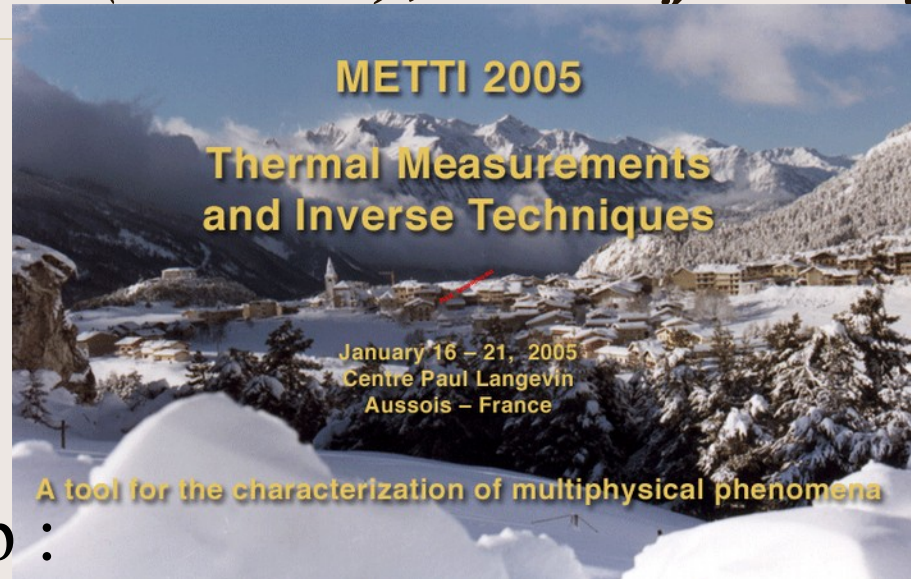


Eurotherm Winter School, METTI 2005, Aussois (France), 16-21 january 2005



Workshop :

« How to solve the inverse problems with
Femlab and Matlab environments »

By Sebastien ROUQUETTE*, Philippe LE MASSON^o, Jialin GUO^o &
Tahar LOULOU^o

*Laboratoire des Systèmes Mécaniques et Ingénierie Simultanée (LASMIS), Université de Technologie de Troyes, FRE
CNRS 2719, 12 Rue Marie Curie, 10010 TROYES Cedex.

^oLaboratoire d'Etudes thermophysiques, Energétiques et Environnement. Equipe Etude Thermophysique des Matériaux.
Centre de recherche de l'Université de Bretagne Sud, Rue saint Maudé, 56321 LORIENT Cédex.

sebastien.rouquette@utt.fr; philippe.le-masson@univ-ubs.fr; jialin.guo@univ-ubs.fr; tahar.loulou@univ-ubs.fr

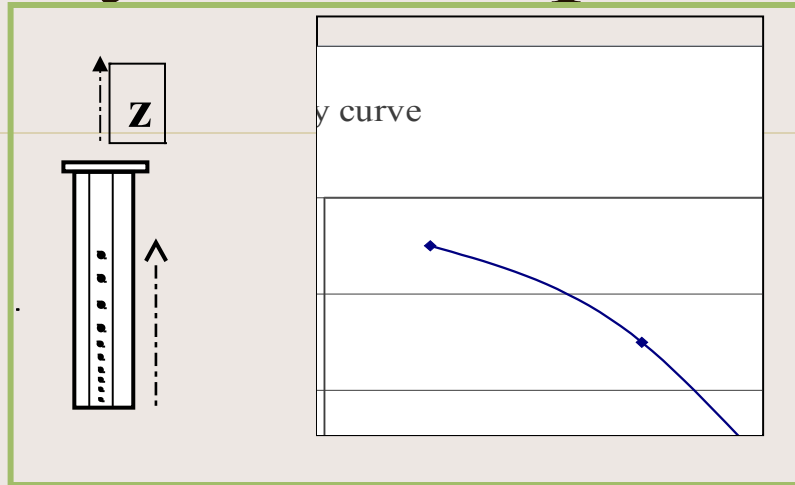
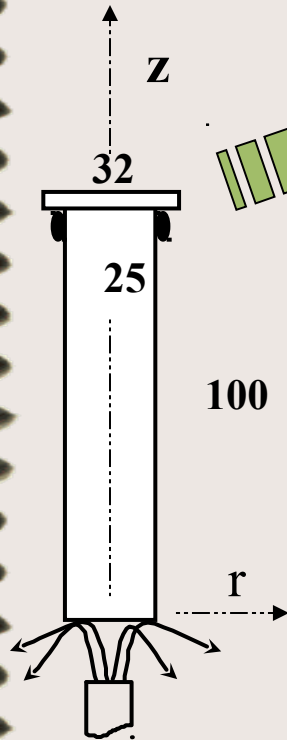
Goals

- We use Levenberg Marquardt method for parameter estimation
- We use Femlab for the direct and sensitivity problem definitions
- We save these problems in a matlab file (« *.m »)
- We introduce the algorithm in this file
- At last the resolution of the inverse problem is realised with Matlab.

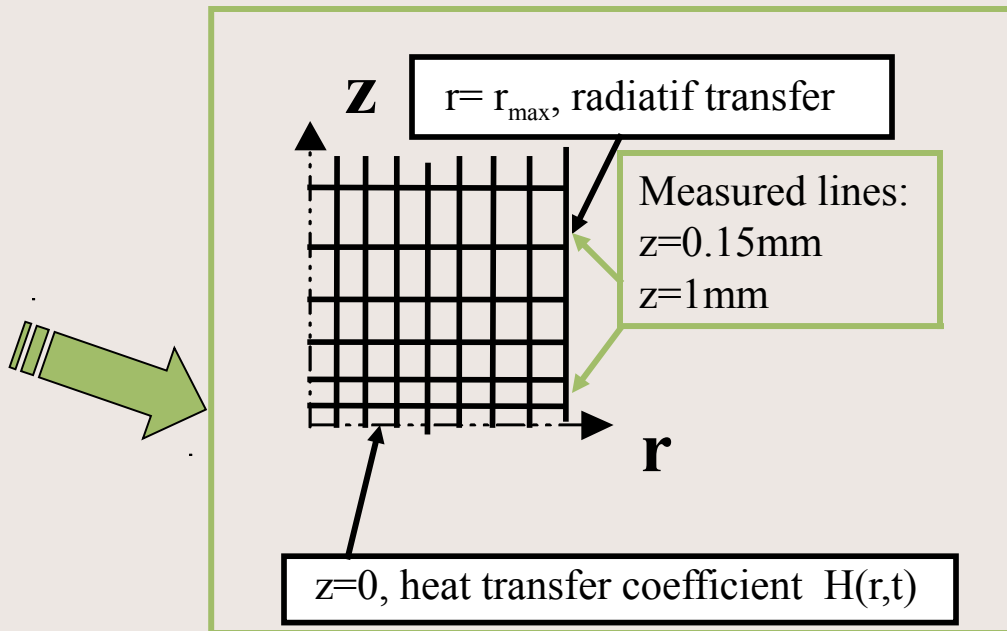
CONTENTS

- Resolution of the direct quenching problem with femlab
- The Levenberg-Marquardt algorithm
- Resolution of the sensitivity problem with femlab
- Resolution of the inverse boundary problem with matlab
- Conclusions

The « Jominy » end-quench test



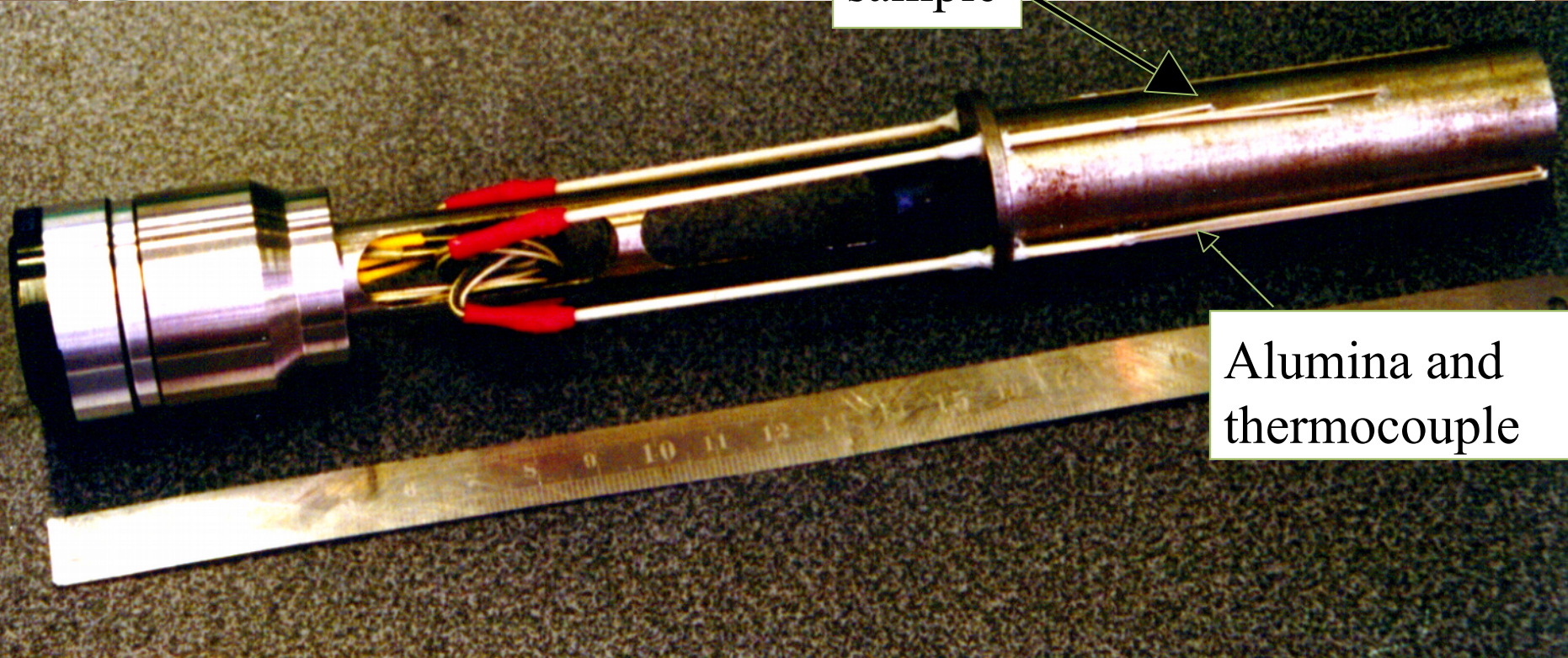
*Revue de
Métallurgie,
sept. 2000, Le
Masson et al.*



The sample

sample

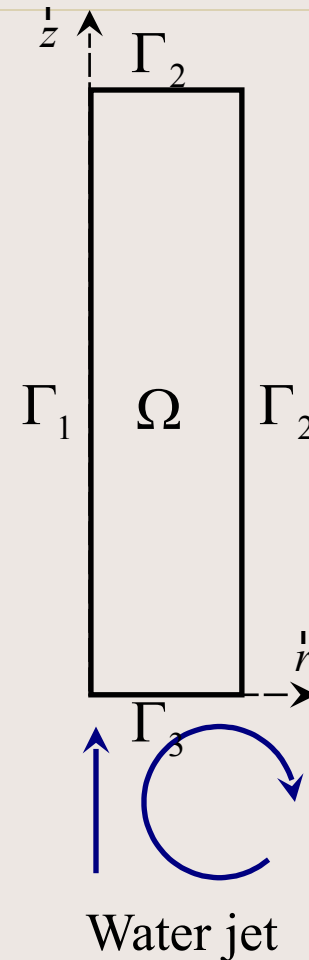
Alumina and
thermocouple



The direct quenching problem

- The governing equations [1] :

$$\begin{aligned}\rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (k \cdot \nabla T) &= 0 \in \Omega \times T \\ -k \frac{\partial T}{\partial \vec{n}} &= 0 \text{ on } \Gamma_2 \times T \\ -k \frac{\partial T}{\partial \vec{n}} &= \varepsilon \sigma (T^4 - T_{room}^4) \text{ on } \Gamma_2 \times T \\ -k \frac{\partial T}{\partial \vec{n}} &= Hcv (T - T_{room}) \text{ on } \Gamma_3 \times T \\ T(r, z; t=0) &= 1173 K \in \Omega\end{aligned}$$



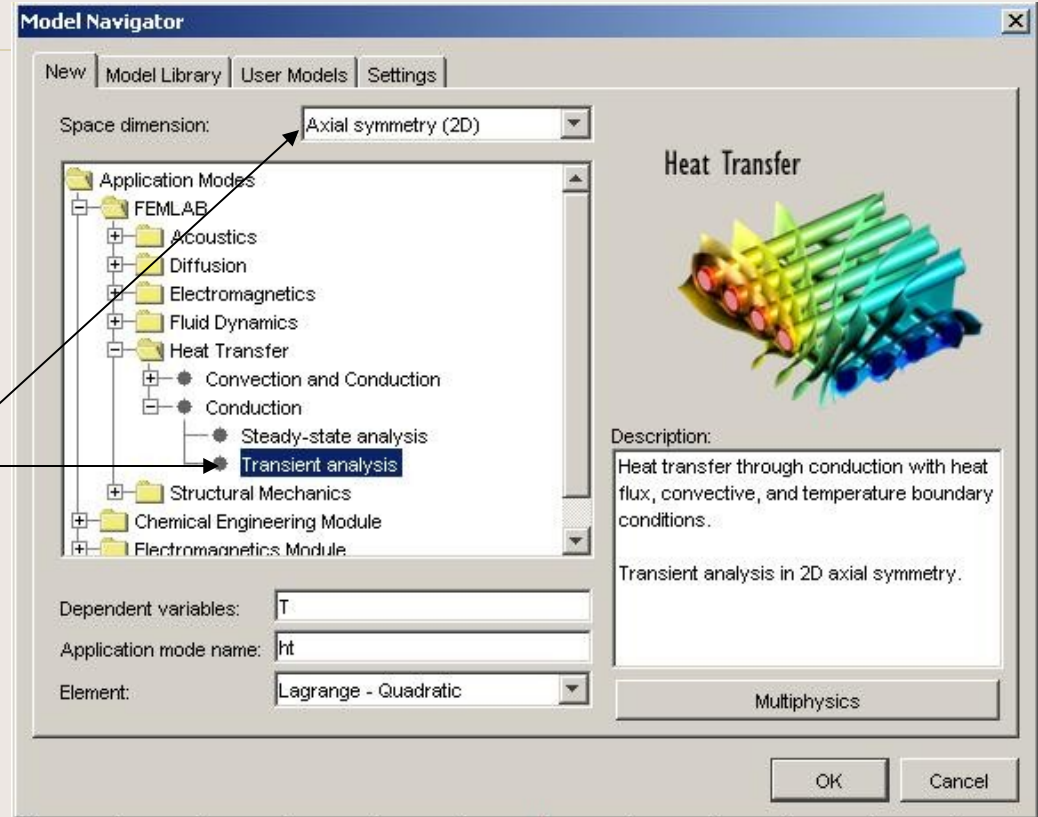
[1] Essai de trempabilité jominy. Norme NF A 04-303 (juin 1979).

