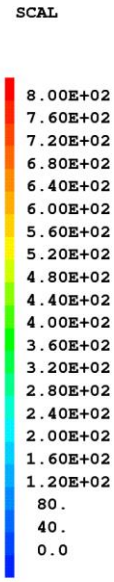
- **Structural mechanics (historical field)**
  Quasi-static (non linear materials, geometry, boundary conditions)
  Contact/friction, Buckling
  Dynamics (temporal, modal, fluid structure interaction)
  Fracture mechnics (XFEM, dynamic propagation, cohesive zone model)



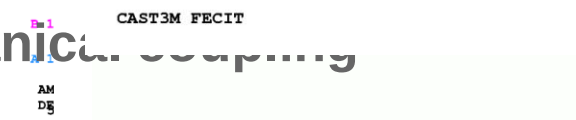
- **Thermal analysis**
  Conduction, convection, ra

- **Fluid mechanics**
- **Metallurgy**
- **Magnetostatics**
- **Multi spieces diffusion (F**





- **Thermo-hydro-mechanical coupling**



Temperature au temps 0.00000E+00 (0 %)

Proportions de BAIN au temps : 0.00000E+00

THE PASAPAS PROCEDURE

- **Objective**
  incremental solving of non linear **progressive**
  thermal and **mechanical** problems

  time can be physical (e.g. thermal transients)
  or not (e.g. plasticity with progressive loading)

  ➔ *time or pseudo-time is called the **evolution parameter***

- **Non linear phenomena considered**
  behavior (plasticity, damage, variable material properties, …)
  geometry (large displacements)
  strains (large rotations)
  boundary conditions (radiation, friction, following pressure, …)

PASAPAS USE

- **Create a table** containing all the data:

```
TAB1                          = TABL ;
TAB1 . MODELE                 = MOD1 ET MOD2 ;
TAB1 . CARACTERISTIQUES       = MAT1 ET MAT2 ;
TAB1 . BLOCAGES_MECANIQUES    = BLO1 ;
TAB1 . CHARGEMENT             = CHA1 ET CHA2 ET CHA3 ;
TAB1 . TEMPS_CALCULES         = PROG 0.1 'PAS' 0.1 50. ;
TAB1 . TEMPS_SAUVES           = PROG 4. 8. 15. 16. 23. 42. ;
TAB1 . PRECISION              = 1.E-6 ;
TAB1 . GRANDS_DEPLACEMENTS    = VRAI ;
…
```

some indexes are required
other are optional

- **Procedure call:**

```
PASAPAS TAB1 ;
```

- **Results post-processing**

OVERVIEW OF INPUT PARAMETERS

General

MODELE (MMODEL) — *Equations to solve, F. E. formulation* **(MODE)**

CARACTERISTIQUES (MCHAML) — *Material and/or geometrical parameters* **(MATE)**

CHARGEMENT (CHARGEME) — *Boundary conditions and loading variation during calculation* **(CHAR)**

Thermal analysis

BLOCAGES_THERMIQUES (RIGIDITE) — *Stiffness matrix associated to the imposed values of the unknowns (*DIRICHLET boundary conditions) **(BLOQ)**

CELSIUS (LOGIQUE) — *=VRAI (true) if temperature unit is* CELSIUS

TEMPERATURES . 0 (CHPOINT) — *Initial conditions*

Mechanics

BLOCAGES_MECANIQUES (RIGIDITE) — *Stiffness matrix associated to the imposed values of the unknowns (*DIRICHLET boundary conditions) **(BLOQ)**

GRANDS_DEPLACEMENTS (LOGIQUE) — *Equilibrium checked on the deformed mesh*

DEPLACEMENTS . 0 (CHPOINT)

CONTRAINTES . 0 (MCHAML)

VARIABLES_INTERNES . 0 (MCHAML)

DEFORMATIONS_INELASTIQUES . 0 (MCHAML)

} *Initial conditions*

OVERVIEW OF INPUT PARAMETERS

Mechanics (dynamics)

DYNAMIQUE (LOGIQUE) — =VRAI (true) *for dynamics calculations*

AMORTISSEMENT (RIGIDITE) — *Damping matrix*

VITESSES . 0 (CHPOINT)

ACCELERATIONS . 0 (CHPOINT)

} *Initial conditions*

List of the values of evolution parameter

TEMPS_CALCULES (LISTREEL) — *List of the values of the evolution parameter (or time) for which results are requested*

TEMPS_SAUVES (LISTREEL) — *List of times for which results are saved*

```
OPTI 'SAUV' 'my_file' ;
```

TEMPS_SAUVEGARDES (LISTREEL) — *List of times for which* **SAUV** *operator is called*

MES_SAUVEGARDES (TABLE) — *Results to be stored in addition to those automatically stored (total strains, …)*

OVERVIEW OF OUTPUT PARAMETERS

Results are stored in the table

TEMPS (TABLE) — *Times values, corresponding to « **TEMPS_SAUVES** »*

TEMPERATURES (TABLE)
PROPORTIONS_PHASE (TABLE)

DEPLACEMENTS (TABLE)
CONTRAINTES (TABLE)
DEFORMATIONS_INELASTIQUES (TABLE)
VARIABLES_INTERNES (TABLE)
REACTIONS (TABLE)
VITESSES (TABLE)
ACCELERATIONS (TABLE)

} *Fields (solution) calculated for each stored time « **TEMPS_SAUVES** »*

POST PROCESSING (EXAMPLES)

- **Solution fields extraction:**
  from the table index

```
SIG1 = TAB1 . CONTRAINTES . 5 ;
```

  or from the calculation time

```
SIG1 = PECHE TAB1 'CONTRAINTES' 28.3 ;
```

- **Graphical mode, interactive plot (limited):**

```
EXPLORER TAB1 ;
```

- **Evolution of calculated field with time:**

```
EV1 = EVOL 'TEMP' TAB1 'DEPLACEMENTS' 'UX' P1 ;
```

PASAPAS OPERATION

PASAPAS OPERATION

- **Main algorithm**

Initializations

Loop on time steps (BEXTERN)
- Loop for thermo-mechanical convergence (BO_BOTH)
  - **Thermal solving [for the step]**
  - **Mechanical solving [for the step]**
  - Thermo-mechanical convergence?
- Results recording

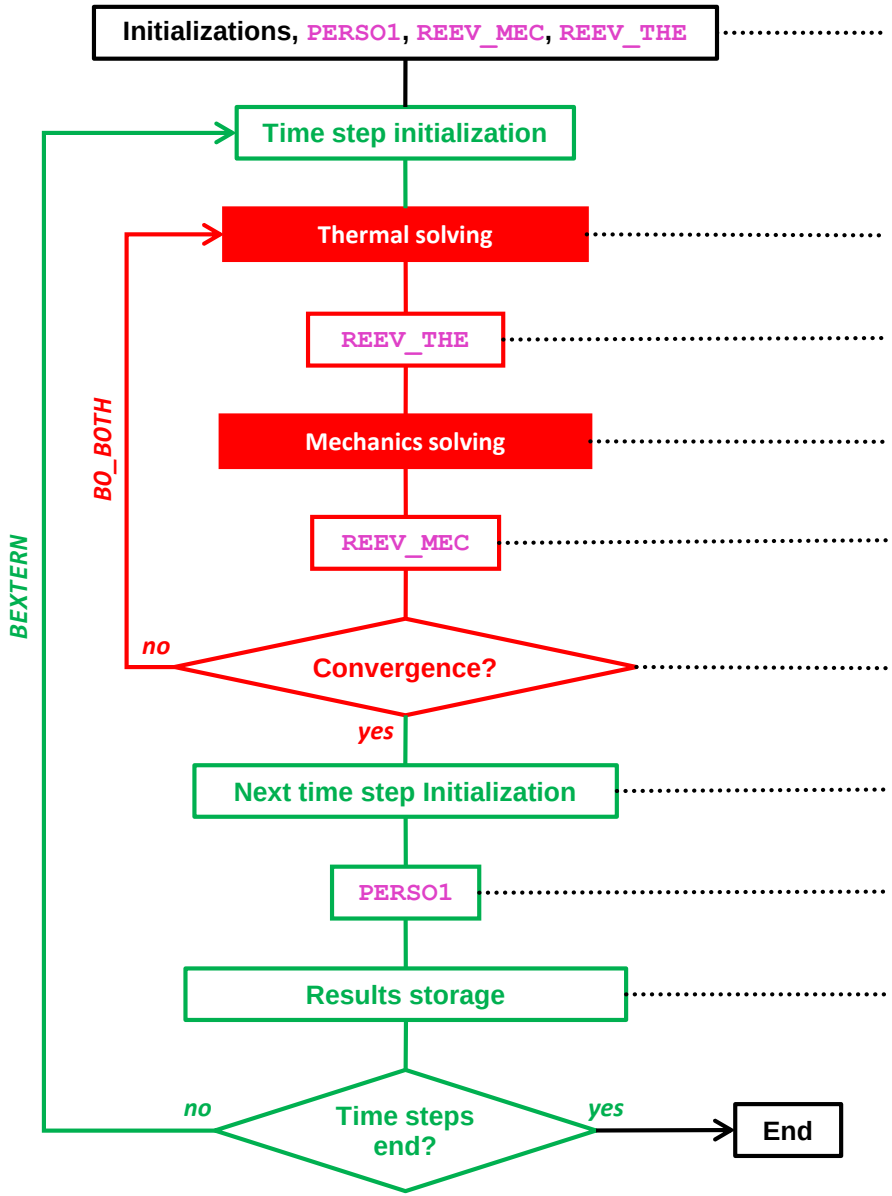
End

+ User procedures calls (optional, user have to ask it)

PERSO1 PERSO2 REEV_MEC REEV_THE
CHARMECA CHARTHER PARATHER

procedures to be defined by the user!

PASAPAS OPERATION



Options reading, initializations (materials, unknowns, temporary variables, indicators…), calculation times **WTABLE**, …

Call of the procedure for thermal calculation:
TRANSNON, TRANSLIN *or* DUPONT2
(in the case of large displacements, solving on the last deformed shape)
Results are stored in **ESTIMATION**

User procedure (if asked)

Call to the procedure for mechanical calculation **UNPAS**
Results are stored in **ESTIMATION**

User procedure (if asked)

Test of convergence for thermo-mechanics, to activate with:
TAB1 . CONVERGENCE_MEC_THE = VRAI ;

$[\text{MAXI ABS } (T_i - T_{i-1})] \;/\; [\text{MAXI ABS } T_i] < \text{CRITERE\_COHERENCE } (10E\text{-}2)$

Recording of the new state as the initial time step state in WTABLE

User procedure (if asked)

PAS_RESU: *writing in* **CONTINUATION** *quantities of* **ESTIMATION**
PAS_SAUV: *writing in* TAB1 *if the time step belongs to* TEMPS_SAUVE,
call to SAUV *if the time step belongs to* TEMPS_SAUVEGARDE

ACCESS TO PASAPAS DATA

- **TAB1 . 'ESTIMATION'**
  contains all results calculated/converged by TRANSNON and UNPAS
  but not converged for BO_BOTH loop

```
TAB1 . 'ESTIMATION' . 'TEMPS'
TAB1 . 'ESTIMATION' . 'DEPLACEMENTS'
TAB1 . 'ESTIMATION' . 'CONTRAINTES'
TAB1 . 'ESTIMATION' . 'TEMPERATURES'
…
…
```

- **TAB1 . 'CONTINUATION'**
  contains converged results (for BO_BOTH loop)
  this index is updated at the **end of the time-step!!!**
  ➔ useful for PASAPAS restart

```
TAB1 . 'CONTINUATION' . 'TEMPS'
TAB1 . 'CONTINUATION' . 'DEPLACEMENTS'
TAB1 . 'CONTINUATION' . 'CONTRAINTES'
TAB1 . 'CONTINUATION' . 'TEMPERATURES'
…
…
```

ACCESS TO PASAPAS DATA

- **TAB1 . 'WTABLE'**
  Contains all useful variables for PASAPAS (*chosen options, models, materials and instant loading, intermediate results …)*

  Some indexes:

`WTABLE . 'CHARGEMENT'`	*Loading on the time step*
`WTABLE . 'THER_COURANT'`	*Value of temperature at last iteration (during a time step)*
`WTABLE . 'BLOCAGES_MECANIQUES'`	*Mechanical constraint matrix*
`WTABLE . 'BLOCAGES_THERMIQUES'`	*Thermal constraint matrix*
`WTABLE . 'FOR'`	*Configuration at the time step beginning*
`WTABLE . 'FOR0'`	*Initial Configuration*
`WTABLE . 'MODELE'`	*Models*
`WTABLE . 'CARACTERISTIQUES'`	*Material properties*

  …

For more information, see:

→ Comments in procedure PAS_DEFA (http://www-cast3m.cea.fr/index.php?page=procedures&procedure=pas_defa)

→ The PASAPAS documentation (http://www-cast3m.cea.fr/html/Documentation_Cast3M/Pasapas.pdf)

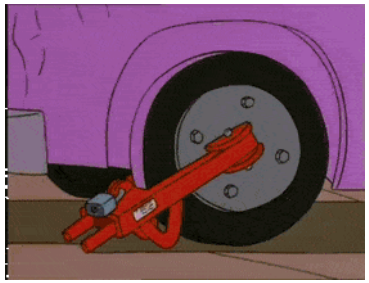
FINDING THE MECHANICAL EQUILIBRIUM

THE UNPAS PROCEDURE

REMINDER ON EQUATIONS

Local equations of the static equilibrium

equilibrium	$\mathrm{div}(\boldsymbol{\sigma}) + f = 0$	on V
prescribed surface forces	$\boldsymbol{\sigma}.n = t$	on ∂V^t
prescribed displacements	$u = d$	on ∂V^d