[2] P. Le Masson, P. Rogeon, D. Carron and J.J. Quémener. Simulation numérique thermométallurgique : influence des paramètres d'entrée sur la modélisation d'un essai de trempabilité Jominy. La revue de métallurgie-CIT/ Science et génie des matériaux, p.1055-1064, septembre 2000.

The direct quenching problem

- the resolution with femlab -

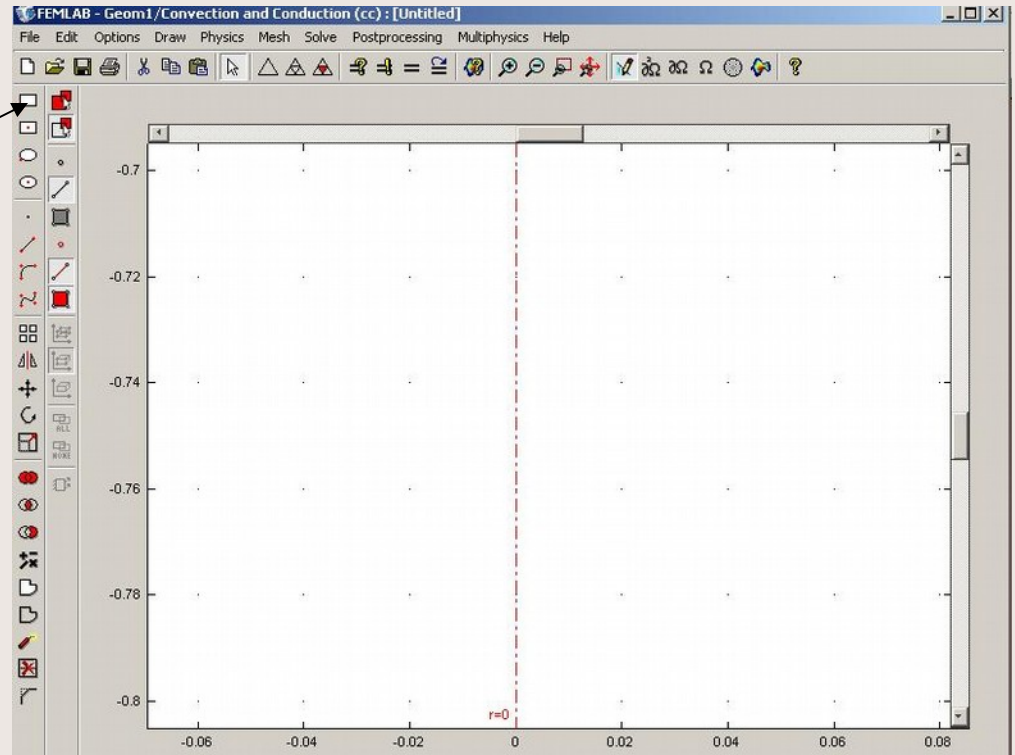
- Open FEMLAB3.1 with matlab
- In the model navigator window, choose « this configuration »
- And click OK



The direct quenching problem

- the resolution with femlab -

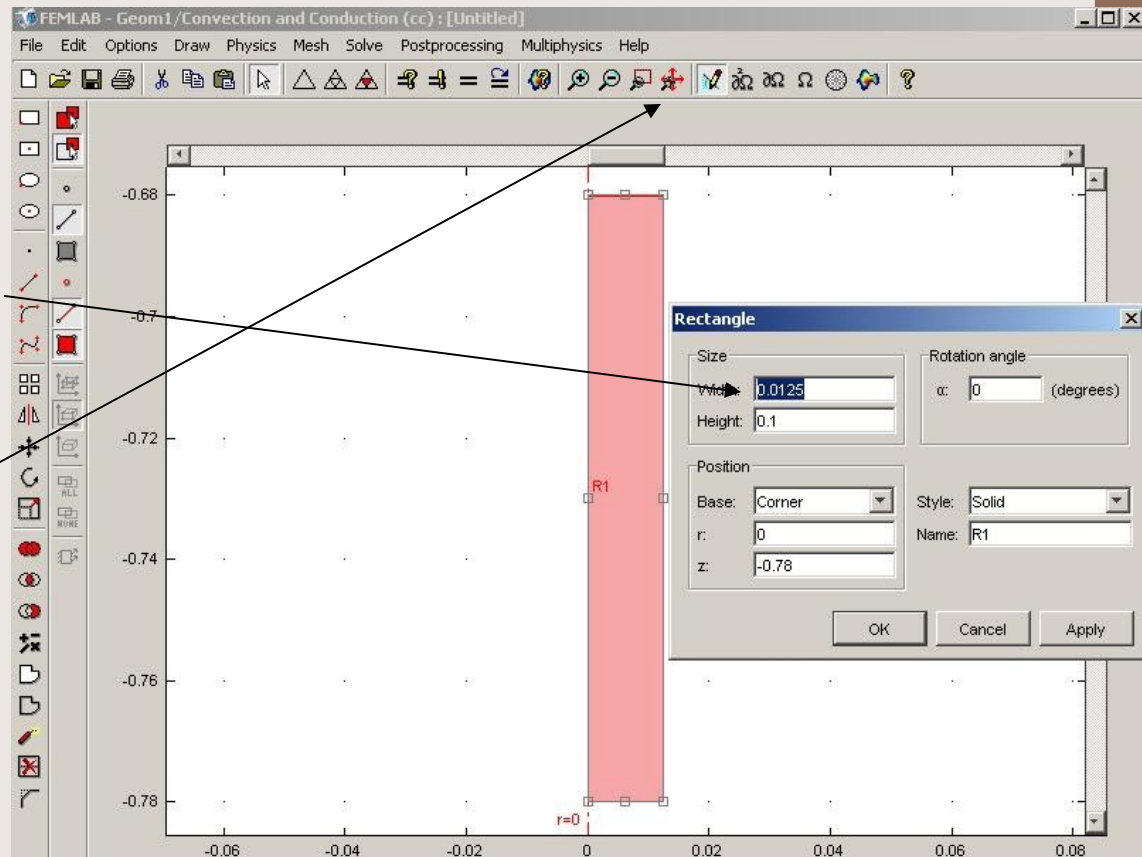
- To draw a rectangle of 100mm x 12,5mm
- Click on the « rectangle icon »
- Draw a rectangle in the draw window



The direct quenching problem

- the resolution with femlab -

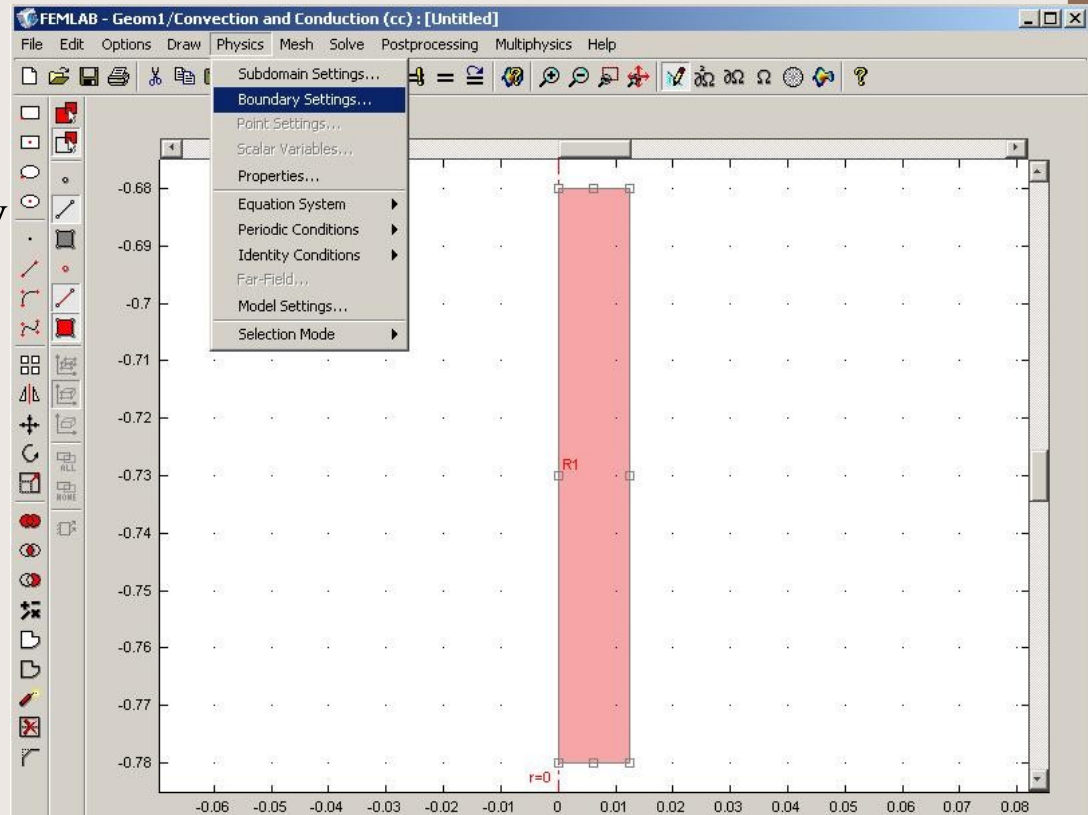
- Click two times on the rectangle and the object properties window appears
- Correct the dimensions
- APPLY and OK
- Redimension the rectangle with the « Zoom extents icon »



The direct quenching problem

- the resolution with femlab -

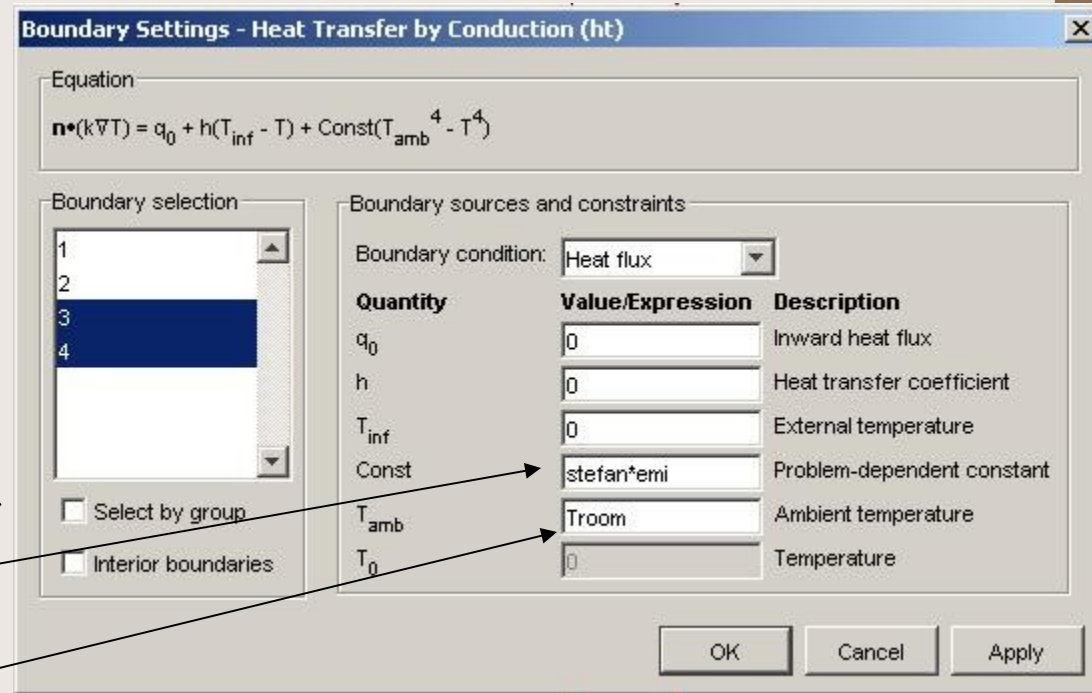
- In bar menu, choose «Physics» then «boundary settings» to define the boundary conditions.