Weak form + FE discretization

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

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

Vectors of Equivalent nodal forces (N)

$\{F\}^S$	to the prescribed surface forces t (Pa)	(PRES, FSUR, FORC, …)
$\{F\}^R$	to the surface forces in reaction to the prescribed displacements d (m)	(REAC)
$\{F\}^V$	to the prescribed volume forces f (N.m^{-3})	(CNEQ)
$[B]\{\sigma\}$	to the internal volume forces	(BSIG)

Matrices

$[N]$	matrix of shape (interpolation) functions
$[B]$	matrix of derivatives of shape functions

IMBALANCE

- **Imbalance = measure of deviation of equilibrium**

$$\{R\} = \underbrace{\{F\}^S + \{F\}^R + \{F\}^V}_{\{F\}^{ext}} - \underbrace{[B]\{\sigma\}}_{\{F\}^{int}}$$

- **The balance is obtained when**

$$\|R\| < \varepsilon . F^{ref}$$

$\|R\|$	norm of the imbalance vector, for instance infinity norm
ε	computational precision (given by user)
F^{ref}	reference force for the studied problem

NON LINEAR BEHAVIOR

- **Introduction of the linear elastic stiffness matrix**

$$\boldsymbol{\sigma} = \mathbb{E}:\boldsymbol{\varepsilon}^{lin} + \boldsymbol{\sigma}^{nl} \qquad \text{with } \boldsymbol{\varepsilon}^{lin} = \frac{1}{2}(\nabla u + \nabla^T u)$$

$$\{\varepsilon\}^{lin} = [B]\{U\}$$

- **Equilibrium can be re written**

$$[B]\{\sigma\} = \{F\}^S + \{F\}^R + \{F\}^V$$

$$[B][E]\{\varepsilon\}^{lin} = \{F\}^S + \{F\}^R + \{F\}^V - [B]\{\sigma\}^{nl}$$

$$\underbrace{\int_V [B]^T[E][B]dV}_{[K]}\ \{U\} = \{F\}^S + \{F\}^R + \{F\}^V - [B]\{\sigma\}^{nl}$$

$$[K]\{U\} = \{F\}^S + \{F\}^R + \{F\}^V - [B]\{\sigma\}^{nl}$$

$\{U\}$	nodal displacements
$\{\varepsilon\}^{lin}$	linear strains (« small » strains)
$\{\sigma\}^{nl}$	complementary non linear stresses
$[E]$	Hooke matrix (ELAS)
$[K]$	elastic stiffness matrix (RIGI)

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

$$\{\varepsilon\}^{lin} = [B]\{U\} \quad \text{(EPSI)}$$

INTRODUCING THE PRESCRIBED DISPLACEMENTS

Boundary condition matrix

$$u = d \quad \text{on } \partial V^d$$
$$[A]\{U\} = \{d\}$$

Lagrange Multipliers

For a lin. system without prescribed displacements: $[K]\{U\} = \{F\}^S + \{F\}^V - [B]\{\sigma\}^{nl}$

We add new unknowns $\{\lambda\}$

$$[K]\{U\} + \underbrace{[A]^T\{\lambda\}}_{-\{F\}^R} = \{F\}^S + \{F\}^V - [B]\{\sigma\}^{nl} \qquad \text{with } [A].\{U\} = \{d\}$$

Equilibrium is now written:

$$\underbrace{\begin{bmatrix} K & A^T \\ A & 0 \end{bmatrix}}_{[\hat{K}]} \begin{Bmatrix} U \\ \lambda \end{Bmatrix} = \begin{Bmatrix} F^S + F^V - [B]\{\sigma\}^{nl} \\ d \end{Bmatrix}$$

depends on $\{U\}$!!
non linear behavior,
large displacements,
following forces,
…

$[A]$ boundary condition matrix `(BLOQ, RELA)`
$\{d\}$ prescribed nodal displacements `(DEPI)`
$\{\lambda\}$ Lagrange multipliers

INCREMENTAL DECOMPOSITION (1/2)

- **Computation on a time step between t_0 and t_1**

  $[t_0, \{U\}_0, \{\lambda\}_0, \{\sigma\}_0]$ known state — beginning of time step

  $[t_1, \{U\}_1, \{\lambda\}_1, \{\sigma\}_1]$ unknown state — end of time step

- **Incremental decomposition of nodal displacements**

$$\{U\}_1^i = \{U\}_0 + \Delta\{U\}_1^i \quad \text{and} \quad \Delta\{U\}_1^{i+1} = \Delta\{U\}_1^i + \delta\{U\}_1^{i+1}$$

and so

$$\{U\}_1^{i+1} = \{U\}_1^i + \delta\{U\}_1^{i+1}$$

$\{U\}_1^i$ estimation of displacement $\{U\}_1$ at iteration i

$\Delta\{U\}_1^i$ estimation of displacement increment at iteration i

$\delta\{U\}_1^{i+1}$ correction of displacement increment at iteration i

- **Incremental decomposition of boundary conditions**

$$[A]\{U\}_1^{i+1} = \{d\}_1$$

$$[A]\delta\{U\}_1^{i+1} = \{d\}_1 - [A]\{U\}_1^i$$

INCREMENTAL DECOMPOSITION (2/2)

- **Incremental decomposition of equilibrium**

$$[K]\{U\}_1^{i+1} + [A]^T\{\lambda\}_1^{i+1} = \{F\}_1^S + \{F\}_1^V - [B]\{\sigma\}_1^{nl,i}$$

$$[K]\delta\{U\}_1^{i+1} = \{F\}_1^S + \{F\}_1^V - [A]^T\{\lambda\}_1^{i+1} - \underbrace{\left([K]\{U\}_1^i + [B]\{\sigma\}_1^{nl,i}\right)}_{[B]\{\sigma\}_1^i}$$

$$[K]\delta\{U\}_1^{i+1} = \underbrace{\{F\}_1^S + \{F\}_1^V - [A]^T\{\lambda\}_1^{i+1} - [B]\{\sigma\}_1^i}_{\{R\}_1^i}$$

- **Equilibrium is finally written**

$$\underbrace{\begin{bmatrix} K & A^T \\ A & 0 \end{bmatrix}}_{[\hat{K}]} \begin{Bmatrix} \delta U_1^{i+1} \\ \lambda_1^{i+1} \end{Bmatrix} = \begin{Bmatrix} F_1^S + F_1^V - [B]\{\sigma\}_1^i \\ \{d\}_1 - [A]\{U\}_1^i \end{Bmatrix}$$

IMBALANCE MINIMIZATION ALGORITHM

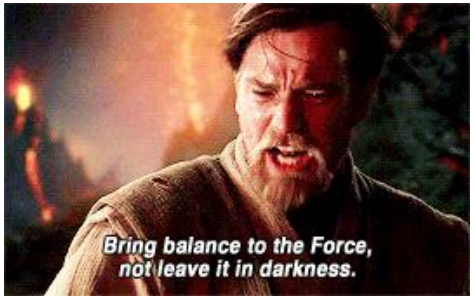
Simple algorithm

$\left[\{U\}_1^{i=0}, \{\lambda\}_1^{i=0}, \{\sigma\}_1^{i=0}\right] = \left[\{U\}_0, \{\lambda\}_0, \{\sigma\}_0\right]$ initializations

$\{F\}_1^{R,i=0} = -[A]^T \{\lambda\}_1^{i=0}$ reaction forces (REAC)

$F^{ref} = \left\| \{F\}_1^S + \{F\}_1^V + \{F\}_1^{R,i=0} \right\|$ convergence norm (MAXI 'ABS')

$\{\boldsymbol{R}\}_1^{\boldsymbol{i=0}} = \{F\}_1^S + \{F\}_1^V + \{F\}_1^{R,i=0} - [B]\{\sigma\}_1^{i=0}$ first imbalance (residual) (BSIG)



While: $\left\|\{\boldsymbol{R}\}_1^{\boldsymbol{i}}\right\| / \boldsymbol{F^{ref}} \geq \boldsymbol{\varepsilon}$

$\left[\delta\{U\}_1^{i+1}, \{\lambda\}_1^{i+1}\right] = \left[\hat{K}\right]^{-1} \{\boldsymbol{R}\}_1^{\boldsymbol{i}}$ resolution (RESO)

$\{U\}_1^{i+1} = \{U\}_1^i + \delta\{U\}_1^{i+1}$ estim. displacements

$\{\varepsilon\}_1^{i+1} = \mathcal{D}\left(\{U\}_1^{i+1}\right)$ estim. strains (EPSI)

$\{\sigma\}_1^{i+1} = \mathcal{C}\left(\Delta\{\varepsilon\}_1^{i+1}\right)$ estim. stresses (COMP)

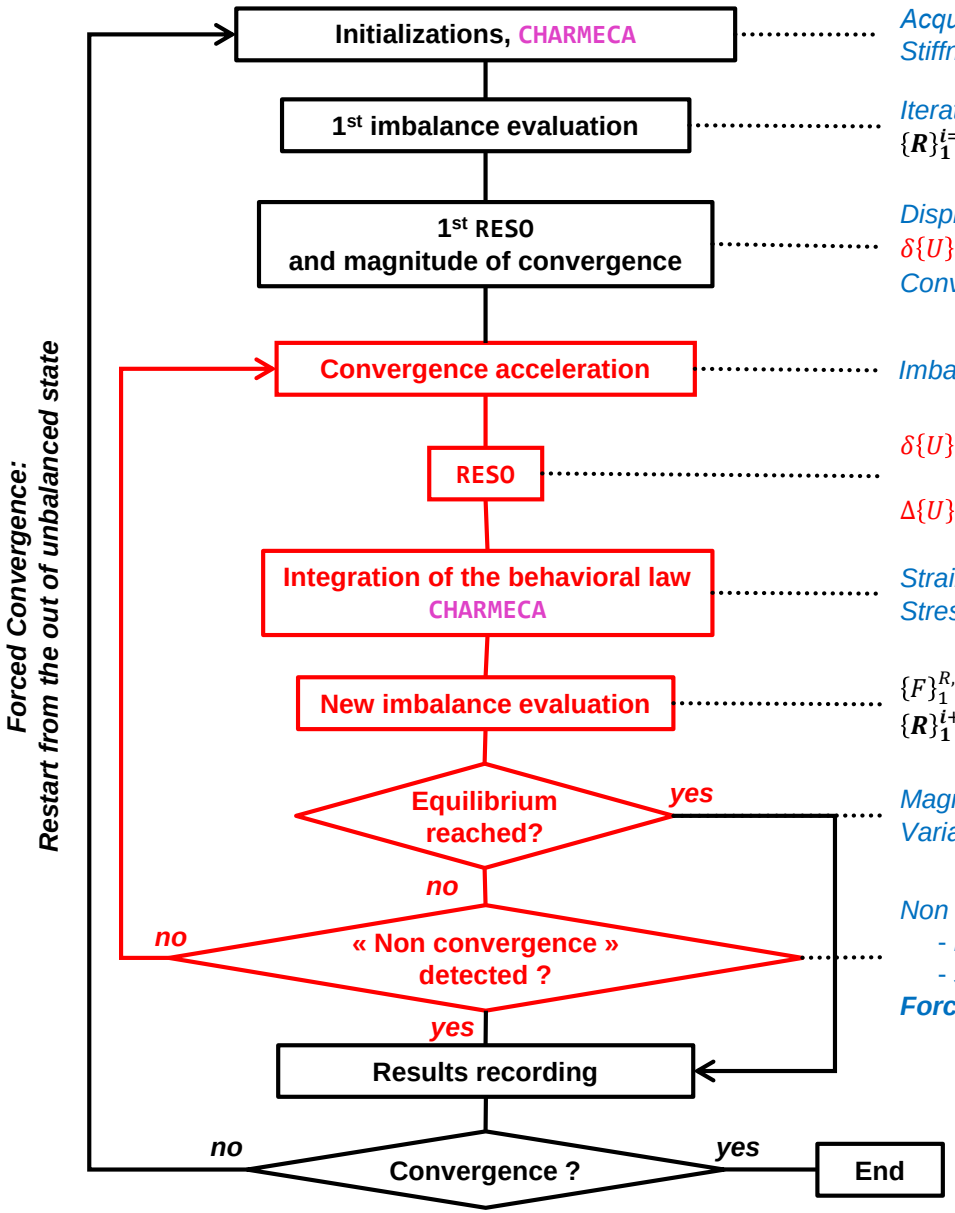
$\{F\}_1^{R,i+1} = -[A]^T \{\lambda\}_1^{i+1}$ estim. reaction forces (REAC)

$\{\boldsymbol{R}\}_1^{\boldsymbol{i+1}} = \{F\}_1^S + \{F\}_1^V + \{F\}_1^{R,i+1} - [B]\{\sigma\}_1^{i+1}$ new imbalance (residual) (BSIG)

$i = i + 1$

End

UNPAS OPERATION



Acquisition of options, materials updating, WTABLE, …
Stiffness matrix calculation $[K]$, *external prescribed forces* $\{F\}^{ext}$

Iteration engine, computes the imbalance between the internal and external forces
$\{\boldsymbol{R}\}_1^{i=0} = \{F\}_1^S + \{F\}_1^V + \{F\}_0^R - [B]\{\sigma\}_0$ (BSIG operator)

Displacement increment on the step $\Delta\{U\}_1^{i=0} = \delta\{U\}_1^{i=0}$
$\delta\{U\}_1^{i=0} = [K]^{-1}\{\boldsymbol{R}\}_1^{i=0}$ (RESO operator)
Convergence norm: F^{ref} *measure the step, along* $\{\boldsymbol{R}\}_1^{i=0}$ *and* $\Delta\{U\}_1^{i=0}$

Imbalance correction from the 3 previous imbalances (ACT3 operator)

$\delta\{U\}_1^{i+1}$ and $\{\lambda\}_1^{i+1} = [K]^{-1}\{\boldsymbol{R}\}_1^{i}$ *correction of the displ. increment on the step* (RESO operator)
$\Delta\{U\}_1^{i+1} = \Delta\{U\}_1^{i} + \delta\{U\}_1^{i+1}$ *displacement increment on the step* (ZDEPT)

Strains increment calculation on the step $\Delta\{\varepsilon\}_1^{i+1}$ (EPSI operator)
Stresses $\{\sigma\}_1^{i+1}$, *internal variables* $\{V\}_1^{i+1}$ *calculation, end of the step* (COMP op.)