The direct quenching problem

- the resolution with femlab -

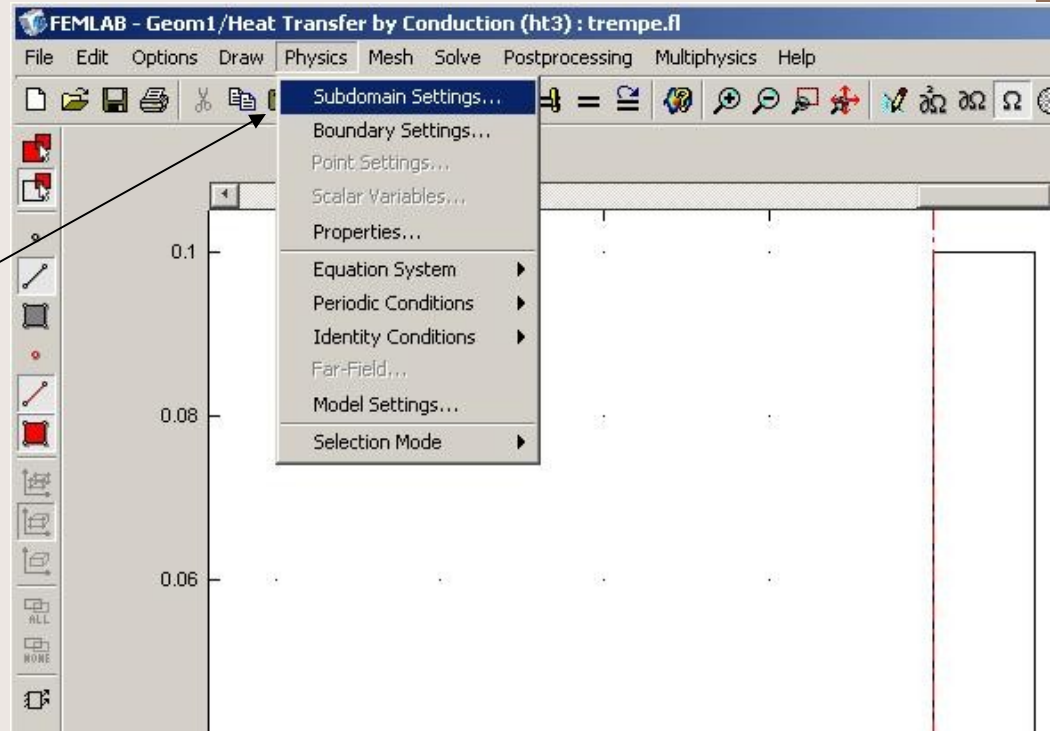
- Boundary 1: Axial symmetry
- Boundary 2: select heatflux $h=HCV$ and $T_{inf} = 'Troom'$,
- Boundaries 3 and 4: select heat flux « const » = 'stefan*emi' and « T_{amb} » = 'Troom'
- APPLY and click OK



The direct quenching problem

- the resolution with femlab -

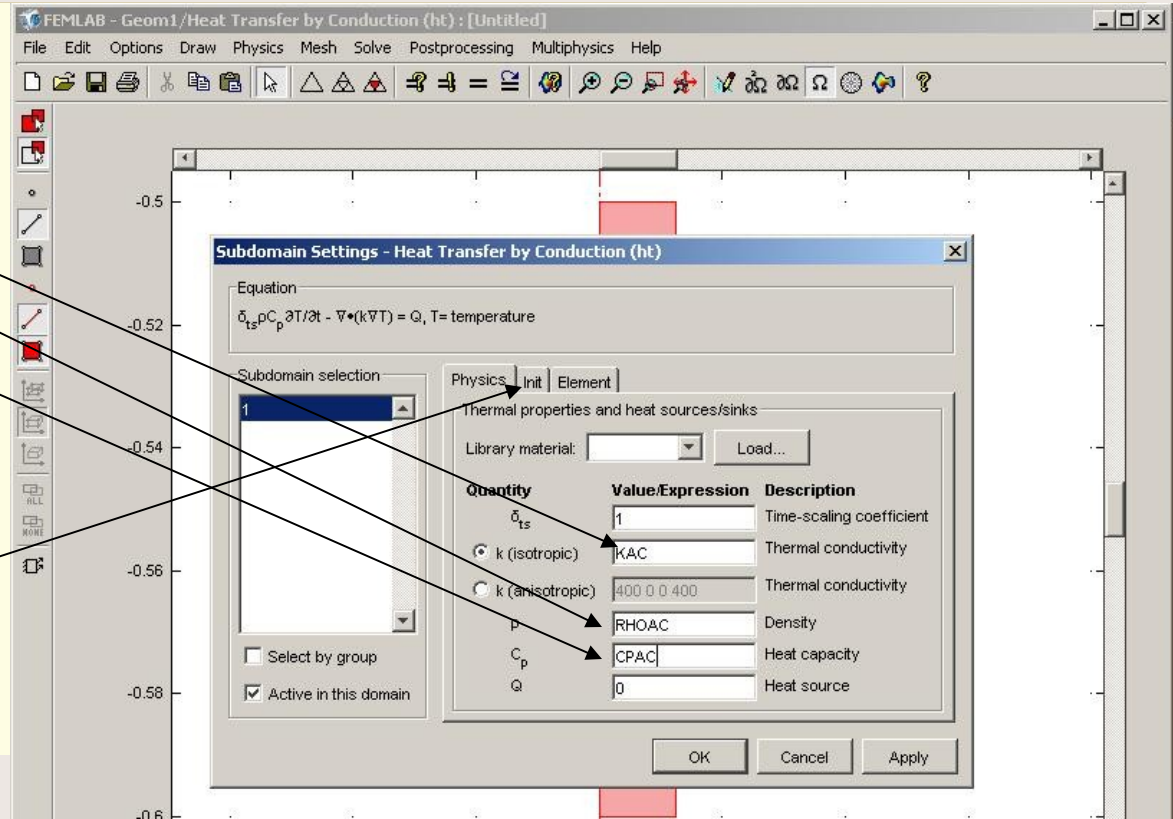
- In « Physics » menu bar, choose « subdomain settings » to define the material properties.
- The subdomain settings window appears.



The direct quenching problem

- the resolution with femlab -

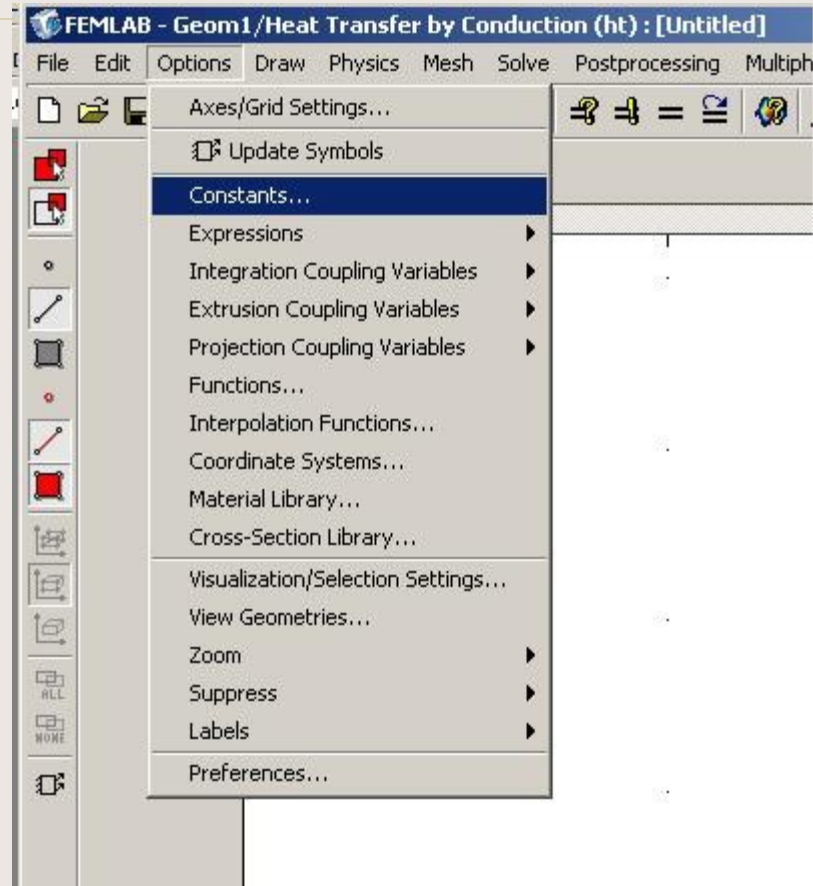
- In the coefficients domain selection, enter:
KAC for the thermal conductivity,
RHOAC for the Density,
CPAC for the heat capacity,
0 for the heat source;
- In the init part, give the initial temperature : Tsteel
- OK



The direct quenching problem

- the resolution with femlab -

- Go to the « options » in the menu bar, choose « constants » to define all the parameters and their values.



The direct quenching problem

- the resolution with femlab -

- To validate the name of the parameter click « apply » to validate his value.

$$RHOAC = 7500 \text{ kg} / \text{m}^3$$

$$CPAC = 520 \text{ J} / \text{kg} / \text{C}$$

$$KAC = 32 \text{ W} / \text{C} / \text{m}$$

$$Troom = 20 \text{ C}$$

$$HCV = 15000 \text{ W} / \text{m}^2 / \text{C}$$

$$emi = 0.6$$

$$stefan = 5.67e-8$$

$$Tsteel = 900 \text{ C}$$

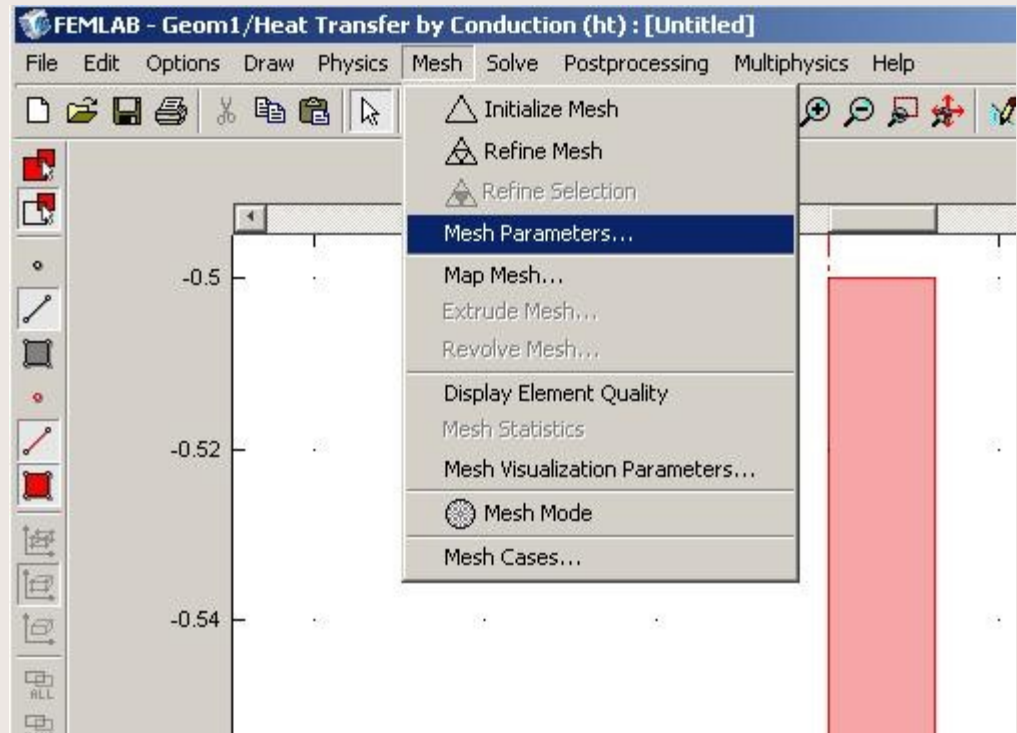
Name	Expression	Value
RHOAC	7500	7500
CPAC	520	520
KAC	32	32
Tsteel	900	900
Troom	20	20
HCV	15000	15000
emi	0.6	0.6
stefan	5.67e-8	5.67e-8

- To exit: click OK

The direct quenching problem

- the resolution with femlab -

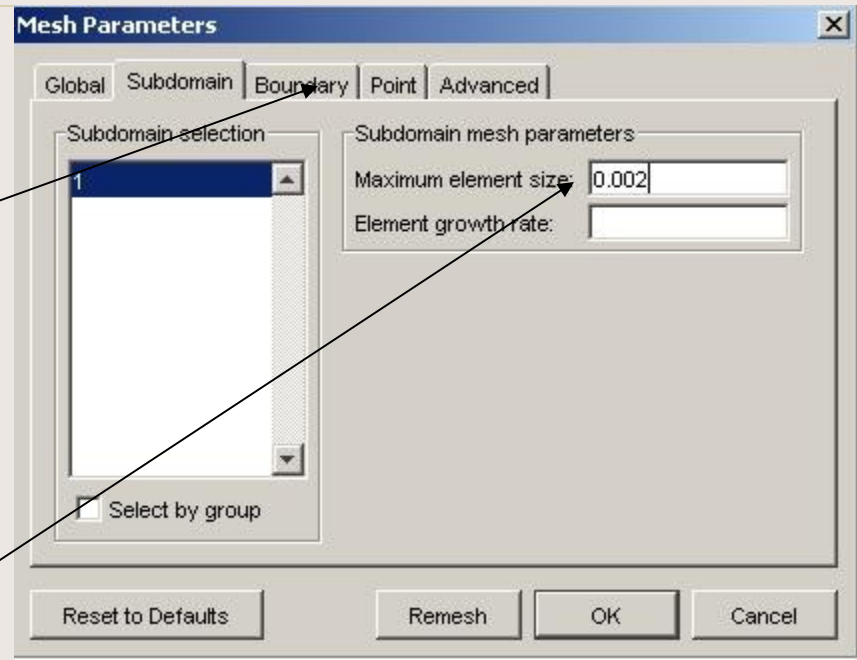
- Mesh step
- In the menu, select « mesh » then « mesh parameters » to open the mesh parameters window



The direct quenching problem

- the resolution with femlab -

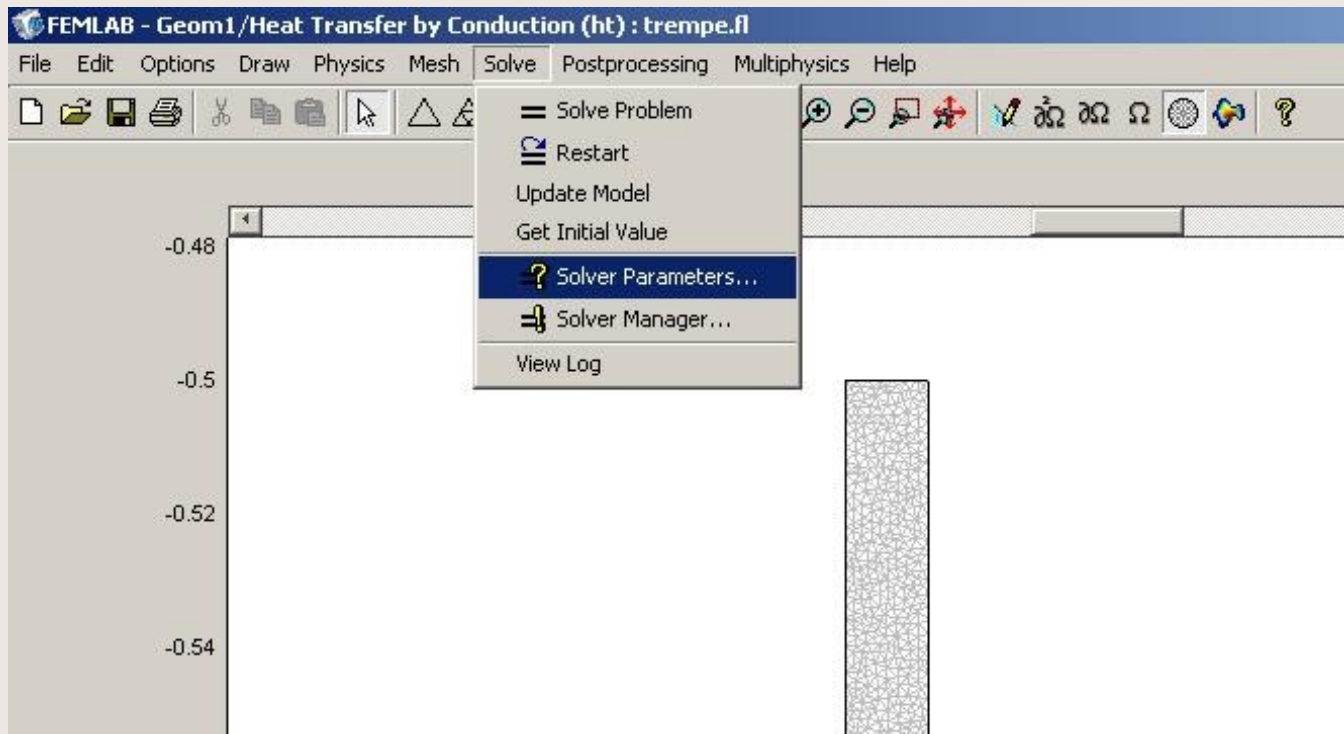
- Change the « mesh growth rate »
- Refine the mesh near the quenched boundary 2 : « maximum element size » = $5e-4$
- Impose the size of the element in the domain 1 : « maximum element size » = $2e-3$
- And click the button « REMESH »