$\{F\}_1^{R,i+1} = -[A]^T\{\lambda\}_1^{i+1}$ *reaction forces* (REAC operator)
$\{\boldsymbol{R}\}_1^{i+1} = \{F\}_1^S + \{F\}_1^V + \{F\}_1^{R,i+1} - [B]\{\sigma\}_1^{i+1}$ *new imbalance* (BSIG operator)

Magnitude of the imbalance < criterion (PRECISION, FTOL, MTOL) ??
Variation of the strains increment between 2 iterations < criterion ??

Non convergence ***detection****:*
- *maximum number of iterations reached*
- *significant increase of the imbalance over several iterations*

Forced convergence process

USERS PROCEDURES IN PASAPAS

PERSO1 REEV_MEC CHARMECA

PERSO2 REEV_THE CHARTHER PARATHER

USERS PROCEDURES: INSTRUCTIONS FOR USE

- There exists 7 users procedures called at different stages of PASAPAS algorithm

  PERSO1 PERSO2 REEV_MEC REEV_THE
  CHARMECA CHARTHER PARATHER

- 1) According to the desired action, choose the procedure to be used as a function of its location on the algorithm

  for instance: since PERSO1 is called after computing a time step it is dedicated to update the next time step data
  e.g. boundary conditions, material parameters, …

- 2) Analyze the procedure syntax
  - in the PASAPAS manual pages
  - In the PASAPAS, UNPAS or TRANSNON procedures

  the following table recalls the procedures syntaxes

PERSO1 PROCEDURE EXAMPLE

- 3) Define the desired procedure

```
DEBP PERSO1 T1*'TABLE' ;
  MESS 'Hello! The PERSO1 procedure is talking to you.' ;
FINP ;
```

- 4) Indicate in the PASAPAS table that this procedure should be called

```
TAB1 . 'MODELE' = MOD1 ;
TAB1 . 'CARACTERISTIQUES' = MAT1 ;
TAB1 . 'TEMPS_CALCULES' = PROG 1. 2. 3. 4. ;
TAB1 . 'PROCEDURE_PERSO1' = VRAI ;
...
...
PASAPAS TAB1 ;
```

USERS PROCEDURES LIST

Name	Index to be activated in PASAPAS table	Syntax	Function
PERSO1	TAB1 . 'PROCEDURE_PERSO1' = VRAI ;	PERSO1 T1 ;	Updates the problem after the mechanical calculation step
REEV_MEC	TAB1 . 'PROCEDURE_REEV_MEC' = VRAI ;	REEV_MEC T1 N1 ;	Ditto (but in BO_BOTH) ➔ the step can be started again
CHARMECA	TAB1 . 'PROCEDURE_CHARMECA' = VRAI ;	T2 = CHARMECA T1 TPS1 ;	Mechanical loads addition during UNPAS iterations
PERSO2	TAB1 . 'PROCEDURE_PERSO2' = VRAI ;	PERSO2 T1 ;	Updates the problem after the thermal calculation step
REEV_THE	TAB1 . 'PROCEDURE_REEV_THE' = VRAI ;	REEV_THE T1 N1 ;	Ditto (but in PASAPAS) ➔ the step can be started again
CHARTHER	TAB1 . 'PROCEDURE_CHARTHER' = VRAI ;	T2 = CHARTHER T1 TPS1 ;	Thermal loads addition during TRANSNON iterations
PARATHER	TAB1 . 'PROCEDURE_PARATHER' = VRAI ;	PARATHER T1 TPS1 ;	Updating of the external variables of thermal parameters

With

- **T1 : the PASAPAS table**
- **TPS1 : the current calculated time**
- **N1 : the call number of the procedure (0 or 1) → see the PASAPAS procedure**
- **T2 : the output table (only for the CHARMECA and CHARTHER procedures)**

A FEW REMARKS

- **Instructions inside these procedures are free !!!**

- **CHARMECA and CHARTHER output is a table with 2 possible indexes:**
  - `'ADDI_MATRICE'` contains the stiffness matrices (`RIGIDITE` type object) to be added to the 1st member
  - `'ADDI_SECOND'` contains the `CHPOINT` to be added to the 2nd member (nodal forces)

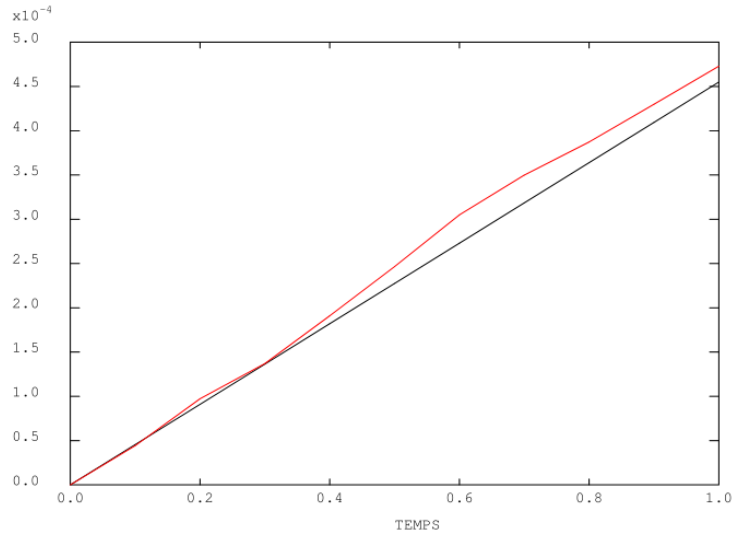
- **For large displacements (`'GRANDS_DEPLACEMENTS'` option), CHARMECA is called on the deformed shape**

A SIMPLE EXAMPLE

Random variation of Young's modulus map at each time step

```
OPTI 'DIME' 2 'ELEM' 'QUA4' ;
P0 = 0. 0. ;   P1 = 1. 0. ;
L1 = DROI 1 P0 (0. 0.1) ;
S1 = L1 TRAN 10 (1. 0.) ;
L2 = S1 COTE 3 ;   P2 = L2 POIN 'PROC' (1. 0.) ;

MO = MODE S1 'MECANIQUE' ;
MA = MATE MO 'YOUN' 200.E9 'NU' 0.3 ;

BL  = (BLOQ 'UX' L1) ET (BLOQ 'UY' P0) ;
F   = PRES 'MASS' MO L2 -1.E8 ;
EV  = EVOL 'MANU' (PROG 0. 1.) (PROG 0. 1.) ;
CHA = CHAR 'MECA' F EV ;

TAB1 = TABL ;
TAB1 . 'MODELE'                = MO ;
TAB1 . 'CARACTERISTIQUES'      = MA ;
TAB1 . 'BLOCAGES_MECANIQUES' = BL ;
TAB1 . 'CHARGEMENT'            = CHA ;
TAB1 . 'TEMPS_CALCULES'        = PROG 0.1 'PAS' 0.1 1. ;
TAB1 . 'PROCEDURE_PERSO1'      = VRAI ;

PASAPAS TAB1 ;

EVU = EVOL 'TEMP' TAB1 'DEPLACEMENTS' 'UX' P2 ;
DESS EVU ;
```

```
DEBP PERSO1 T1*'TABLE' ;
  MOD   = T1 . 'WTABLE' . 'MO_TOT' ;
  MAIL  = EXTR MOD 'MAIL' ;
  CHPYO = BRUI 'BLAN' 'UNIF' 200.E9 150.E9 MAIL ;
  CHMYO = CHAN 'CHAM' CHPYO MOD 'RIGIDITE' ;
  MAT   = MATE MOD 'YOUN' CHMYO 'NU' 0.3 ;
  T1 . 'WTABLE' . 'CARACTERISTIQUES'   = MAT ;
FINP ;
```



EXERCISE 1:

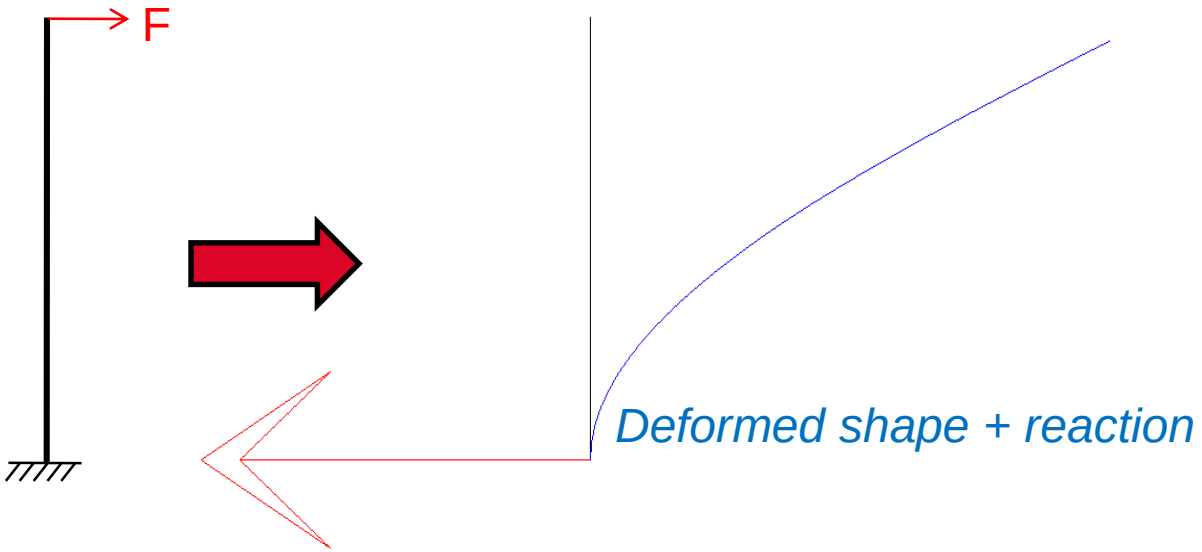
BEAM WITH FOLLOWING FORCE

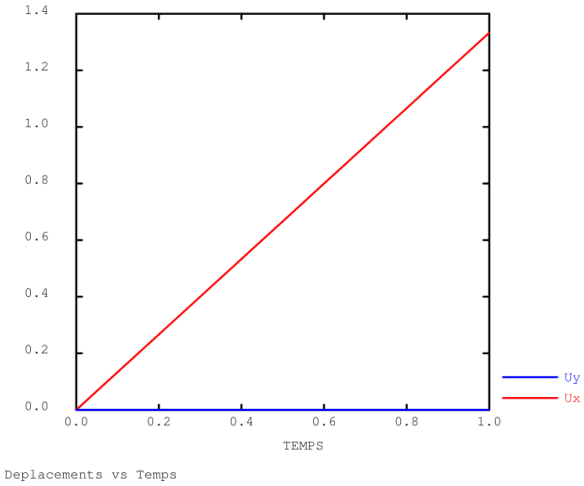
DOWNLOAD THE STARTING FILE ON THE WEBSITE:

HTTP://WWW-CAST3M.CEA.FR/INDEX.PHP?PAGE=EXEMPLES&EXEMPLE=FORMATION_PASAPAS_1_INITIAL

EXERCISE 1: BEAM WITH FOLLOWING FORCE

- **Beam bending**
  base is clamped
  point shear force at the top of the beam (perpendicular to the beam)
  large displacement at the top of the beam





- **Problem: the force is calculated on the initial shape and not updated**
- **Purpose: to correctly apply the force during the calculation**

It's up to you!

EXERCISE 1: BEAM WITH FOLLOWING FORCE

A few indications …

- **Useful objects**

`P2` : point at the top of the beam, where the force is prescribed
`EV1` : force magnitude to be applied as a function of time

- **Useful operators**

`EXTR, COS, SIN`
`IPOL` : to interpolate the force magnitude as a function of the calculated step
`FORC` : to apply a point force

EXERCISE 1: BEAM WITH FOLLOWING FORCE

Solution

- use option **'GRANDS_DEPLACEMENTS'**
  (equilibrium verified on the deformed shape)
- use procedure **PERSO1** (1 call / time step)
  to re-calculate the force on deformed shape (at beginning of time step)
- create a “load” with it (CHARGEMEnt type object)
- overwrite the load in **WTABLE**

```
DEBP PERSO1 T1*'TABLE' ;
* donnees utiles
  TAM = T1 . 'MY_DATA' ;
  PF = TAM . 'POINT' ;
  EV = TAM . 'EVOL' ;
* angle de rotation de la section droite
  U1 = T1 . 'ESTIMATION' . 'DEPLACEMENTS' ;
  RS = EXTR U1 'RZ' PF ;
  RSD = RS * 180. / PI ;
* direction de l'effort
  DIR1 = (COS RSD) (SIN RSD) ;
* creation du second membre
  F1 = FORC DIR1 PF ;
  CHAR2 = CHAR 'MECA' F1 EV ;
  T1 . 'WTABLE' . 'CHARGEMENT' = CHAR2 ;
FINP ;
```

PERSO1 procedure

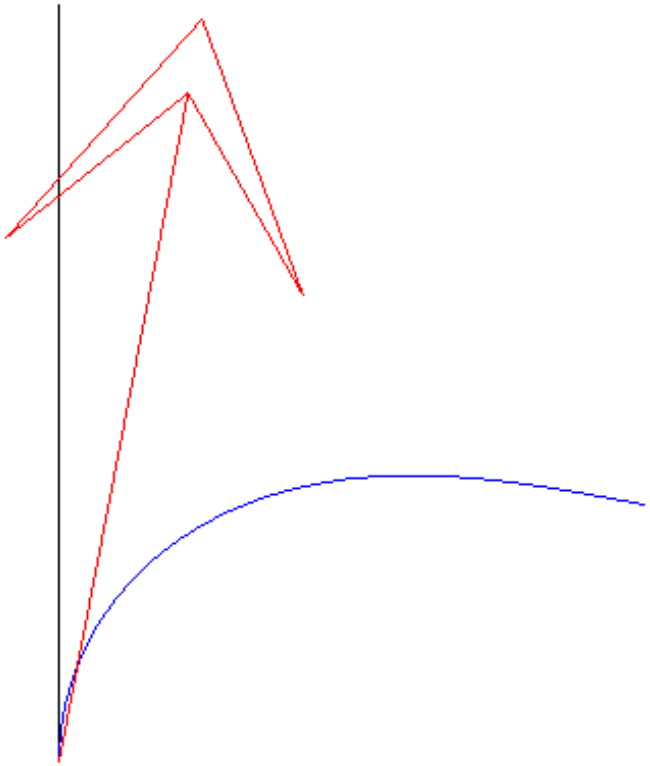
```
TAB1 . 'GRANDS_DEPLACEMENTS' = VRAI ;
TAB1 . 'PREDICTEUR'          = 'HPP' ;
TAB1 . 'PROCEDURE_PERSO1'    = VRAI ;
TAB1 . 'MY_DATA'             = TABL ;
TAB1 . 'MY_DATA' . 'EVOL'    = EV1 ;
TAB1 . 'MY_DATA' . 'POINT'   = P2 ;
```

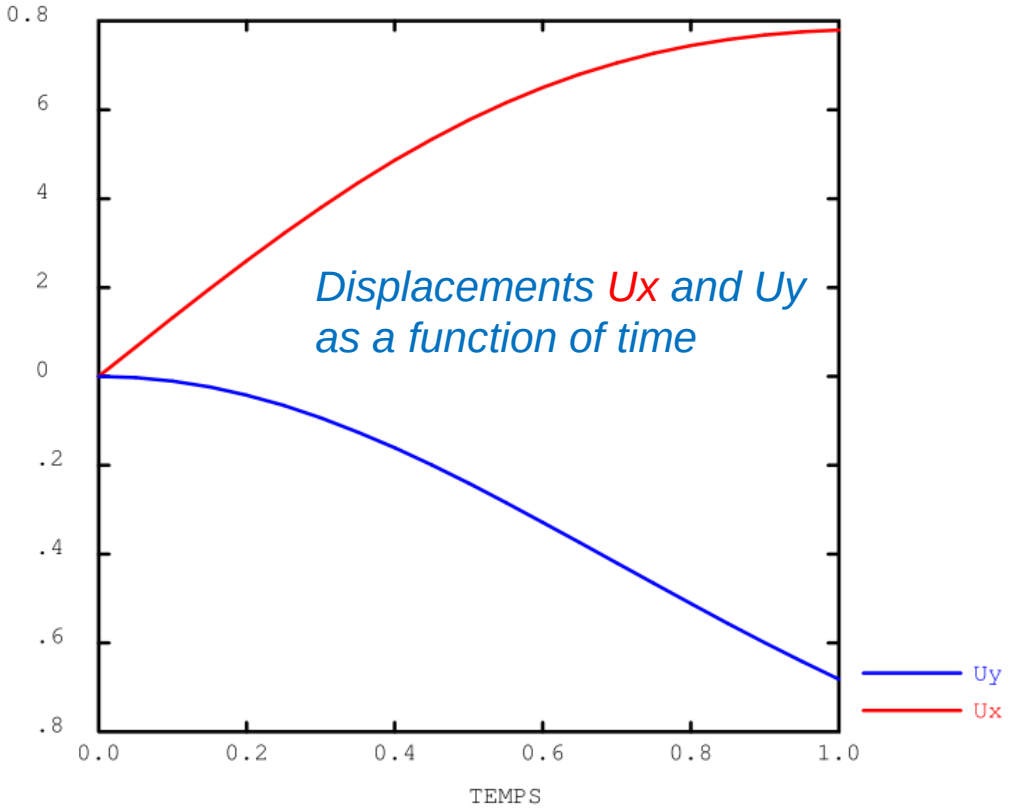
Main program

EXERCISE 1: BEAM WITH FOLLOWING FORCE

Results

Deformed shape + reaction





Deplacements vs Temps

EXERCISE 1: BEAM WITH FOLLOWING FORCE

- **Solution (bis)**
- idem but with the **CHARMECA** procedure
- suppress initial mechanical loading
- no need of CHARGEMEnt type object
- longer: 1 call / each iteration / time step
- results are identical

```
DEBP CHARMECA T1*'TABLE' TPS1*'FLOTTANT' ;
* donnees utiles
  TAM = T1 . 'MY_DATA' ;
  PF = TAM . 'POINT' ;
  EV = TAM . 'EVOL' ;
* angle de rotation de la section droite
  U1 = T1 . 'ESTIMATION' . 'DEPLACEMENTS' ;
  RS = EXTR U1 'RZ' PF ;
  RSD = RS * 180. / PI ;
* direction de l'effort
  DIR1 = (COS RSD) (SIN RSD) ;
* creation du second membre
  VAL1 = IPOL TPS1 EV ;
  F1 = FORC (VAL1 * DIR1) PF ;
  T2 = TABLE ;
  T2 . 'ADDI_SECOND' = F1 ;
FINP T2 ;
```

CHARMECA procedure

```
***TAB1 . 'CHARGEMENT'           = CHAR1 ;
TAB1 . 'GRANDS_DEPLACEMENTS' = VRAI ;
TAB1 . 'PREDICTEUR'          = 'HPP' ;
TAB1 . 'PROCEDURE_CHARMECA'  = VRAI ;
TAB1 . 'MY_DATA'             = TABL ;
TAB1 . 'MY_DATA' . 'EVOL'    = EV1 ;
TAB1 . 'MY_DATA' . 'POINT'   = P2 ;
```

Main program

EXERCISE 1: BEAM WITH FOLLOWING FORCE

Solution (ter)

- idem, but the force is updated on the deformed shape at the end of time step!
- more precise: when converged, the deformed geometry is exactly perpendicular to applied force
- watch out: can become unstable 😱 !

```
DEBP CHARMECA T1*'TABLE' TPS1*'FLOTTANT' ;
* donnees utiles
  TAM = T1 . 'MY_DATA' ;
  PF = TAM . 'POINT' ;
  EV = TAM . 'EVOL' ;
* angle de rotation de la section droite
  DRS = 0. ;
  SI (EGA (TYPE ZDEPT) 'CHPOINT') ;
    DRS = EXTR ZDEPT 'RZ' PF ;
  FINSI ;
  U0 = T1 . 'CONTINUATION' . 'DEPLACEMENTS' ;
  RS = (EXTR U0 'RZ' PF) + DRS ;
  RSD = RS * 180. / PI ;
* direction de l'effort
  DIR1 = (COS RSD) (SIN RSD) ;
* creation du second membre
  VAL1 = IPOL TPS1 EV ;
  F1 = FORC (VAL1 * DIR1) PF ;
  T2 = TABLE ;
  T2 . 'ADDI_SECOND' = F1 ;
FINP T2 ;
```

CHARMECA procedure

```
***TAB1 . 'CHARGEMENT'           = CHAR1 ;
TAB1 . 'GRANDS_DEPLACEMENTS' = VRAI ;
TAB1 . 'PREDICTEUR'          = 'HPP' ;
TAB1 . 'PROCEDURE_CHARMECA'  = VRAI ;
TAB1 . 'MY_DATA'             = TABL ;
TAB1 . 'MY_DATA' . 'EVOL'    = EV1 ;
TAB1 . 'MY_DATA' . 'POINT'   = P2 ;
```

Main program

EXERCISE 2:

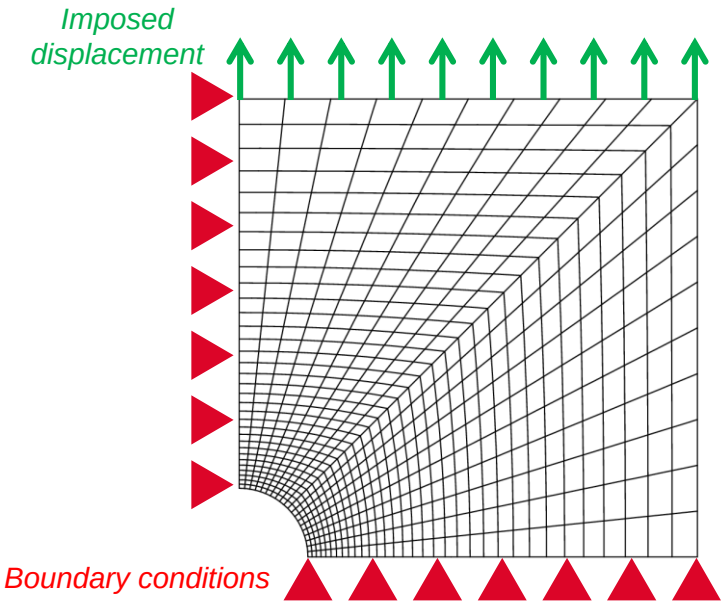
RUPTURE BY ELEMENTS REMOVAL

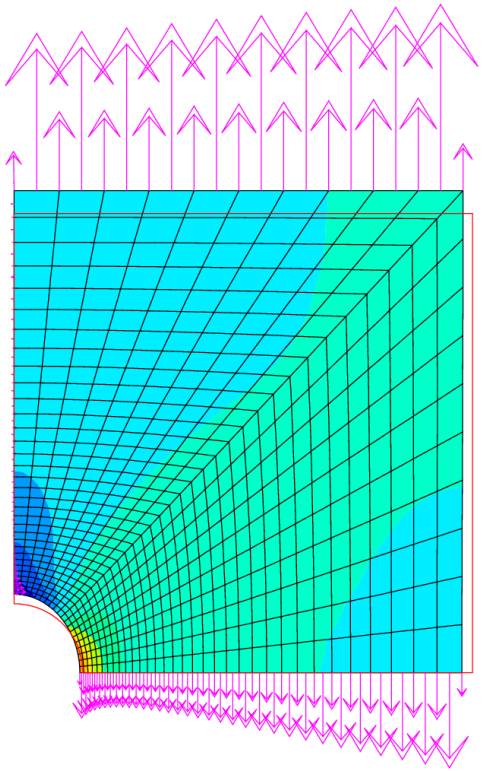
DOWNLOAD THE STARTING FILE ON THE WEBSITE:

HTTP://WWW-CAST3M.CEA.FR/INDEX.PHP?PAGE=EXEMPLES&EXEMPLE=FORMATION_PASAPAS_2_INITIAL

EXERCISE 2: RUPTURE BY ELEMENTS REMOVAL

- **Traction of a perforated plate**
  elastic behavior, large displacements





σ_I stress and reaction forces on the deformed shape

- **Purpose: to model fracture by removing elements during calculations**
  We will use a simple criterion based on the 1st principal stress: fracture if $\sigma_I \geq 22$ GPa

It's up to you!

EXERCISE 2: RUPTURE BY ELEMENTS REMOVAL

A few indications …

Useful objects

`PRIN`: principal stress field calculations
`CHAN`: change the support points of a field by elements
`ELEM`: select the elements of a field that meets a criteria
`REDU`: reduction of a model on a sub-mesh

Useful information

Modify the temporary objects for computation in **`TAB1 . WTABLE`**

EXERCISE 2: RUPTURE BY ELEMENTS REMOVAL

- **Solution**
- reduce time steps
- use the **PERSO1** procedure
- extract the model and the calculated stresses (**T1 . ESTIMATION**)
- calculate the principal stresses
- determine the mesh of “intact elements”
- reduce the model on this mesh
- overwrite the model in the working table **WTABLE**

```
DEBP PERSO1 T1*'TABLE' ;
  MO1 = T1 . 'WTABLE' . 'MODELE' ;
  SG1 = T1 . 'ESTIMATION' . 'CONTRAINTES' ;
  SGP1 = PRIN SG1 MO1 ;
  SG11 = CHAN 'GRAVITE' (EXCO 'SI11' SGP1) MO1 ;
  MAIL2 = SG11 ELEM 'INFERIEUR' 2.2E10 ;
  MO2 = REDU MO1 MAIL2 ;
  T1 . 'WTABLE' . 'MODELE'   = MO2 ;
FINP ;
```

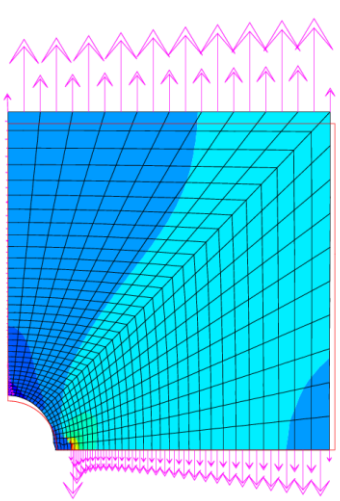
PERSO1 procedure

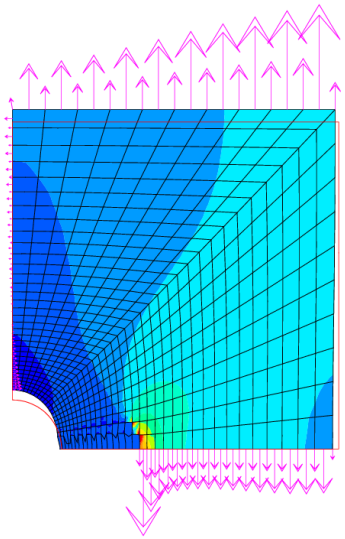
```
TAB1 . 'PROCEDURE_PERSO1' = VRAI ;
TAB1 . 'TEMPS_CALCULES'   = PROG 0.1 'PAS' 0.1  0.6
                                     'PAS' 0.01 1. ;
```