The direct quenching problem

- the resolution with femlab -

- Select « solve » in the bar menu,
- Then « solve parameters » and click
- The solver parameters window appears

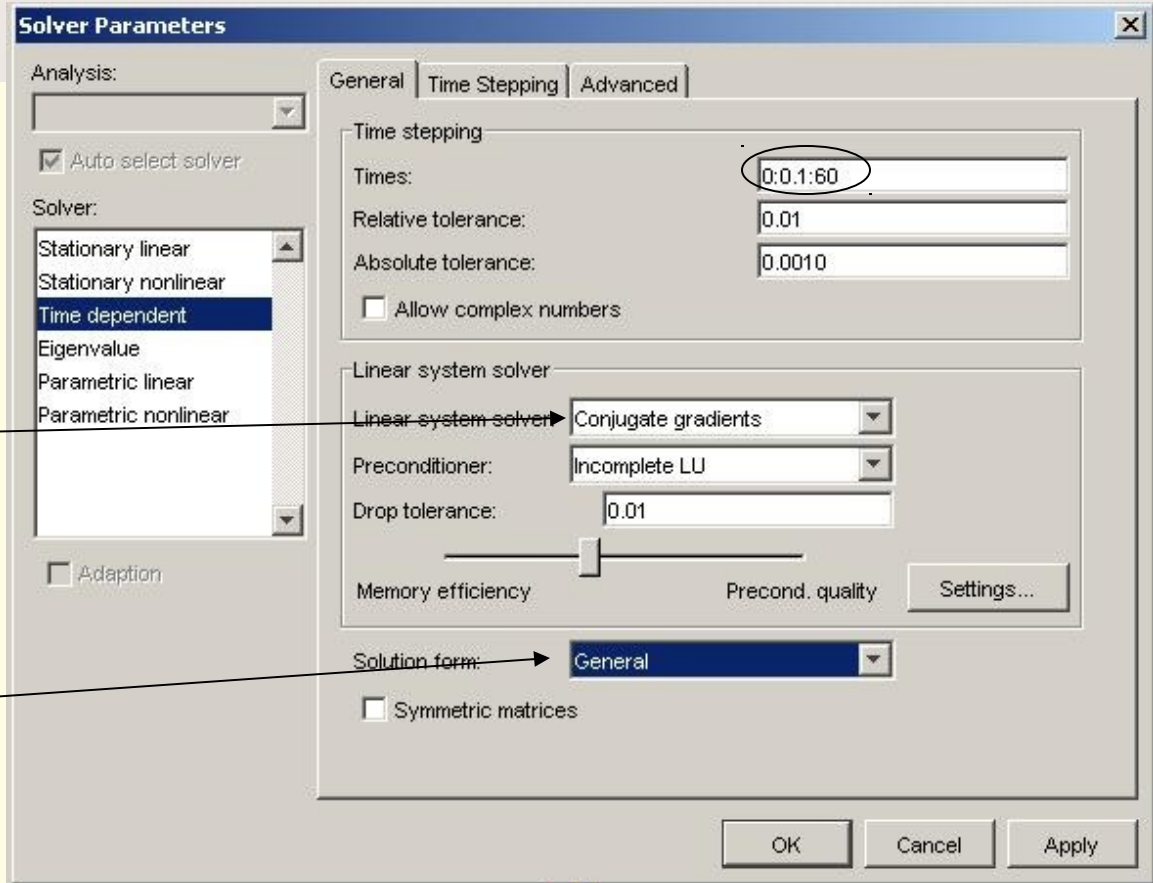


The direct quenching problem

- the resolution with femlab -

- In the « general » menu, verify that is a ‘time dependent’ problem in « solver type »

- In « solution form » select ‘general’



The Levenberg-Marquardt method

- the inverse boundary problem formulation -

- The inverse boundary problem formulation [3]:

Find the parameter $Z=\{H,E\}$ which minimizes the quadratic criterion $S(Z,T)$:

$$S(\bar{Z}, T) = [Y_i - T_i(\bar{Z})]^T W [Y_i - T_i(\bar{Z})]$$

With Y_i is the measurements, T_i the calculated temperature, and W a diagonal matrix where the diagonal elements are given by the inverse of the standard deviation of the measurement errors, i is the total number of measurements.

- The Levenberg-Marquardt Method says [4,5]:

$$\bar{Z}^{k+1} = \bar{Z}^k + [J^T W J + \lambda^k \Omega^k]^{-1} J^T W [Y_i - T_i(\bar{Z}^k)]$$

with J is the sensitivity matrix, λ is the damping parameter and Ω is a diagonal matrix equal to the identity matrix.

[3] A.N. Tikhonov & V.Y. Arsenin. Solutions of ill-posed problems. V.H. Wistom & Sons, Washington, DC (1977).

[4] K. Levenberg. A method for the solution of certain non linear problems in least squares. Quart. Appli. Math. 2 (1944) 4164-168.

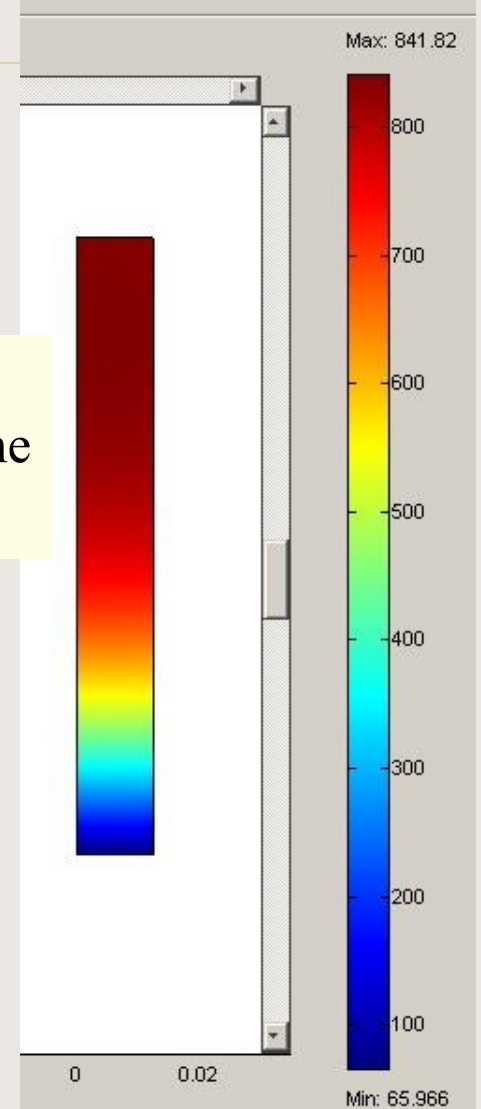
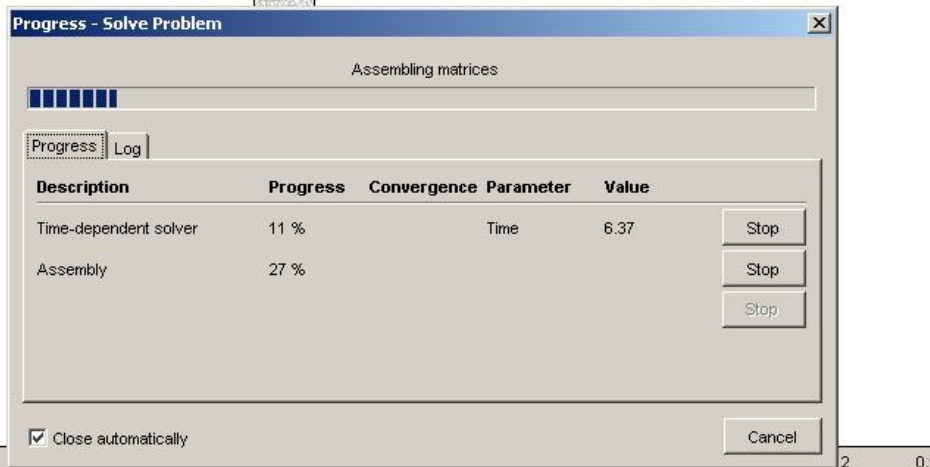
[5] D.W. Marquardt. An algorithm for least squares estimation of non linear parameters. J. soc. Ind. Appli. Math. 11 (1963) 431-441.

The direct quenching problem

- the resolution with femlab -

- Solve the direct quenching problem by using the « solve » icon (symbol equal)

- We obtain the temperature field at the final time



The Levenberg-Marquardt method

- the inverse boundary problem formulation -

- The inverse boundary problem formulation [3]:

Find the parameter $Z=\{H,E\}$ which minimizes the quadratic criterion $S(Z,T)$:

$$S(\bar{Z}, T) = [Y_i - T_i(\bar{Z})]^T W [Y_i - T_i(\bar{Z})]$$

With Y_i is the measurements, T_i the calculated temperature, and W a diagonal matrix where the diagonal elements are given by the inverse of the standard deviation of the measurement errors, i is the total number of measurements.

- The Levenberg-Marquardt Method says [4,5]:

$$\bar{Z}^{k+1} = \bar{Z}^k + [J^T W J + \lambda^k \Omega^k]^{-1} J^T W [Y_i - T_i(\bar{Z}^k)]$$

with J is the sensitivity matrix, λ is the damping parameter and Ω is a diagonal matrix equal to the identity matrix.

[3] A.N. Tikhonov & V.Y. Arsenin. Solutions of ill-posed problems. V.H. Wistom & Sons, Washington, DC (1977).

[4] K. Levenberg. A method for the solution of certain non linear problems in least squares. Quart. Appl. Math. 2 (1944) 4164-168.

[5] D.W. Marquardt. An algorithm for least squares estimation of non linear parameters. J. soc. Ind. Appl. Math. 11 (1963) 431-441.

The Levenberg-Marquardt method

- the sensitivity matrix -

- Sensitivity coefficients calculus [6]:

$$\text{First method: } J_{\bar{Z}} = \frac{\partial T(\bar{Z})}{\partial \bar{Z}} \quad \text{Second method: } J_Z = \frac{T(Z + \varepsilon Z) - T(Z - \varepsilon Z)}{2\varepsilon Z}$$

The expression of the sensitivity matrix becomes:

$$J[H, E] = \begin{bmatrix} \frac{\partial T_1}{\partial H} & \frac{\partial T_2}{\partial H} & \cdots & \frac{\partial T_I}{\partial H} \\ \frac{\partial T_1}{\partial E} & \frac{\partial T_2}{\partial E} & \cdots & \frac{\partial T_I}{\partial E} \end{bmatrix}$$

Stopping criterion:

$$S(\bar{Z}^k, T) \leq \varepsilon$$

The Levenberg-Marquardt method

- the algorithm -

- Levenberg-Marquardt Algorithm:

1-Solve the direct problem with for the unknown parameters Z^k to obtain the calculated temperatures $T(Z^k)$.

2-Compute $S(Z^k, T(Z^k))$

3-Compute the sensitivity matrix $J(Z^k)$ et the matrix Ω^k .

4-Calculate the new estimated Z^{k+1} .

5-Solve the direct problem with Z^{k+1} , Compute $S(Z^{k+1}, T(Z^{k+1}))$

6-if $S(Z^{k+1}) > S(Z^k)$ replace λ^k by $10 * \lambda^k$ and go back to step 4 else if $S(Z^{k+1}) < S(Z^k)$ replace λ^k by $0,1 * \lambda^k$ and continue.

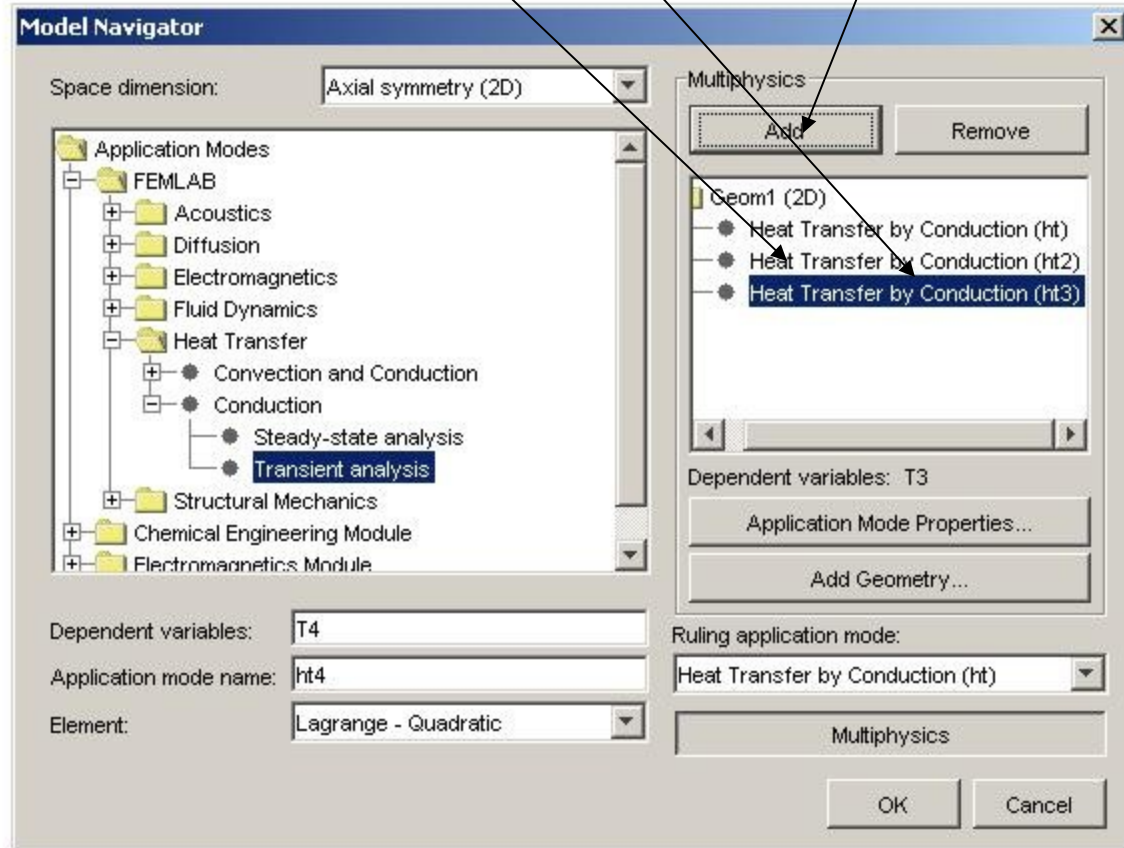
7-Test if $S(Z^{k+1}) < \varepsilon$, Stop if it is true else do $k = k + 1$ and go back to step 3.

- Let define the sensitivity problems under femlab ...

The sensitivity problems

- the resolution with femlab -

- First, in the bar menu, click on « Multiphysics » and select « Model Navigator », a window appears. Now add two new heat conduction systems : ht2 and ht3.

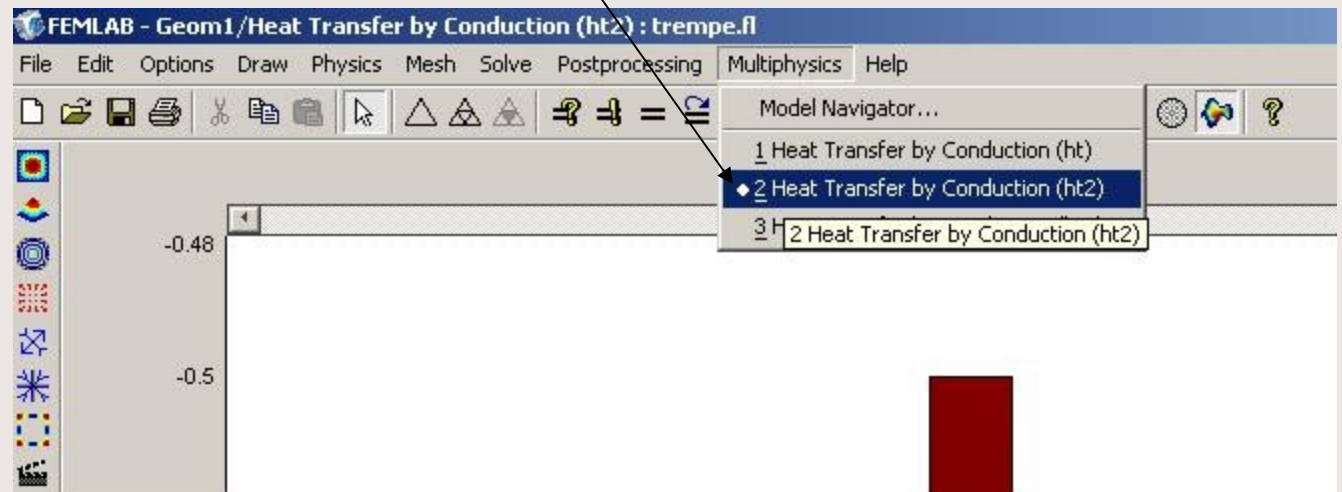


The sensitivity problems

- the resolution with femlab -

- Go to « multiphysics » and select the 2nd mode (ht2)
- In « physics » menu ... « boundary settings » to define the boundary conditions of the sensitivity problem for the convective coefficient J_{HCV} .

$$\begin{aligned}\rho C_p \frac{\partial J_{HCV}}{\partial t} - \nabla \cdot (k \cdot \nabla J_{HCV}) &= 0 \in \Omega \times T \\ -k \frac{\partial J_{HCV}}{\partial \vec{n}} &= 0 \text{ on } \Gamma_2 \times T \\ -k \frac{\partial J_{HCV}}{\partial \vec{n}} &= (T - T_{room}) + HCV * J_{HCV} \text{ on } \Gamma_2 \times T \\ -k \frac{\partial J_{HCV}}{\partial \vec{n}} &= 4 \varepsilon \sigma T^3 J_{HCV} + \sigma (T^4 - T_{room}^4) \text{ on } \Gamma_{3,4} \times T \\ J_{HCV}(r, z; t=0 s) &= 0 \in \Omega\end{aligned}$$



The sensitivity problems

- the resolution with femlab -

- Boundary selection :
 - 1 : 'axial symmetry'
 - 2 : 'heat flux'
 - 3, 4 : 'heat flux' => h : '4*emi*stefan*T^3'

$$\rho C_p \frac{\partial J_{HCV}}{\partial t} - \nabla \cdot (k \cdot \nabla J_{HCV}) = 0 \in \Omega \times T$$

$$-k \frac{\partial J_{HCV}}{\partial \vec{n}} = 0 \text{ on } \Gamma_2 \times T$$

$$-k \frac{\partial J_{HCV}}{\partial \vec{n}} = (T - T_{room}) + HCV * J_{HCV} \text{ on } \Gamma_2 \times T$$

$$-k \frac{\partial J_{HCV}}{\partial \vec{n}} = 4 \epsilon \sigma T^3 J_{HCV} + \sigma (T^4 - T_{room}^4) \text{ on } \Gamma_{3,4} \times T$$

$$J_{HCV}(r, z; t=0s) = 0 \in \Omega$$

Equation

$$\mathbf{n} \cdot (k \nabla T) = q_0 + h(T_{inf} - T) + \text{Const}(T_{amb}^4 - T^4)$$

Boundary selection

2

3

4

Select by group

Interior boundaries

Boundary sources and constraints

Boundary condition: Heat flux

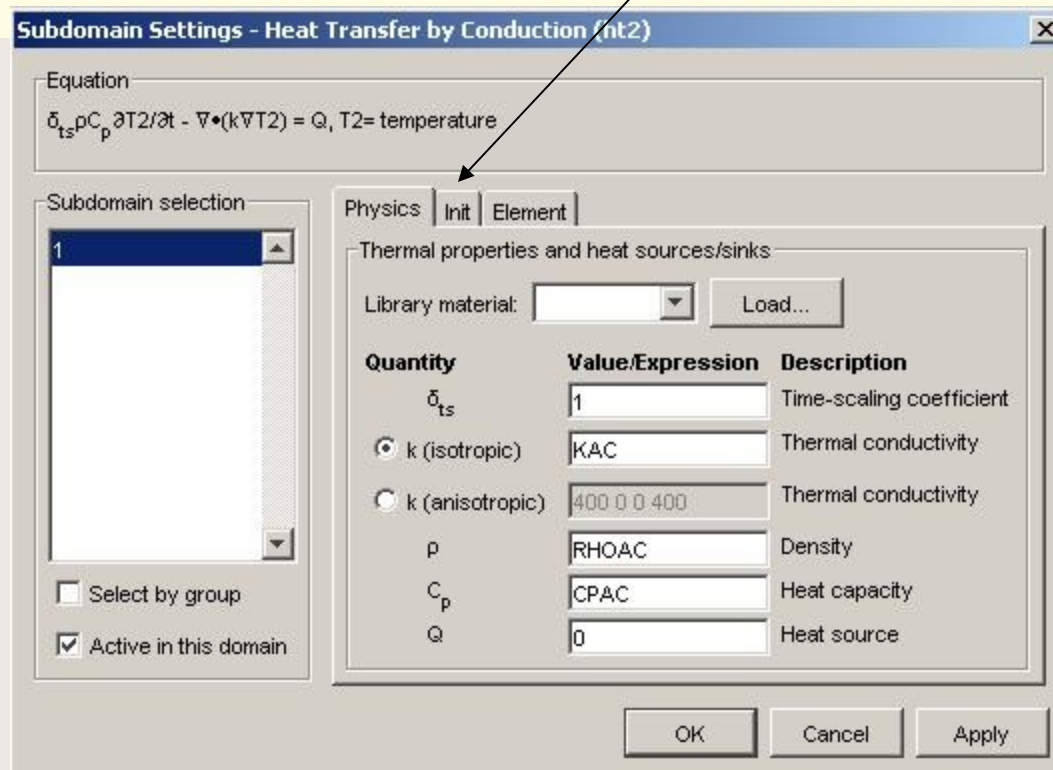
Quantity	Value/Expression	Description
q_0	-(T-Troom)	Inward heat flux
h	HCV	Heat transfer coefficient
T_{inf}	0	External temperature
Const	0	Problem-dependent constant
T_{amb}	0	Ambient temperature
T_0	0	Temperature

OK Cancel Apply

The sensitivity problems

- the resolution with femlab -

- In « physics » menu ... select « subdomain settings » to define the domain properties. And verify that the initial value is equal to zero.



The sensitivity problems

- the resolution with femlab -

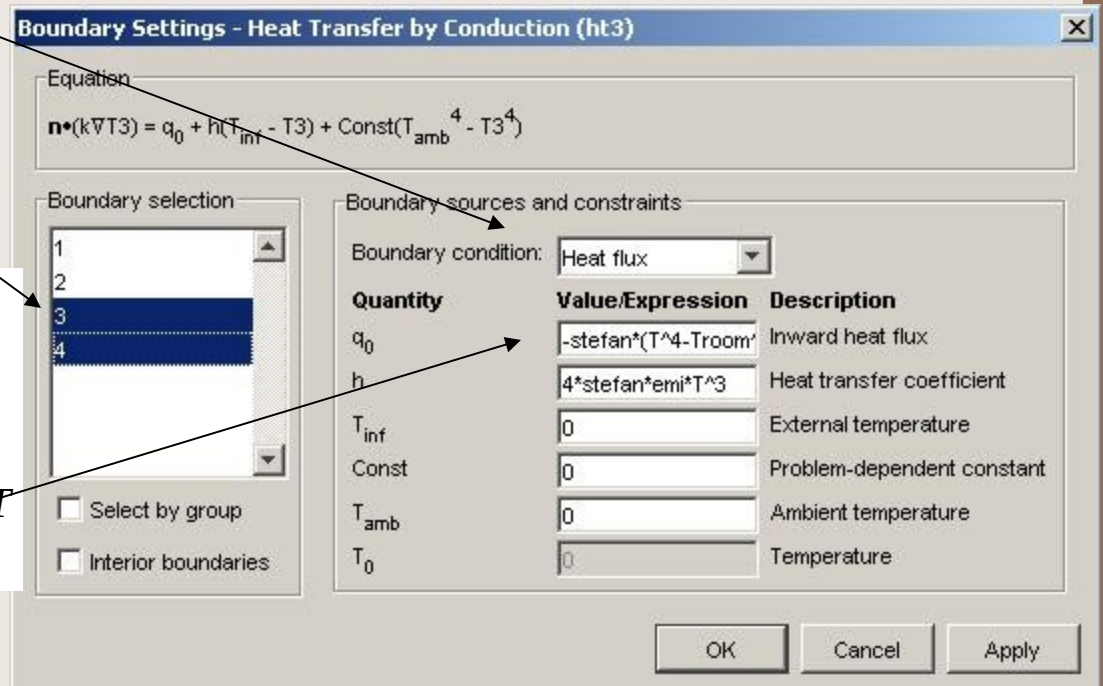
- Go to « multiphysics » and select the 3rd mode (ht3)
- In « physics » menu ... « boundary settings » to define the boundary conditions of the sensitivity problem for the emissivity coefficient.

$$\begin{aligned}\rho C_p \frac{\partial J_{emi}}{\partial t} - \nabla \cdot (k \cdot \nabla J_{emi}) &= 0 \in \Omega \times T \\ -k \frac{\partial J_{emi}}{\partial \vec{n}} &= 0 \text{ on } \Gamma_2 \times T \\ -k \frac{\partial J_{emi}}{\partial \vec{n}} &= HCV * J_{emi} \text{ on } \Gamma_2 \times T \\ -k \frac{\partial J_{emi}}{\partial \vec{n}} &= 4 \varepsilon \sigma T^3 J_{emi} + \sigma (T^4 - T_{room}^4) \text{ on } \Gamma_{3,4} \times T \\ J_{emi}(r, z; t=0 s) &= 0 \in \Omega\end{aligned}$$

The sensitivity problems

- the resolution with femlab -

- Enter the boundary conditions for the modes ht3 :
- Boundary selection : 1 => axial symmetry, 2 => heat flux with h:'HCV',
 3,4 => heat flux



$$\rho C_p \frac{\partial J_{emi}}{\partial t} - \nabla \cdot (k \cdot \nabla J_{emi}) = 0 \in \Omega \times T$$

$$-k \frac{\partial J_{emi}}{\partial \mathbf{n}} = 0 \text{ on } \Gamma_2 \times T$$

$$-k \frac{\partial J_{emi}}{\partial \mathbf{n}} = \text{HCV} * J_{emi} \text{ on } \Gamma_2 \times T$$