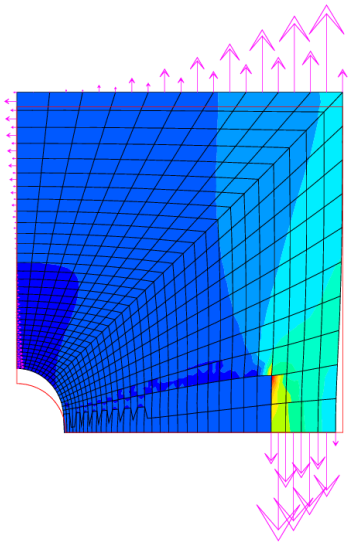
Main program

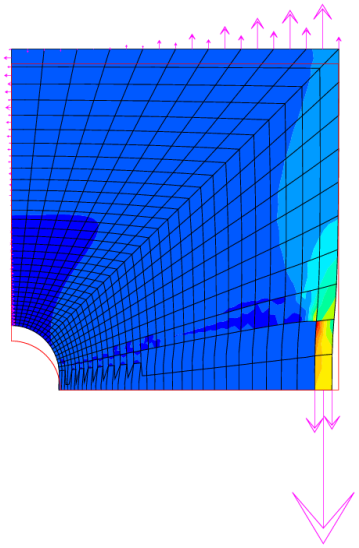
EXERCISE 2: RUPTURE BY ELEMENTS REMOVAL

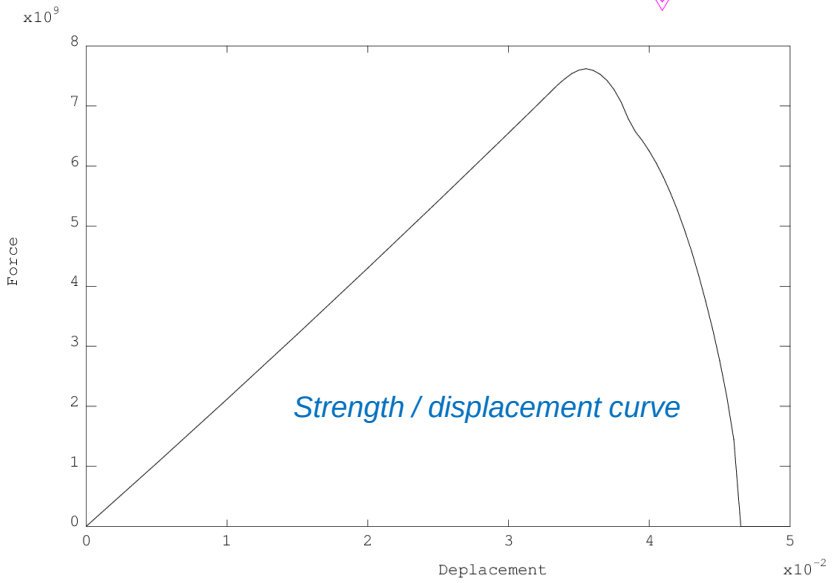
Results











Strength / displacement curve

***Undependable model**:*
results quite sensitive to time/space discretization

EXERCISE 2: RUPTURE BY ELEMENTS REMOVAL

- **Solution (bis)**
- reduce time steps
- use the **PERSO1** procedure
- extract the model and the calculated stresses (**T1 . ESTIMATION**)
- calculate the principal stresses
- determine the mesh of intact elements
- reduce the **boundary conditions** on this mesh
- overwrite the boundary conditions in the working table **WTABLE**

```
DEBP PERSO1 T1*'TABLE' ;
  MO1 = T1 . 'WTABLE' . 'MODELE' ;
  MAIL1 = EXTR MO1 'MAIL' ;
  SG1 = T1 . 'ESTIMATION' . 'CONTRAINTES' ;
  SGP1 = PRIN SG1 MO1 ;
  SG11 = CHAN 'GRAVITE' (EXCO 'SI11' SGP1) MO1 ;
  MAIL2 = SG11 ELEM 'INFERIEUR' 2.2E10 ;
  MAIL3 = DIFF MAIL1 MAIL2 ;
  NE3 = NBEL MAIL3 ;
  SI (NE3 > 0) ;
    MESS ' [PERSO1 :] Removal of' NE3 'elements' ;
    BL01 = T1 . 'WTABLE' . 'BLOCAGES_MECANIQUES' ;
    MAILBL1 = EXTR BL01 'MAIL' ;
    MAILBL2 = MAILBL1 ELEM 'APPU' 'LARG' MAIL3 ;
    MAILBL3 = DIFF MAILBL1 MAILBL2 ;
    BL02 = REDU BL01 MAILBL3 ;
    T1 . 'WTABLE' . 'BLOCAGES_MECANIQUES' = BL02 ;
  FINSI ;
FINP ;
```

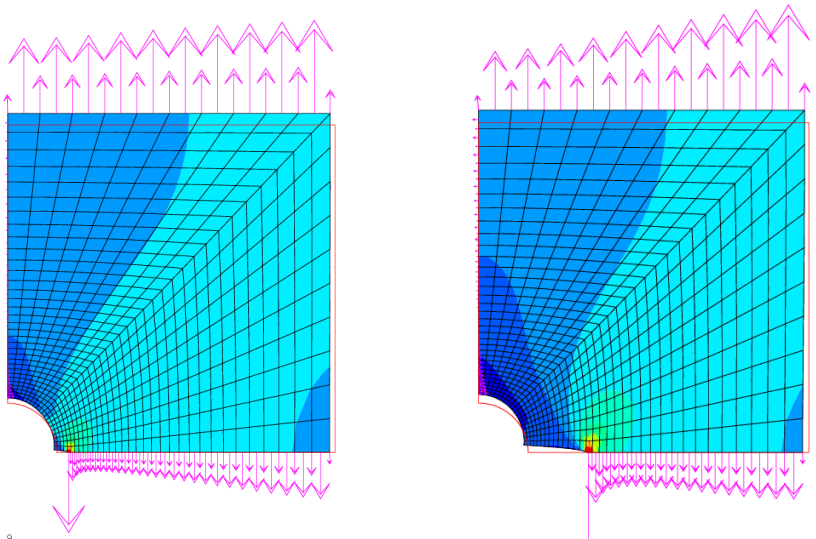
PERSO1 procedure

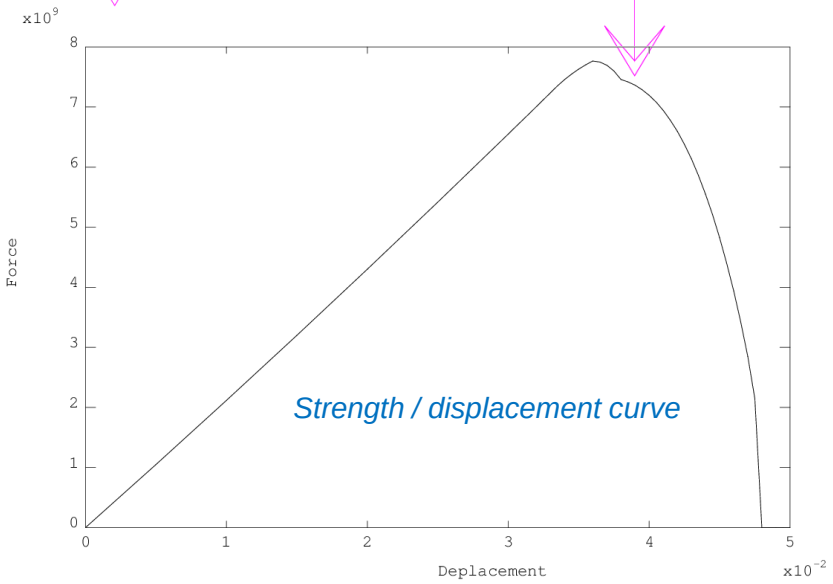
```
TAB1 . 'PROCEDURE_PERSO1' = VRAI ;
TAB1 . 'TEMPS_CALCULES'   = PROG 0.1 'PAS' 0.1  0.6
                                     'PAS' 0.01 1. ;
```

Main program

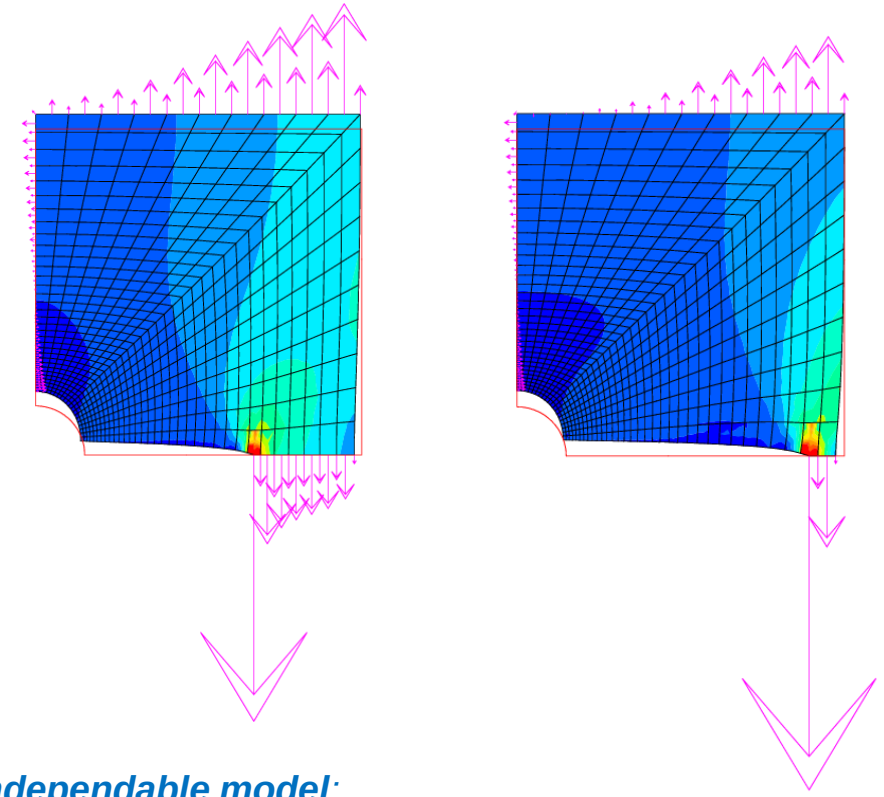
EXERCISE 2: RUPTURE BY ELEMENTS REMOVAL

Results





Strength / displacement curve



***Undependable model**:*
results quite sensitive to time/space discretization

TRANSNON OPERATION

THERMAL SOLVER

REMINDER ON EQUATIONS

Local equations in transient thermal analysis

heat flux density	$\vec{\varphi} = -\lambda \overrightarrow{\text{grad}}(T)$	on V
heat equation	$\rho c_p \frac{\partial T}{\partial t} + \text{div}\, \vec{\varphi} - q = 0$	on V
prescribed flux	$\vec{\varphi}.\vec{n} = \varphi_{imp} + \underbrace{h(T_f - T)}_{\text{convection}} + \underbrace{\varepsilon\sigma(T_\infty^4 - T^4)}_{\text{radiation}}$	on ∂V^{φ}
prescribed temperature	$T = T_{imp}$	on ∂V^{T}

REMINDER ON EQUATIONS

Weak form + FE discretization

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

Vectors of Equivalent Nodal Heat (W) $\{F\} = \{F\}^{\varphi} + \{F\}^{V}$

$\{F\}^{\varphi} = \int_{\partial V^{\varphi}} [N]^T \varphi_{imp} dS$ to the prescribed surface heat flux φ_{imp} (J.m^{-2}) (FLUX)

$+ \int_{\partial V^{\varphi}} [N]^T h T_f dS$ to the prescribed convection flux hT_f (J.m^{-2}) (CONV)

$+ \int_{\partial V^{\varphi}} [N]^T \varepsilon\sigma(T_{\infty}^4 - T^4) dS$ to the prescribed radiation flux (J.m^{-2})

$\{F\}^{V} = \int_V [N]^T q \, dV$ to the prescribed volume heat q (J.m^{-3}) (SOUR)

Matrices

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

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

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

$[B]$ matrix of derivatives of shape functions

THETA METHOD

- **Weak form + FE discretization**

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

- **Time discretization: theta method**

$$[C]_* \frac{\{T\}_{n+1} - \{T\}_n}{\Delta t} + [K]_*\{T\}_* = \{F\}_*$$

with:

Δt time step

$\{T\}_n$ vector of nodal temperatures at time t_n

$\{T\}_{n+1}$ vector of nodal temperatures at time t_{n+1}

$[C]_*$ $[K]_*$ and $\{F\}_*$ are estimated:

at time $t_* = \theta t_{n+1} + (1-\theta)t_n$

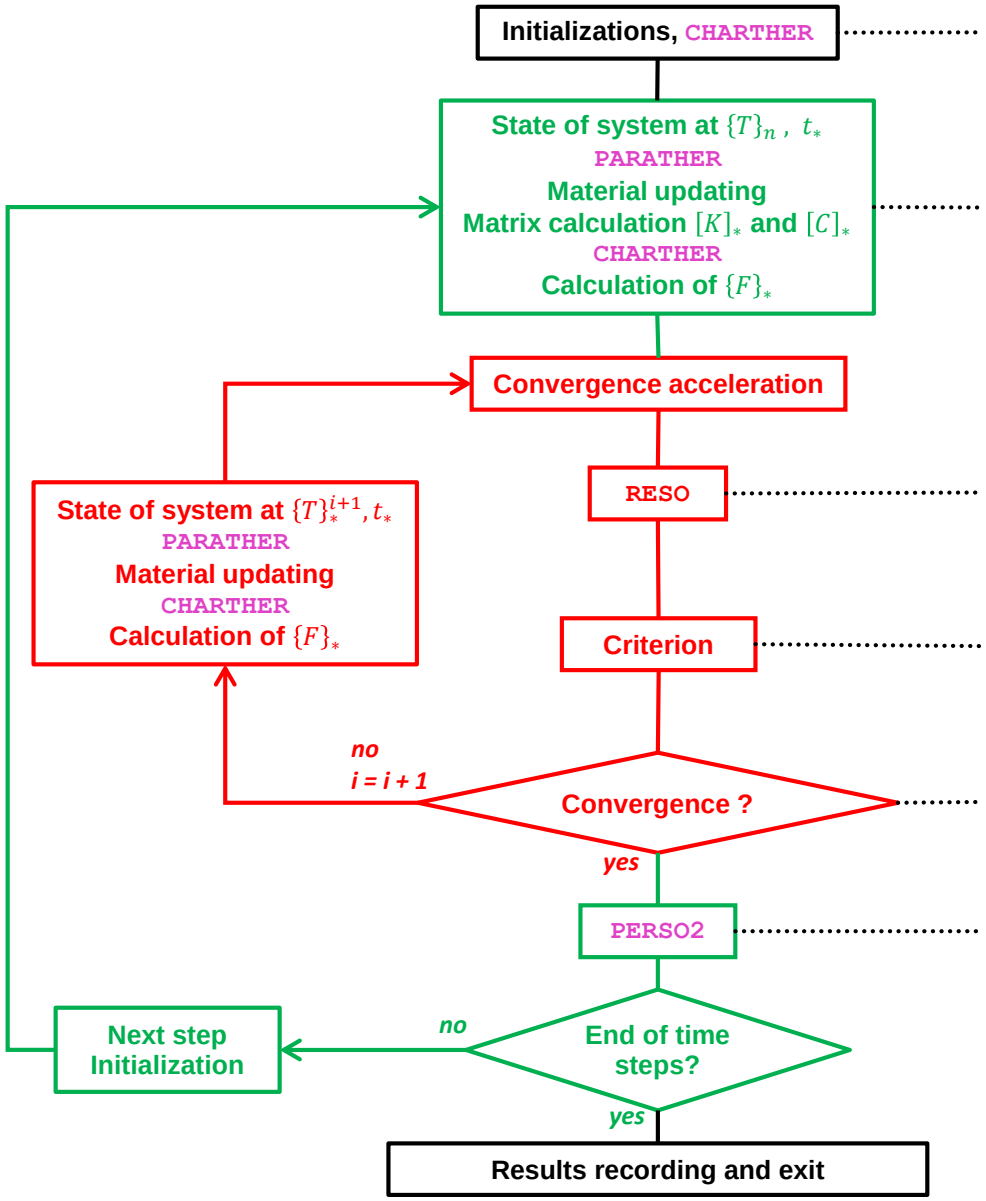
at temperature $\{T\}_* = \theta\{T\}_{n+1} + (1-\theta)\{T\}_n$

θ relaxation coefficient (between 0 and 1)

$\theta = 0$ explicit scheme

$\theta = 1$ implicit scheme (default value)

TRANSNON OPERATION



Options, times calculated, units (°C or K), initial state, shape functions (for radiation)…

External variables updating, material characteristics
Calculation of conductivity matrix $[K]_*$ *(cond. + conv. + radi. + constrain.)*
and capacity matrix $[C]_*$ *at temperature* $\{T\}_n$ *and time* t_*
Calculation of 2nd member $\{F\}_*$ *(T impo. + T conv. + T rayo. + Flux impo.)*
at temperature $\{T\}_n$ *and time* t_*

We define the operator as: $[L]_* = \frac{1}{\Delta t}[C]_* + \theta[K]_*$

Temperature increment calculation: $\Delta\{T\}^{i+1} = [L]_*^{-1}\{F\}_*$
and so we get: $\{T\}_{n+1}^{i+1} = \{T\}_n + \Delta\{T\}^{i+1}$
and also: $\{T\}_*^{i+1} = \theta\{T\}_{n+1}^{i+1} + (1-\theta)\{T\}_n$

Measure the maximal difference on the temperature increment for two successive steps:
`Crit = [MAXI 'ABS'` $\left(\Delta\{T\}^{i+1} - \Delta\{T\}^{i}\right)$`] / [MAXI 'ABS'` $\{T\}_{n+1}^{i+1}$`]`

If the criterion is small enough:
`Crit ≤ 'PRECISION'` (1.E-4 by default)

User procedure

Remark:
when TRANSNON is called by PASAPAS,
the **time step loop** is performed once

EXERCISE 3:

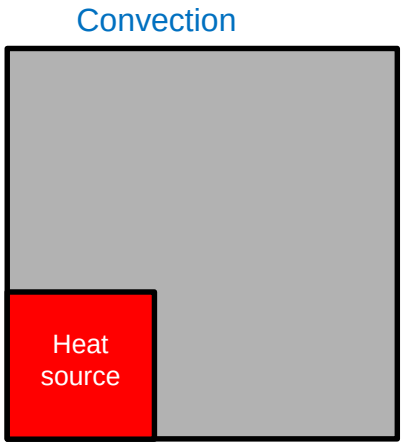
HEAT SOURCE DEPENDING ON TEMPERATURE

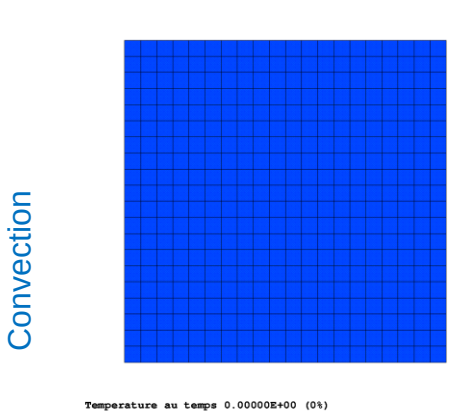
DOWNLOAD THE STARTING FILE ON THE WEBSITE:

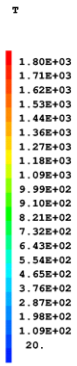
HTTP://WWW-CAST3M.CEA.FR/INDEX.PHP?PAGE=EXEMPLES&EXEMPLE=FORMATION_PASAPAS_3_INITIAL

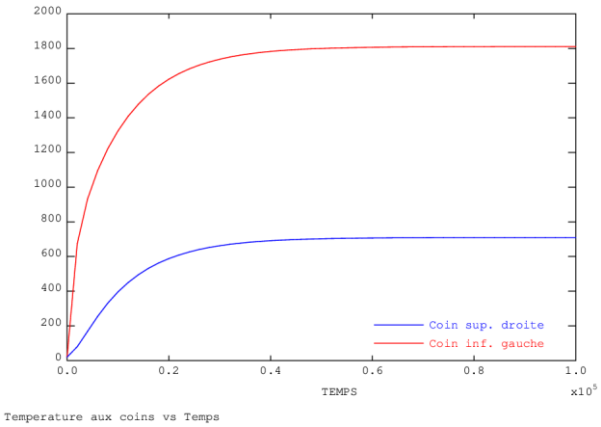
EXERCISE 3: HEAT GENERATION DEPENDING ON TEMPERATURE

- **Square section with heat source and cooled by convection**



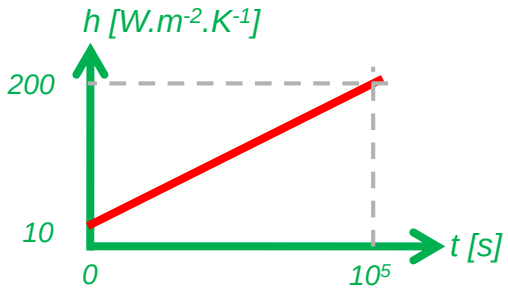






- **Purpose: make the problem depending on parameters**
  - → Conductivity as a function of temperature
    $\lambda(T) = 0{,}3.T + 200$
  - → Convection as a function of time
  - → Source as a function of temperature

$$f(T) = 4\,000\,000\ e^{-\left(\frac{T-1000}{700}\right)^2}$$



It's up to you!

EXERCISE 3: HEAT GENERATION DEPENDING ON TEMPERATURE

A few indications …

Useful objects

`MAIL1` : source mesh
`MOD1` : thermal model reduced on MAIL1

Useful operators

`REDU` : to reduce the temperature field on the "source“ area
`SOUR` : to impose a volumetric heat source

EXERCISE 3: HEAT GENERATION DEPENDING ON TEMPERATURE

- **Solution**
- use the **PERSO2** procedure (called at each time step)
- recalculate the source (2nd member) as a function of temperatures at the beginning of the time step
- create a “load” with it and overwrite the load in WTABLE

```
DEBP PERSO2 T1*'TABLE' ;
  MA1 = T1 . 'AMOI' . 'MAIL' ;
  MO1 = T1 . 'AMOI' . 'MODE' ;
* reduction du champ de temperatures au maillage de la source
  CHT1 = T1 . 'ESTIMATION' . 'TEMPERATURES' ;
  CHT2 = REDU CHT1 MA1 ;
* calcul du champ de source a partir du champ de temperature
  CHS = 4.E6 * (EXP (-1. * (((CHT2 - 1000.) / 700.) ** 2))) ;
* creation du second membre et d'un chargement
  CHFL1 = SOUR MO1 CHS ;
  EV1   = EVOL 'MANU' (PROG 0. TPSMAX) (PROG 1. 1.) ;
  CHA1  = CHAR 'Q' CHFL1 EV1 ;
* modification du chargement dans la table de calcul
  T1 . 'WTABLE' . 'CHARGEMENT' = CHA1 ;
FINP T2 ;
```

PERSO2 procedure

```
LT       = PROG -1.E6  1.E6 ;
LL       = 200. + (0.3 * LT) ;
LAMBDA   = EVOL 'MANU' 'T' LT 'K' LL ;


HCONV    = EVOL 'MANU' 'TEMP' (PROG 0. TPSMAX) 'H' (PROG 10. 200.) ;


VSOUR    = 4.E6 * (EXP (-1. * (((T_INI - 1000.) / 700.) ** 2))) ;


TAB1 . 'PROCEDURE_PERSO2'    = VRAI ;
TAB1 . 'MY_DATA'             = TABL ;
TAB1 . 'MY_DATA' . 'MAIL'    = MAIL1 ;
TAB1 . 'MY_DATA' . 'MODE'    = MOD1 ;
```

Main program

EXERCISE 3: HEAT GENERATION DEPENDING ON TEMPERATURE

- **Solution (bis)**
- suppress loading CHA1 (heat source)
- use the **CHARTHER** procedure (called at each step iteration)
- recalculate the source (2nd member) as a function of temperatures at the beginning of the time step

```
DEBP CHARTHER T1*'TABLE'  TPS1*'FLOTTANT' ;
  MA1 = T1 . 'MY_DATA' . 'MAIL' ;
  MO1 = T1 . 'MY_DATA' . 'MODE' ;
* reduction du champ de temperatures au maillage de la source
  CHT1 = T1 . 'ESTIMATION' . 'TEMPERATURES' ;
  CHT2 = REDU CHT1 MA1 ;
* calcul du champ de source a partir du champ de temperature
  CHS = 4.E6 * (EXP (-1. * (((CHT2 - 1000.) / 700.) ** 2))) ;
* creation du second membre
  CHFL1 = SOUR MO1 CHS ;
* sortie du second membre
  T2 = TABL ;
  T2 . 'ADDI_SECOND' = CHFL1 ;
FINP T2 ;
```

CHARTHER procedure

```
LT       = PROG -1.E6  1.E6 ;
LL       = 200. + (0.3 * LT) ;
LAMBDA   = EVOL 'MANU' 'T' LT 'K' LL ;


HCONV    = EVOL 'MANU' 'TEMP' (PROG 0. TPSMAX) 'H' (PROG 10. 200.) ;



***TAB1 . 'CHARGEMENT'          = CHA2 ;
TAB1 . 'PROCEDURE_CHARTHER'  = VRAI ;
TAB1 . 'MY_DATA'             = TABL ;
TAB1 . 'MY_DATA' . 'MAIL'    = MAIL1 ;
TAB1 . 'MY_DATA' . 'MODE'    = MOD1 ;
```

Main program

EXERCISE 3: HEAT GENERATION DEPENDING ON TEMPERATURE

- **Solution (ter)**
- idem
- use the **CHARTHER** procedure (called at each step iteration)
- recalculate the source (2nd member) as a function of temperatures at previous iteration (i.e. at the end of the time step)

```
DEBP CHARTHER T1*'TABLE'  TPS1*'FLOTTANT' ;
  MA1 = T1 . 'MY_DATA' . 'MAIL' ;
  MO1 = T1 . 'MY_DATA' . 'MODE' ;
* reduction du champ de temperatures au maillage de la source
  CHT1 = T1 . 'WTABLE' . 'THER_COURANT' ;
  CHT2 = REDU CHT1 MA1 ;
* calcul du champ de source a partir du champ de temperature
  CHS = 4.E6 * (EXP (-1. * (((CHT2 - 1000.) / 700.) ** 2))) ;
* creation du second membre
  CHFL1 = SOUR MO1 CHS ;
* sortie du second membre
  T2 = TABL ;
  T2 . 'ADDI_SECOND' = CHFL1 ;
FINP T2 ;
```