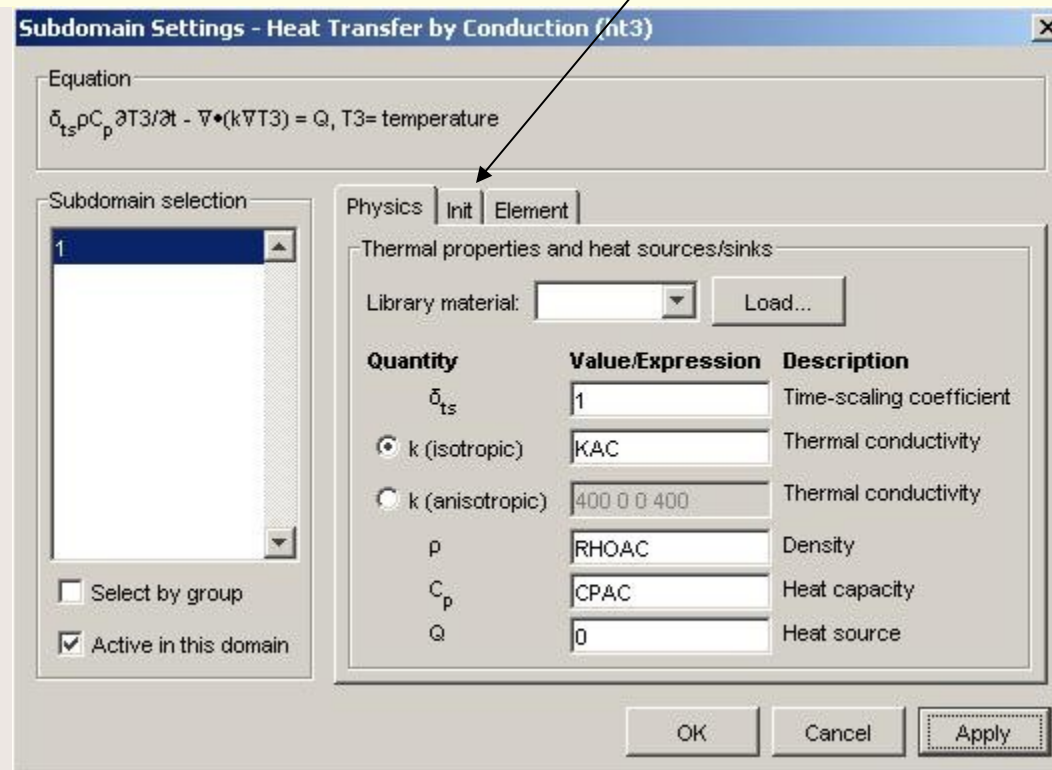
$$-k \frac{\partial J_{emi}}{\partial \mathbf{n}} = 4 \varepsilon \sigma T^3 J_{emi} + \sigma (T^4 - T_{\text{room}}^4) \text{ on } \Gamma_{3,4} \times T$$

$$J_{emi}(r, z; t=0 \text{ s}) = 0 \in \Omega$$

The sensitivity problems

- the resolution with femlab -

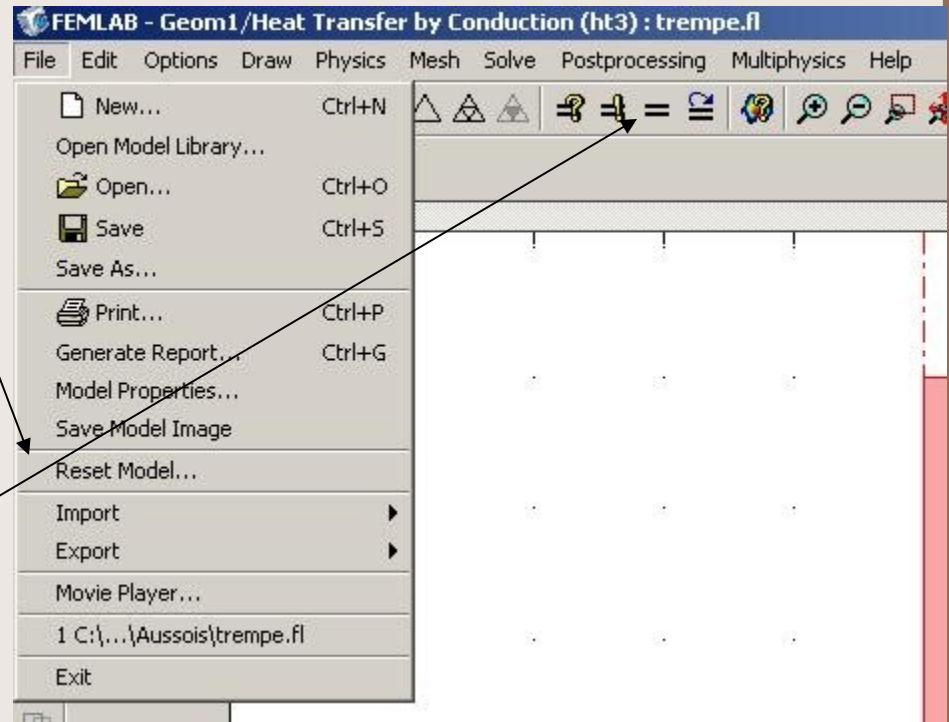
- In « physics » menu ... select « subdomain settings » to define the domain properties. And verify that the initial value is equal to zero.



The sensitivity problems

- the resolution with femlab -

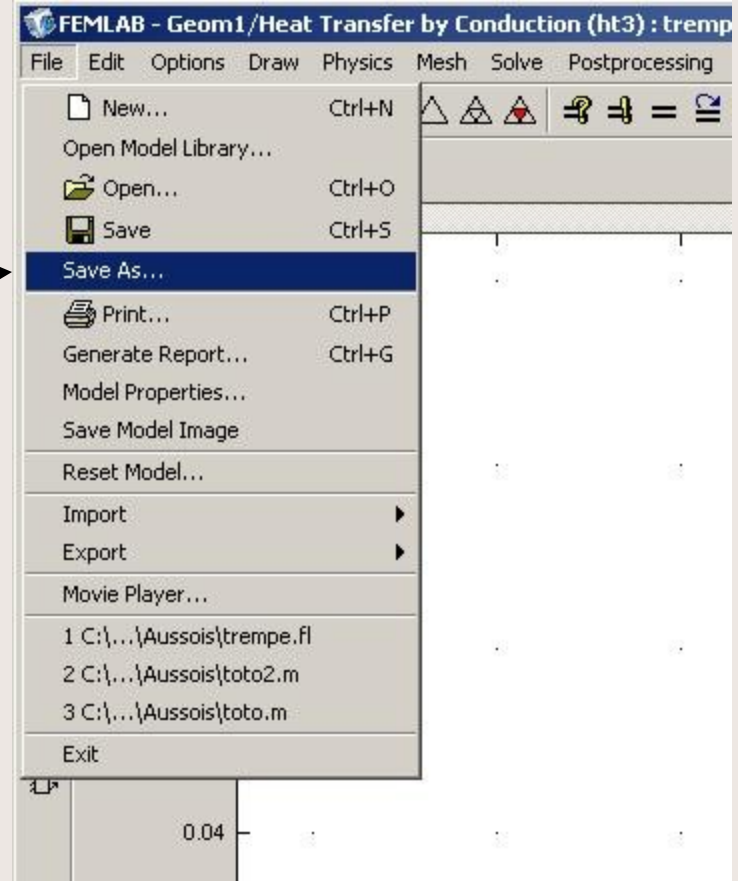
- In the bar menu, choose « file » then « reset M-file » before solving simultaneously the direct problem and the two sensitivity problems.
- Solve these three problems by clicking on « solve (=) »



The sensitivity problems

- the resolution with femlab -

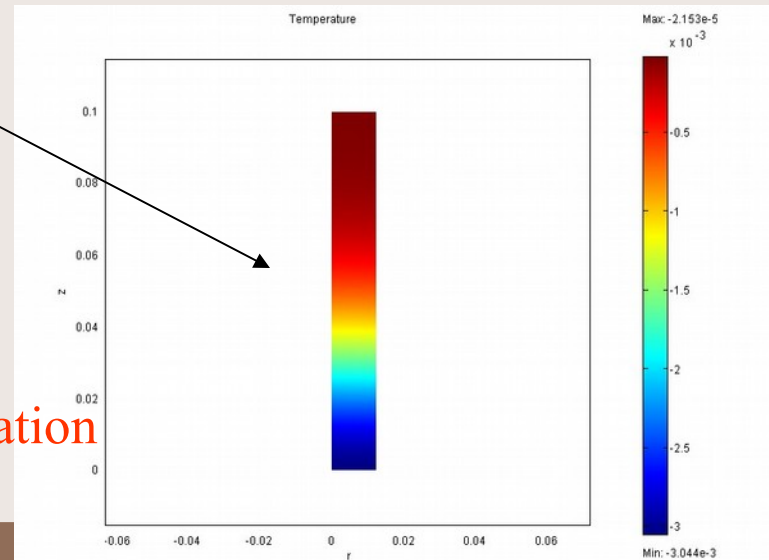
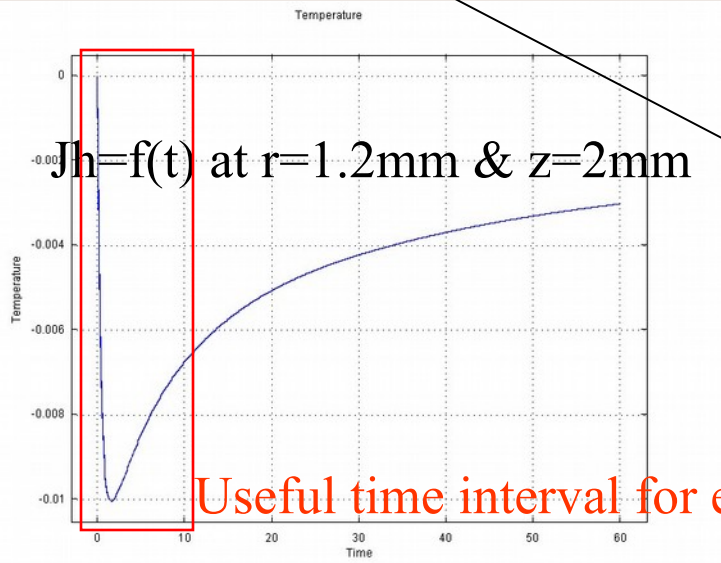
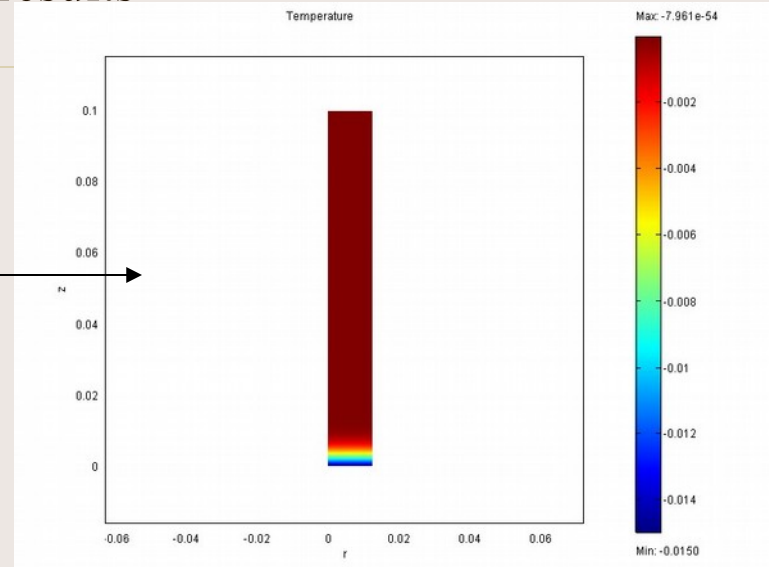
- Save the data in a M-file & .flb file (keep these two file is the same folder).
- Go to « file » in the menu, choose « Save As ». Save it as a matlab file (.m exetension)
- Name it ‘quenching2p’
- The program generates a quenching2p.m file.



The sensitivity problems

- sensitivity results -

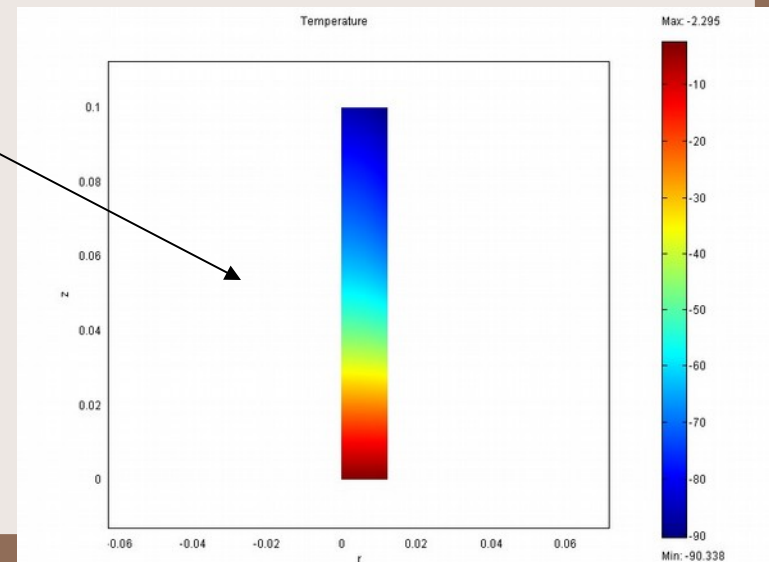
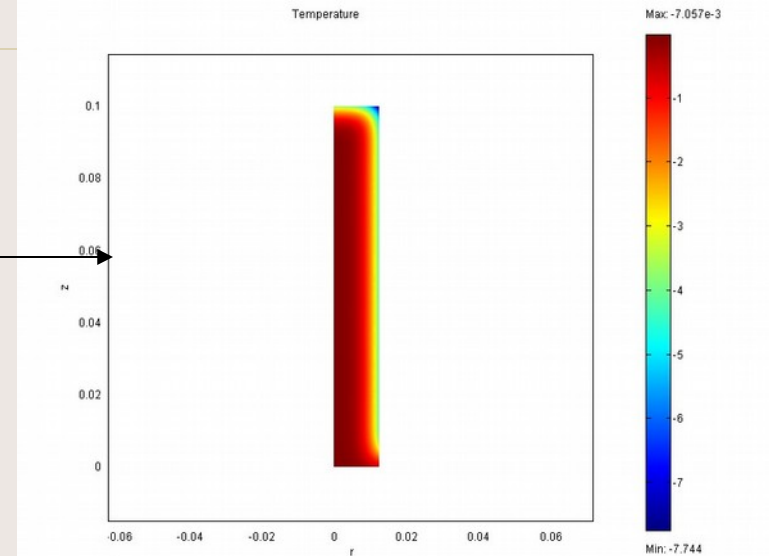
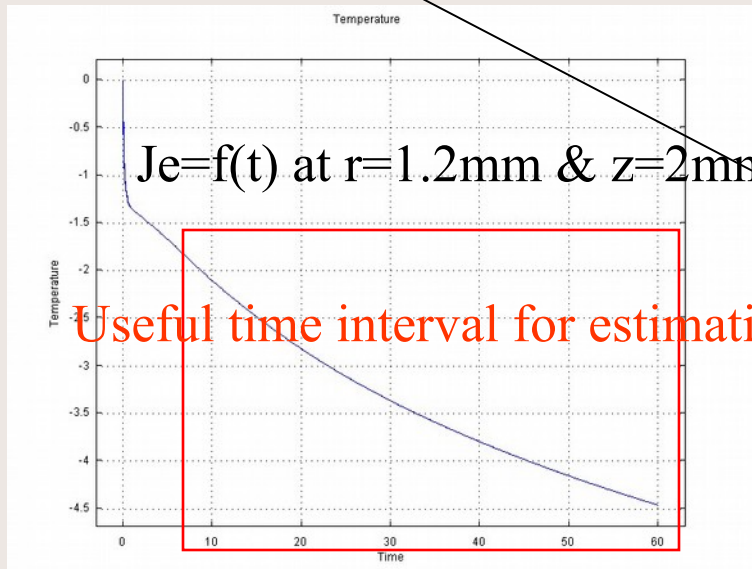
- Isovalue for HCV sensitivity coefficients at time $t=1s$
- Isovalue for HCV sensitivity coefficients at time $t=60s$



The sensitivity problems

- sensitivity results -

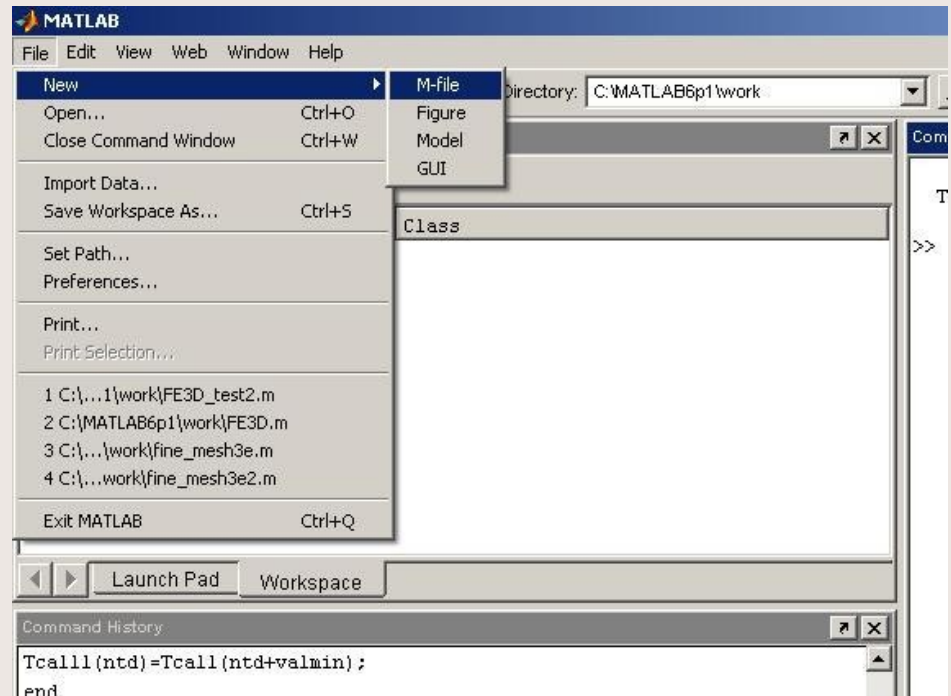
- Isovalue for emi sensitivity coefficients at time $t=1s$
- Isovalue for emi sensitivity coefficients at time $t=60s$



The inverse boundary problem resolution

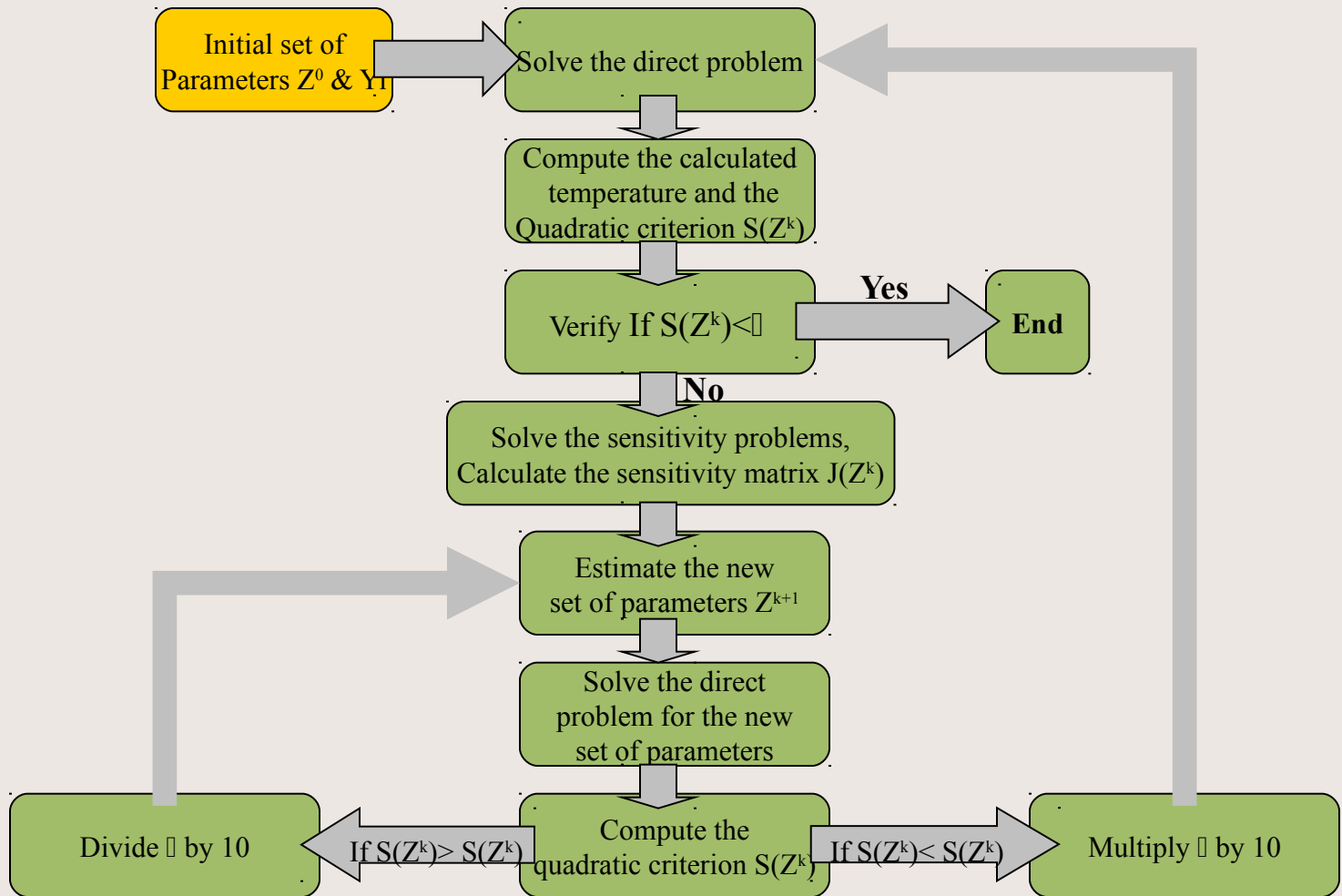
- building of the inverse problem program -

- Go to Matlab environments and open a new M-file in which we are going to program the inverse problem program.
- Name it « main_quenching2p.m »
- To do that : go to « file » menu then « save as »
- Give a title at the begin of the file (the sentence is preceeded of % symbol for comments).



The inverse boundary problem resolution

- building of the inverse problem program -



The inverse boundary problem resolution

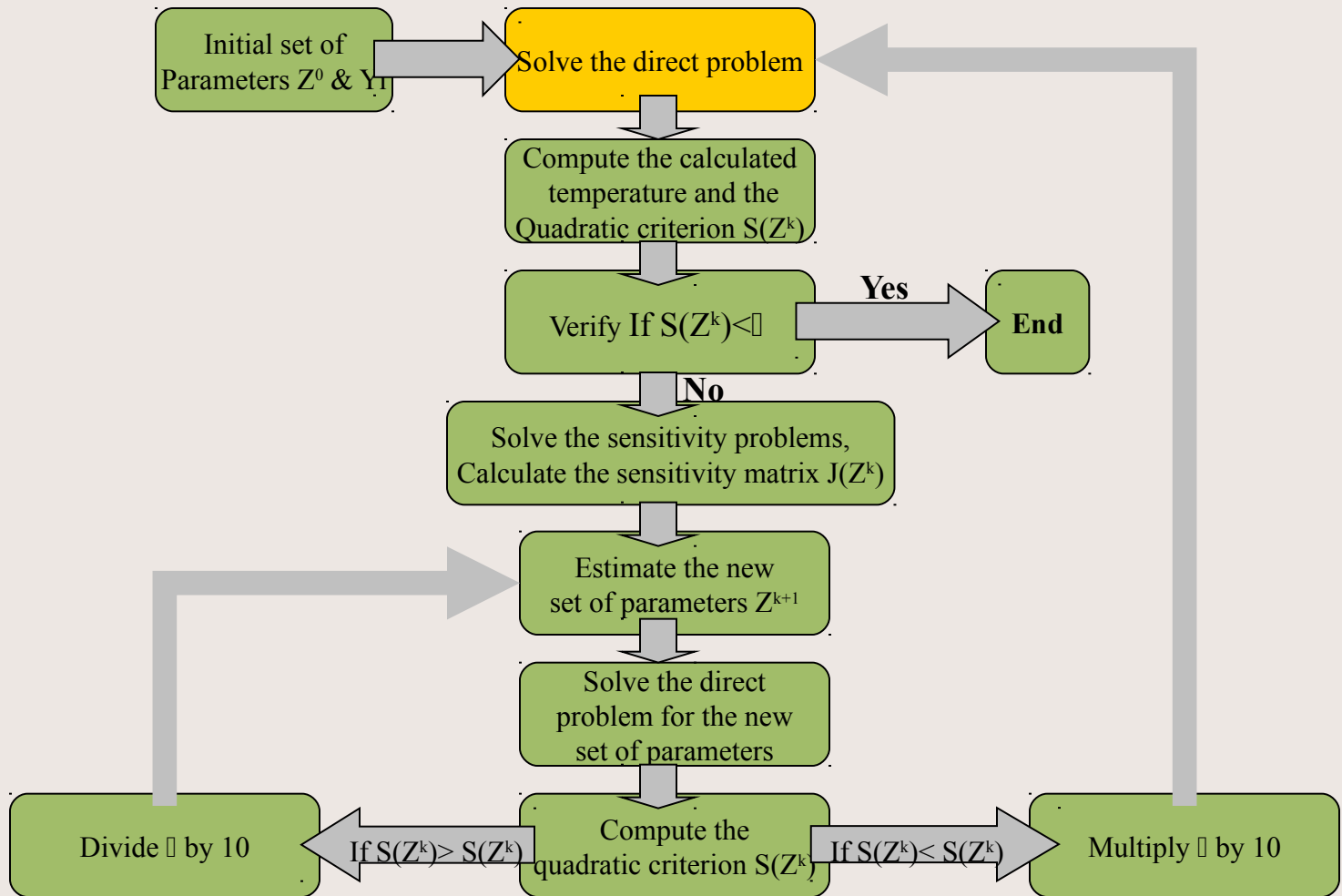
- building of the inverse problem program -

- First let solve the direct problem with the true parameters to obtain the theoretical temperatures Y_i .
- Write the followings lines.

```
C:\Documents and Settings\sébastien\Bureau\Aussois\main_quenching2p.m
File Edit View Text Debug Breakpoints Web Window Help
[Icons] Stack: Base
1 %Identification of the emissivity and the convective heat transfer coefficient
2 %in the quenching process. Used optimization method : Levenberg-Marquardt
3
4 clear
5
6 global coefH coefE
7 global ITEMPS
8
9 ITEMPS=[0:0.01:1,1.2:0.2:60];%time interval
10
11 %Resolution of the direct problem to obtain the exact temperatures (Yi)
12 coefH=15000;%initialization of the convective heat transfer coefficient
13 coefE=0.6;%initialization of the emissivity coefficient
14 toto;%call of the direct problem generates with femlab
15
16 %Interpolation of the exact temperatures (Yi) at point ncl and save it
17 ncl=[0.012;0.002];%r=0m and z=2mm
18 [Yi]=postinterp(femquench,'T',ncl,'solnum',1:length(fem.sol.tlist));
19 save Tquench.txt Yi -ASCII
20 % Yi=load('Tquench.txt');
21
22 % for noised temperature uncomment these following lines :
23 for (it=1:length(Yi))
24     Yi(it)=Yi(it)+0.01*Yi(it)*cos(3.14*rand(1)); % the theoretical temperat
25 end % cos(3.14*rand(1)) gives
```