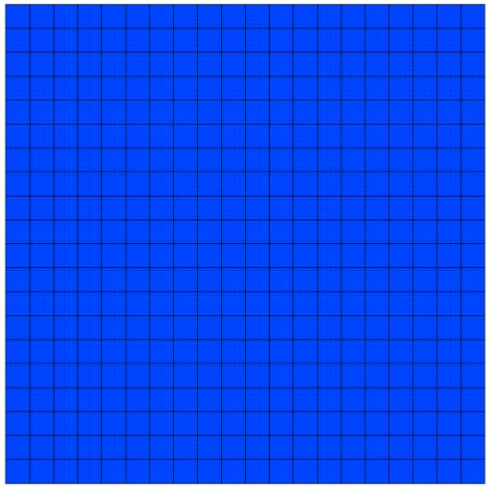
CHARTHER procedure

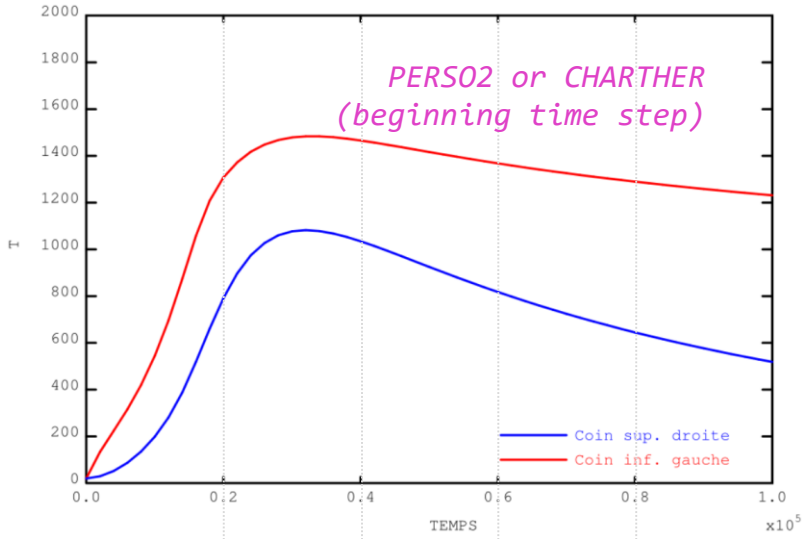
```
LT       = PROG -1.E6  1.E6 ;
LL       = 200. + (0.3 * LT) ;
LAMBDA   = EVOL 'MANU' 'T' LT 'K' LL ;


HCONV    = EVOL 'MANU' 'TEMP' (PROG 0. TPSMAX) 'H' (PROG 10. 200.) ;



***TAB1 . 'CHARGEMENT'          = CHA1 ;
TAB1 . 'PROCEDURE_CHARTHER'  = VRAI ;
TAB1 . 'MY_DATA'             = TABL ;
TAB1 . 'MY_DATA' . 'MAIL'    = MAIL1 ;
TAB1 . 'MY_DATA' . 'MODE'    = MOD1 ;
```

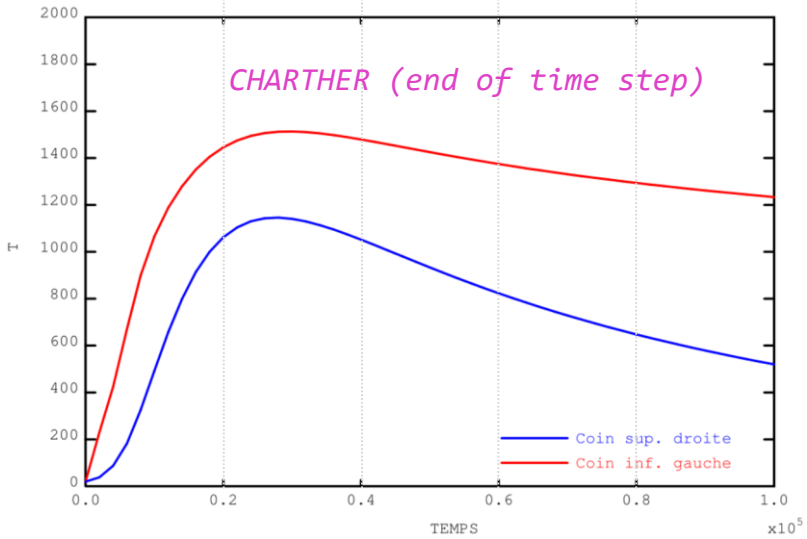
Main program

EXERCISE 3: HEAT GENERATION DEPENDING ON TEMPERATURE

Results







EXERCISE 4:

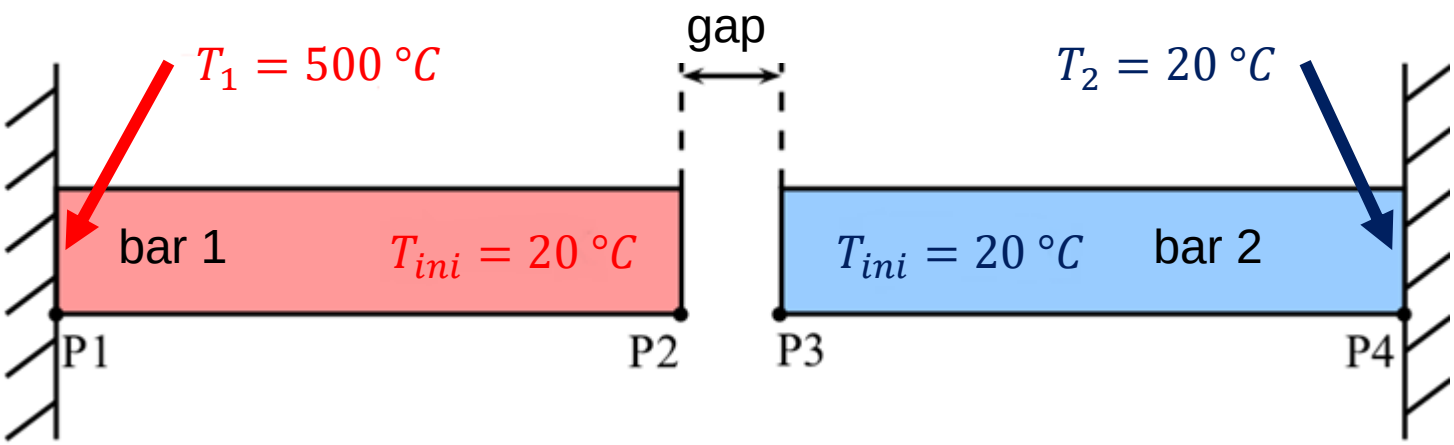
THERMO-MECHANICAL GAP CLOSING

DOWNLOAD THE STARTING FILE ON THE WEBSITE:

HTTP://WWW-CAST3M.CEA.FR/INDEX.PHP?PAGE=EXEMPLES&EXEMPLE=FORMATION_PASAPAS_4_INITIAL

EXERCISE 4: THERMO-MECHANICAL GAP CLOSING

- **Two bars expand and get in contact**
  thermal expansion
  unilateral mechanical contact



- **Purpose: add thermal contact**, this is to say the heat transfer when the mechanical contact takes place.

It's up to you!

EXERCISE 4: THERMO-MECHANICAL GAP CLOSING

A few indications …

- **Useful objects**

P2 et P3: points on the left/right of the gap
L2 et L3: lines on the left/right of the gap

- **Useful operators**

COOR: points coordinates
RELA: relation between degrees of freedom

- **Modify thermal boundary conditions of WTABLE**

EXERCISE 4: THERMO-MECHANICAL GAP CLOSING

Solution

- check the mechanical + thermal convergence
- create a new constraint: relation between the temperatures
  T on line L2 = T on line L3 with **RELA** operator
- use **REEV_MEC** to modify the thermal boundary conditions
- calculate the gap at iteration time
- modify thermal boundary conditions in **WTABLE** as a function of the gap

```
DEBP REEV_MEC T1*'TABLE' N1*'ENTIER' ;
  U1 = T1 . 'ESTIMATION' . 'DEPLACEMENTS' ;
  WT = T1 . 'WTABLE' ;
  TAM = T1 . 'MY_DATA' ;
* calcul du jeu
  PT2 = TAM . 'POINT_2' ;
  PT3 = TAM . 'POINT_3' ;
  X2 = (COOR 1 PT2) + (EXTR U1 'UX' PT2) ;
  X3 = (COOR 1 PT3) + (EXTR U1 'UX' PT3) ;
  J1 = X3 - X2 ;
* si jeu ferme, on utilise le blocage thermique initial + le RELA
  SI (J1 <EG 1.E-15) ;
    WT . 'BLOCAGES_THERMIQUES' = (TAM . 'BLOQ_0') ET
                                 (TAM . 'BLOQ_1') ;
* si jeu ouvert, on utilise le blocage thermique initial seul
  SINON ;
    WT . 'BLOCAGES_THERMIQUES' = TAM . 'BLOQ_0' ;
  FINSI ;
FINP ;
```

REEV_MEC Procedure

```
TAB1 . 'CONVERGENCE_MEC_THE' = VRAI ;
TAB1 . 'PROCEDURE_REEV_MEC'  = VRAI ;
TAB1 . 'MY_DATA'             = TABL ;
TAB1 . 'MY_DATA' . 'POINT_2' = P2 ;
TAB1 . 'MY_DATA' . 'POINT_3' = P3 ;
TAB1 . 'MY_DATA' . 'BLOQ_0'  = CL_TH ;
TAB1 . 'MY_DATA' . 'BLOQ_1'  = RELA 'T' L2 - 'T' L3 ;
```

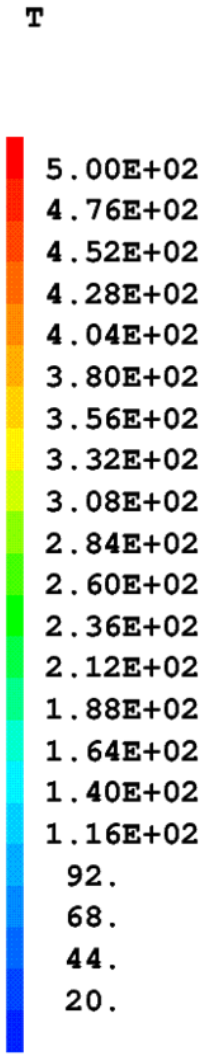
Main program

EXERCISE 4: THERMO-MECHANICAL GAP CLOSING

Results

Without contact: bars are insulated

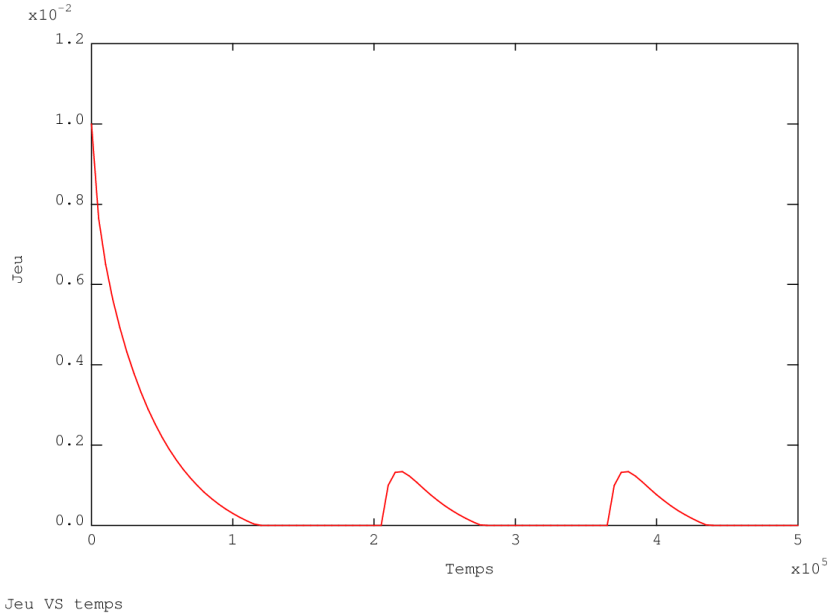
During contact: "perfect" conduction



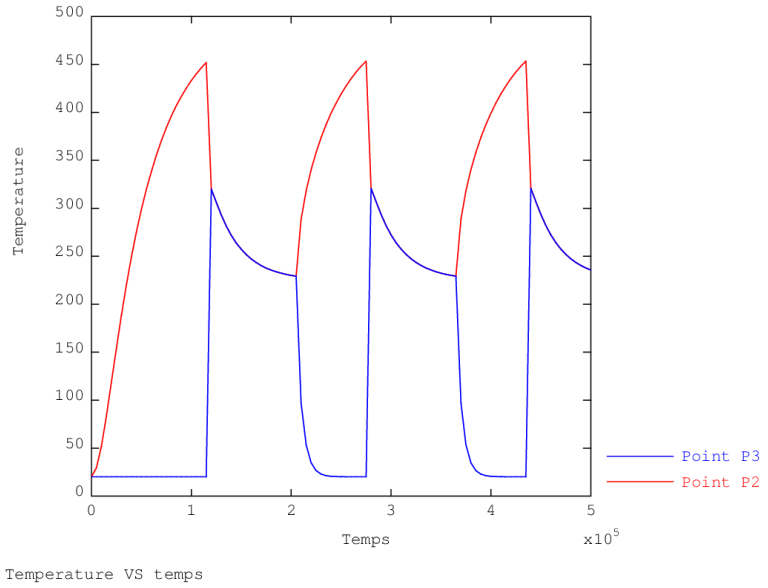
Temperature au temps 0.00000E+00

EXERCISE 4: THERMO-MECHANICAL GAP CLOSING

Results



Gap distance as a function of time



Temperatures as a function of time on the left side and on the riht side of the gap

- Changes made in **REEV_MEC** are taken into account at the same time step (due to thermo-mechanics convergence loop **BO_BOTH**)
- It will be different with **PERSO1**. Changes are taken into account at the next time step

EXERCISE 4: THERMO-MECHANICAL GAP CLOSING

Solution (bis)

- convective heat transfer in a transition element
- use of an heat transfer coefficient depending on pressure
- pressure is described through a loading named 'PC'
- updating of the loading 'PC' in **REEV_MEC**

```
DEBP REEV_MEC T1*'TABLE' N1*'ENTIER' ;
  R1 = T1 . 'ESTIMATION' . 'REACTIONS' ;
  WT = T1 . 'WTABLE' ;
  TAM = T1 . 'MY_DATA' ;
  L3 = TAM . 'LIGNE_3' ;
  MR = TAM . 'MAIL' ;
  SI ((T1 . 'ESTIMATION' . 'TEMPS') NEG 0.) ;
* calcul de la pression de contact
    PARA_P = (MAXI 'ABS' (RESU (REDU R1 L3))) / (MESU L3) ;
* chargment decrivant le parametre 'PC'
    CH_PC = MANU 'CHPO' MR 'PC' PARA_P ;
    EV1 = TAM . 'EVOL' ;
    CHA3 = CHAR 'PC' CH_PC EV1 ;
* on ecrase le chargement global de WTABLE
    WT . 'CHARGEMENT' = (TAM . 'CHAR_0') ET CHA3 ;
  FINS ;
FINP ;
```

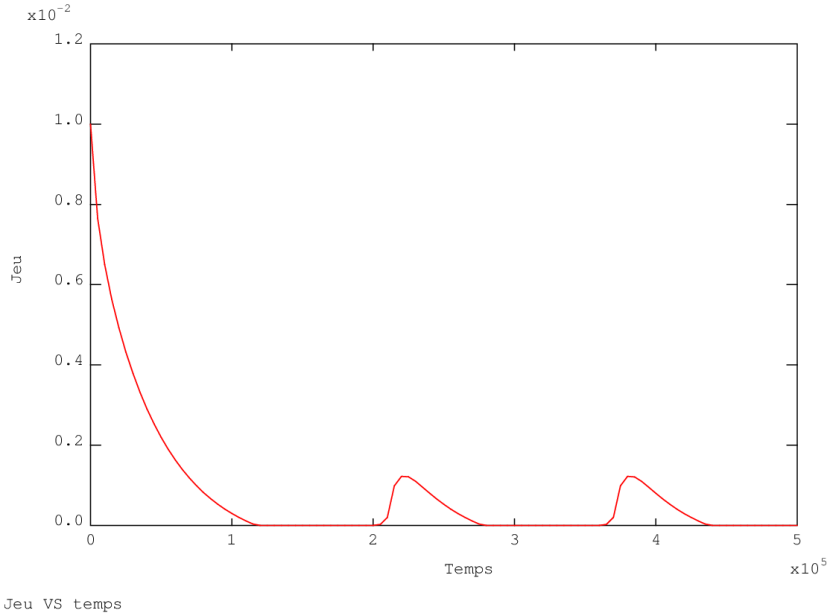
REEV_MEC procedure

```
** Modele de convection variable entre les barreaux
MRACC = RACC (1.1 * JEU_INI) L2 L3 ;
MODRACC = MODE MRACC 'THERMIQUE' 'CONVECTION' ;
HVSP = EVOL 'MANU' 'PC' (PROG 0. 5.E8 5.1E8)
                    'H'  (PROG 0. 1.E4 1.E4) ;
MATRACC = MATE MODRACC 'H' HVSP ;
MOD_TH = MODT1 ET MODT2 ET MODRACC ;
MAT_TH = MATT1 ET MATT2 ET MATRACC ;


** Chargement initial decrivant la pression de contact
CH_PC = MANU 'CHPO' MRACC 'PC' 0. ;
CHA2 = CHAR 'PC' CH_PC EV1 ;


TAB1 . 'CHARGEMENT'          = CHA1 ET CHA2 ;
TAB1 . 'CONVERGENCE_MEC_THE' = VRAI ;
TAB1 . 'PROCEDURE_REEV_MEC'  = VRAI ;
TAB1 . 'MY_DATA'             = TABL ;
TAB1 . 'MY_DATA' . 'LIGNE_3' = L3 ;
TAB1 . 'MY_DATA' . 'MAIL'    = MRACC ;
TAB1 . 'MY_DATA' . 'EVOL'    = EV1 ;
TAB1 . 'MY_DATA' . 'CHAR_0'  = CHA1 ;
```

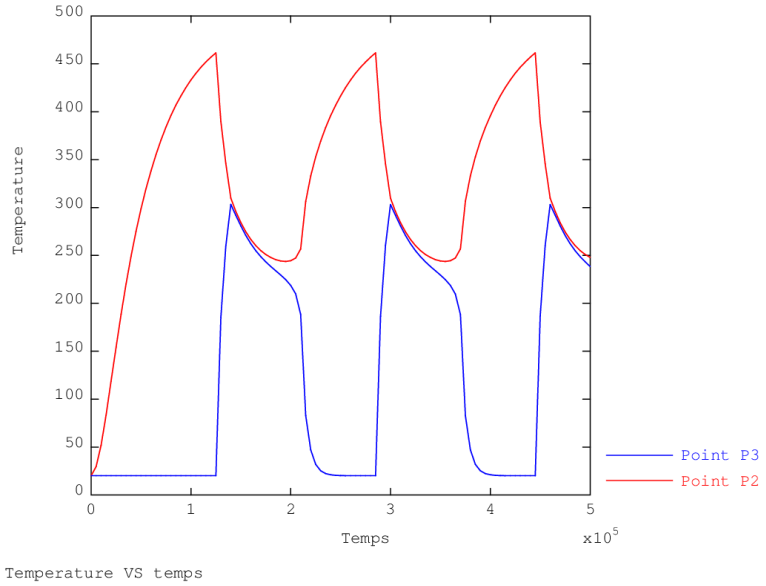
Main program

EXERCISE 4: THERMO-MECHANICAL GAP CLOSING

Results (bis)



Gap distance as a function of time



Temperatures as a function of time on the left side and on the riht side of the gap

- The heat transfer is made with different temperatures over the gap
- The temperature evolution is smoother

ANNEXES

ANNEX: ALGORITHMIC CONTROL PARAMETERS

General

NB_BOTH (ENTIER)	*Maximum number of iterations for the thermo-mechanical convergence loop*
MAXITERATION (ENTIER)	*Maximum number of iterations (**49**)*

Mechanics

PRECISION (FLOTTANT)	*Criterion to compare the imbalance (**1.E-4**)*
FTOL (FLOTTANT)	*Tolerance for strength equilibrium*
MTOL (FLOTTANT)	*Tolerance for moments equilibrium*
GRANDS_DEPLACEMENTS (LOGIQUE)	*Reference configuration = deformed configuration*
PREDICTEUR = 'HPP'	*Small displacements hypothesis → 1st convergence then use the large displacements hypothesis → 2nd convergence*
PRECISINTER (FLOTTANT)	*Precision for the constitutive laws integration local problem (**1.E-8**)*
CONVERGENCE_FORCEE (LOGIQUE)	*Forced convergence use or not in case of non convergence (**VRAI**)*
MAXSOUSPAS (ENTIER)	*Max. number of sub-steps during forced convergence (**200**)*
DELTAITER (ENTIER)	*Number of steps over which non convergence is tested*

ANNEX: ALGORITHMIC CONTROL PARAMETERS

Thermal

PROCEDURE_THERMIQUE (MOT)

Procedures to call:

- ***NONLINEAR*** *TRANSNON procedure*
- *LINEAR* *TRANSLIN procedure*
- *DUPONT* *DUPONT2 procedure*

RELAXATION_THETA (FLOTTANT)

*Relaxation coefficient for the θ-method (**1**)*

Thermo-mechanical coupling

CONVERGENCE_MEC_THE (LOGIQUE)

*Indicates that the thermo-mechanical loop should be repeated in case of dependence (**FAUX**)*

CRITERE_COHERENCE (FLOTTANT)

*Precision for the thermo-mechanical convergence, tested on thermal results (**1.E-2**)*

PROJECTION (LOGIQUE)

*Indicates that the thermal and mechanical meshes are different while the problem is coupled (**FAUX**)*

ANNEX: IMBALANCE MINIMIZATION IN GIBIANE

- Simplified **UNPAS** algorithm (see the **@SOLVMEC** procedure) [small displacements / strains]

```
fex1 = TIRE char tps1 'MECA' ;
flx1 = TIRE char tps1 'DIMP' ;
resi0 = fex1 + fr0 - fi0 + flx1 - flx0 ;
fref = MAXI 'ABS' (fex1 + fr0) ;
k = (RIGI mod mat) ET blo ;
REPE b1 100 ;
  ddu = RESO k resi0 ;
  u1 = u0 + ddu ;
  fr1 = REAC blo u1 ;
  eps1 = EPSI mod u1 ;
  sig1 = ... COMP eps0 eps1 ...
  fi1 = BSIG mod sig1 ;
  resi1 = fex1 + fr1 - fi1 ;
  crit1 = (MAXI 'ABS' resi1) / fref ;
  SI (crit1 < precis) ;
    QUIT b1 ;
  FINSI ;
  u0 = u1 ;
  resi0 = resi1 ;
FIN b1 ;
```

ANNEX: IMBALANCE MINIMIZATION IN GIBIANE

Simplified UNPAS algorithm (see the @SOLVMEC procedure) [large displacements, updated lagrangian]

```
fex1 = TIRE char tps1 'MECA' ;
flx1 = TIRE char tps1 'DIMP' ;
resi0 = fex1 + fr0 - fi0 + flx1 - flx0 ;
fref = MAXI 'ABS' (fex1 + fr0) ;
conf0 = FORM u0 ;
u00 = u0 ;
k = (RIGI mod mat) ET blo ET (KSIG mod sig0) ;
REPE b1 100 ;
  ddu = RESO k resi0 ;
  u1 = u0 + ddu ;
  fr1 = REAC blo u1 ;
  du = u1 - u00 ;
  deps = EPSI mod du ;
  eps1 = eps0 + deps ;
  sig1 = ... (COMP eps0 eps1, EXCO, ...)
  sig1 = PICA mod sig1 du ;
  FORM u1 ;
  fi1 = BSIG mod sig1 ;
  FORM conf0 ;
  resi1 = fex1 + fr1 - fi1 ;
  crit1 = (MAXI 'ABS' resi1) / fref ;
  SI (crit1 < precis) ;
    QUIT b1 ;
  FINSI ;
  u0 = u1 ;
  resi0 = resi1 ;
FIN b1 ;
```

moving to the current configuration (beginning of time step)

strain increment

moving stresses to the final configuration

integration on the final configuration

ANNEX: CONVERGENCE CRITERIA (UNPAS)

Convergence norms (after 1st RESO)

$$F^{ref} = \frac{\left|\delta\{U\}_1^1.\{F\}_1^{S+V} - \{\lambda\}_1^1.\left(\{d\}_1 - [A]^T.\{U\}_1^0\right)\right|}{\max\limits_{\text{dof depl}}\left|\delta\{U\}_1^1\right| + \text{xsmall}} + \frac{\max\limits_{\text{dof depl}}\left|\delta\{U\}_1^1\right|.\max\left|[B]\{\sigma\}_1^0\right|}{\max\limits_{\text{dof depl}}\left|\delta\{U\}_1^1\right| + \text{xsmall}} + \max\left|[B]\{\sigma\}_1^0\right|$$

$$M^{ref} = \frac{\left|\delta\{U\}_1^1.\{F\}_1^{S+V} - \{\lambda\}_1^1.\left(\{d\}_1 - [A]^T.\{U\}_1^0\right)\right|}{\max\limits_{\text{dof rota}}\left|\delta\{U\}_1^1\right| + \text{xsmall}} + \frac{\max\limits_{\text{dof depl}}\left|\delta\{U\}_1^1\right|.\max\left|[B]\{\sigma\}_1^0\right|}{\max\limits_{\text{dof rota}}\left|\delta\{U\}_1^1\right| + \text{xsmall}} + \text{xsmall}$$

Unbalance measure (at each UNPAS iteration)

$$\mathbf{XCONV} = \frac{\max\left\{\max\limits_{\text{ddl depl}}\left|\{F\}_1^{S+V} - [A]^T.\{\lambda\}_1^{i+1} - [B]\{\sigma\}_1^{i+1}\right| ; \max\left|[A]^T.\left(\{\lambda\}_1^{i+1} - \{\lambda\}_1^{i}\right)\right|\right\}}{F^{ref}}$$

$$\mathbf{XCONVM} = \frac{\max\limits_{\text{ddl rota}}\left|\{F\}_1^{S+V} - [A]^T.\{\lambda\}_1^{i+1} - [B]\{\sigma\}_1^{i+1}\right|}{M^{ref}}$$

ANNEX: CONVERGENCE CRITERIA (UNPAS)

- Measure of strain increment variation between 2 iterations

$$\mathbf{DEPSTDM} = \max\left|\Delta\{\varepsilon\}_1^{i+1} - \Delta\{\varepsilon\}_1^{i}\right|$$

- Convergence is reached if

$$\begin{aligned}\mathbf{XCONV} &< \texttt{PRECISION}\\ \mathbf{XCONVM} &< \texttt{PRECISION}\\ \mathbf{DEPSTDM} &< \texttt{PRECISION}\end{aligned}$$

ANNEX: FORCED CONVERGENCE

TO BE FINISHED…

- **If non convergence is confirmed, the step is re-calculated:**

- state at the beginning of the step is redefined
- the last "out of equilibrium" state with minimal imbalance is chosen
- at the beginning of time step, are overwritten:
  - the instant
  - the material properties
  - the displacements
  - the internal variables
  - the inelastic strains
  - the stresses

- UNPAS is re-started with a time-step equal to zero
- no load increment, only disequilibrium

- allows to find a solution with the right behavior and equilibrium but with a different loading path

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

DES/DANS
DM2S
SEMT

Etablissement public à caractère industriel et commercial | R.C.S Paris B 775 685 019