The inverse boundary problem resolution

- building of the inverse problem program -



The inverse boundary problem resolution

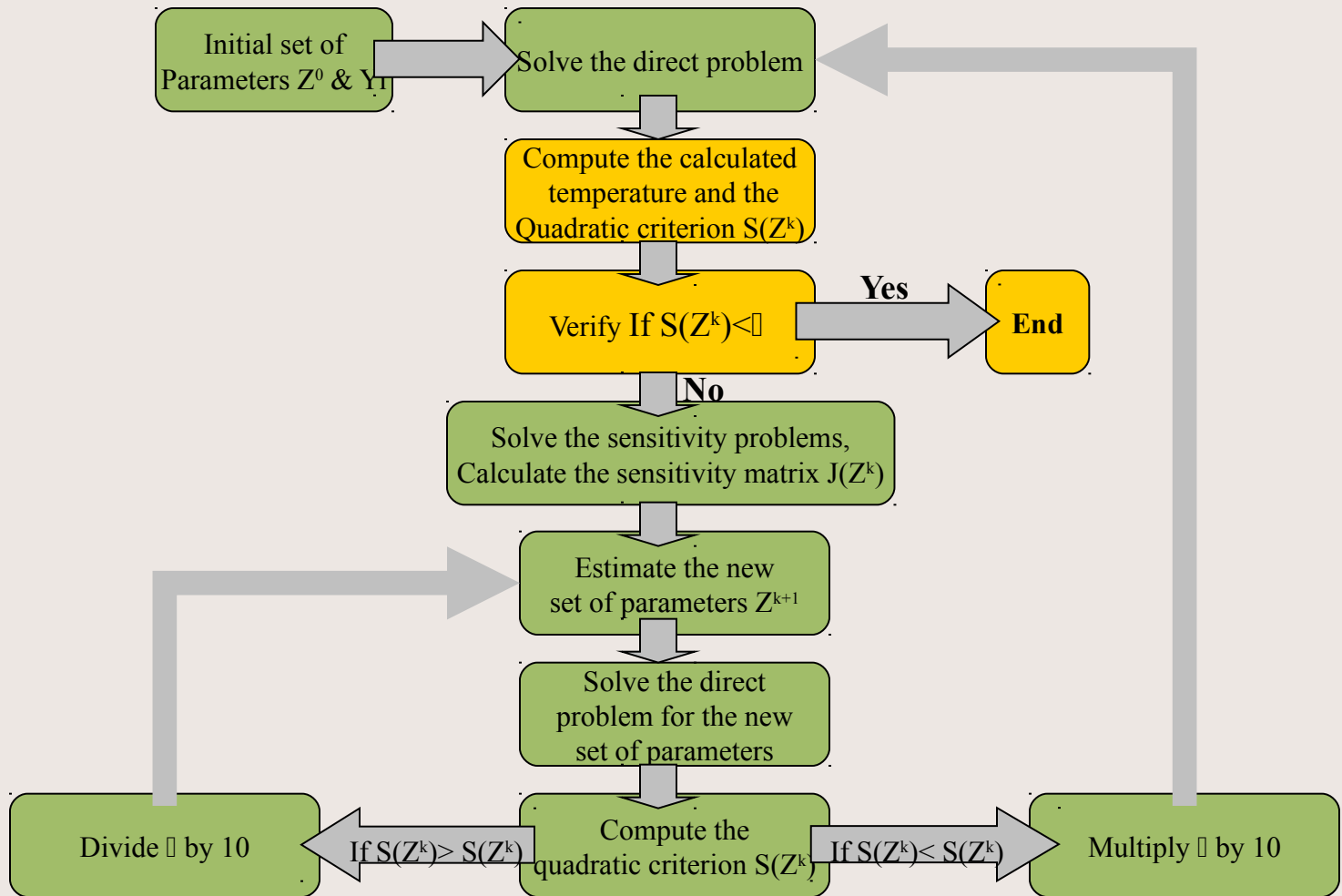
- building of the inverse problem program -

- Initialize the parameters of the levenberg-Marquardt method : unknown parameters, damping parameter, weight matrix, first criterion, ... (for estimation just the first measurements are useful! because the values of the sensitivity coefficient ...)

```
27 % ***** beginning of the inverse problem resolution procedure *****
28 % by the levenberg-Marquardt method :  $P_{k+1} = P_k + \text{inv}([J_h' * WJ * J_h + \lambda(k) * \text{OMEG}(k)]) * (J_h$ 
29 % where  $P_k$  is the unknown parameter at iteration  $k$ ,  $J_h$  the sensitivity coefficients,  $\lambda$ 
30 %  $WJ$  a matrix,  $\text{OMEG}$  an other matrix,  $Y_i$  the measured data and  $T_{\text{cal}}$  the calculated data
31
32 %Initializing of the unknown parameter coefH
33 - coefH=0;
34 - coefHprec=coefH;%necessary for the levenberg-marquardt method, value of the unknown para
35 - coefE=0;
36 - coefEprec=coefE;
37
38 %Initialization of the lambda coefficient used in the levenberg-marquardt method
39 - lambda=0.001;
40
41 %Creation of the matrix WJ (equal to the identity matrix when measurements are not noise
42 -  $WJ = \text{eye}(\text{length}(Y_i(1:21)))$ ;
43
44 %Creation of the matrix OMEG equal to a two dimension identity matrix for two parameters
45 -  $\text{OMEG} = \text{eye}(2)$ ;
46
47 - SOM=Yi-20;
48 -  $\text{somme} = (\text{norm}(\text{SOM}) * \text{norm}(\text{SOM}))$ ;%initialization of the quadratic criterion (exact data minus
49 - niter=1;%initialization of the iteration number
50
```


The inverse boundary problem resolution

- building of the inverse problem program -



The inverse boundary problem resolution

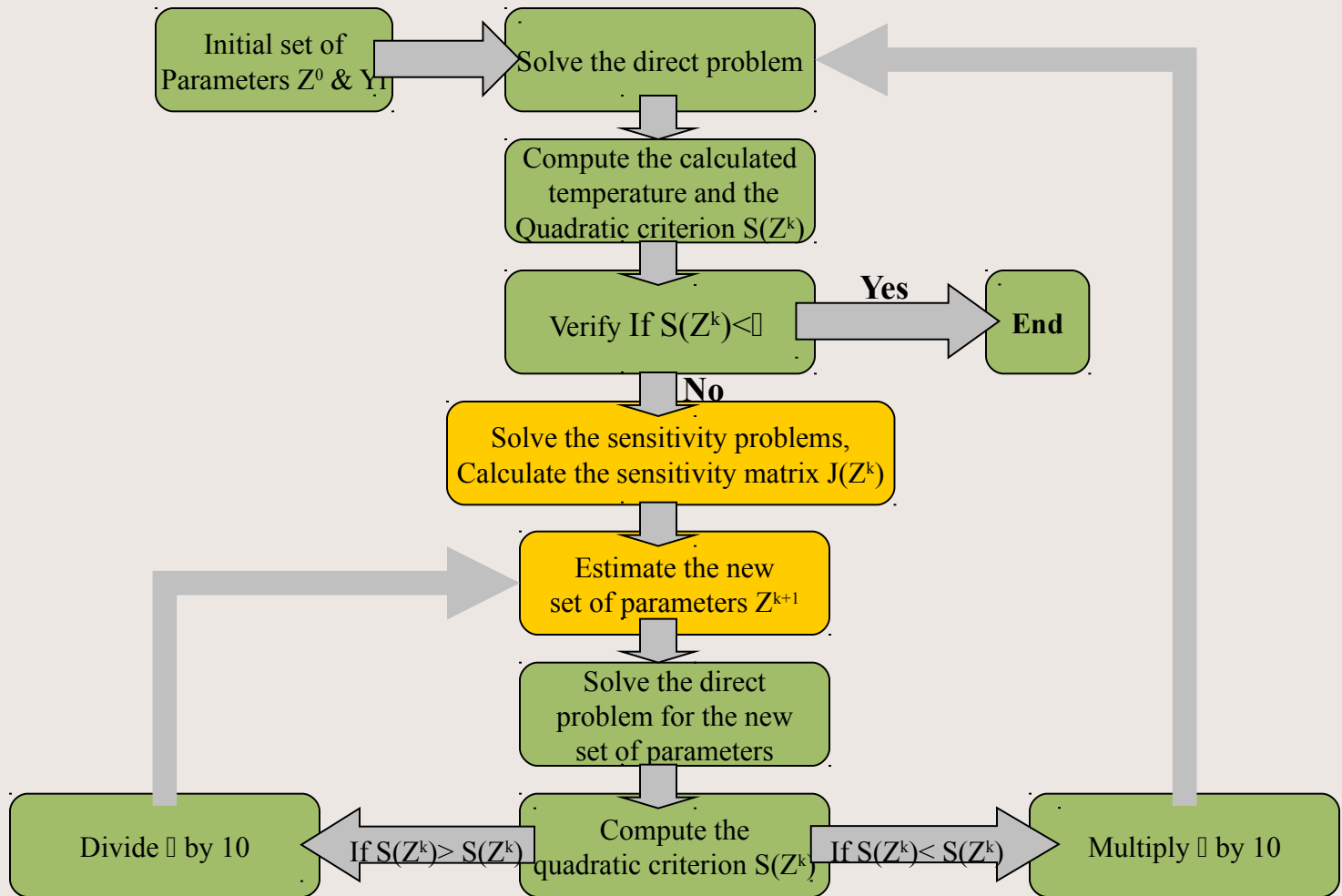
- building of the inverse problem program -

- Open the beginning of the optimization loop
- Solve the direct problem for the set of initial unknown parameters
- Interpolate the calculated temperatures
- Compute the quadratic criterion (called « somme » here)

```
51     %beginning of the optimization loop
52 -   while ((somme>1e-3)&(niter<25))
53
54     %resolution of the direct problem (and the sensitivity problem) at iteration k=0
55 -   toto;
56
57     %Mesure de la température calculée à la première itération k=0
58 -   [Tcal]=postinterp(femquench,'T',ncl,'solnum',1:length(fem.sol.tlist));
59
60     %Calculation of quadratic criterion  $S(k)=(Tcal(t)-Yi(t))*(Tcal(t)-Yi(t))$ 
61 -   SOM=Yi-Tcal;
62 -   somme=norm(SOM)*norm(SOM);
63
64 -   sprintf('critere %18.8f à l iteration %3f',somme,niter)
65
```

The inverse boundary problem resolution

- building of the inverse problem program -



The inverse boundary problem resolution

- building of the inverse problem program -

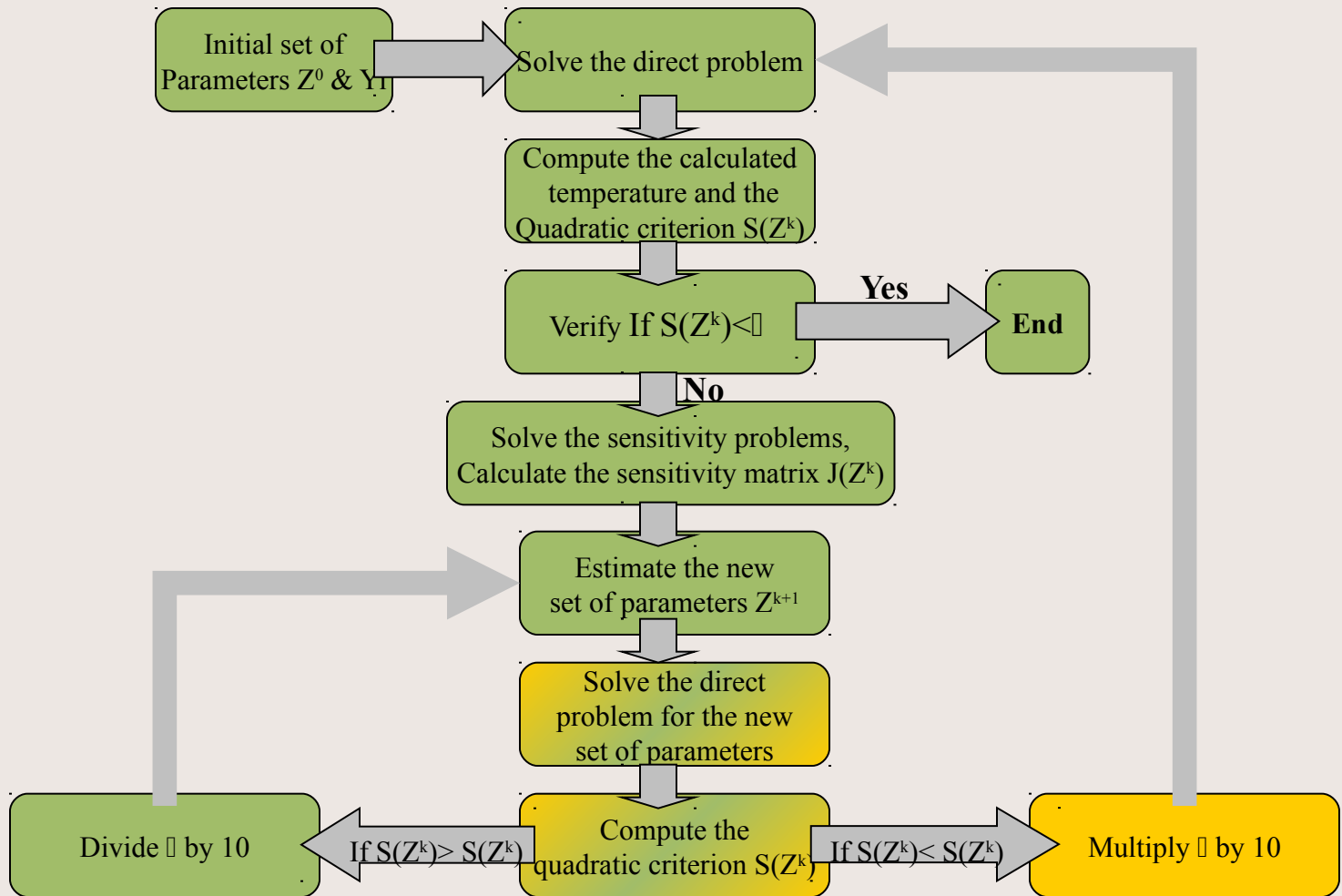
```
66 - if (niter==1)
67 -     %case niter=1 (for the first iteration)
68 -     somprec=somme;
69 -     EVOLH(niter)=coefH;
70 -     EVOLE(niter)=coefE;
71 -     EVOLSOM(niter)=somme;
72 -
73 -     %interpolation of the sensitivities coefficients
74 -     [Jh]=postinterp(femquenched,'T2',ncl,'solnum',1:length(fem.sol.tlist));
75 -     [Je]=postinterp(femquenched,'T3',ncl,'solnum',1:length(fem.sol.tlist));
76 -     Jhe=[Jh(1:21) Je(1:21)];
77 -
78 -     %Estimation of the parameter coefH with  $P(k) = \frac{1}{\lambda} - \frac{1}{\lambda + \text{coefH} * \text{OMEG}}$ 
79 -     dcoef = inv((Jhe'*WJ*Jhe+lambda*OMEG))*(Jhe'*WJ*SOM(1:21));
80 -
81 -     coefH = coefHprec + dcoef(1);
82 -     coefE2 = coefEprec + dcoef(2);
83 -
84 -     if ((coefE2>0)&(coefE2<1)) coefE=coefE2;
85 -     end
86 -
87 -     sprintf('New estimated values for coefH %8.1f and coefE %6.3f',coefH,coefE)
88 -
89 -     niter=niter+1;
90 -
```

We use the 21 first measurements
for the estimation

- Case of the first iteration $k=1$: Computation of the new estimated, and read of the sensitivity coefficients values ('Jh' and 'Je')

The inverse boundary problem resolution

- building of the inverse problem program -



The inverse boundary problem resolution

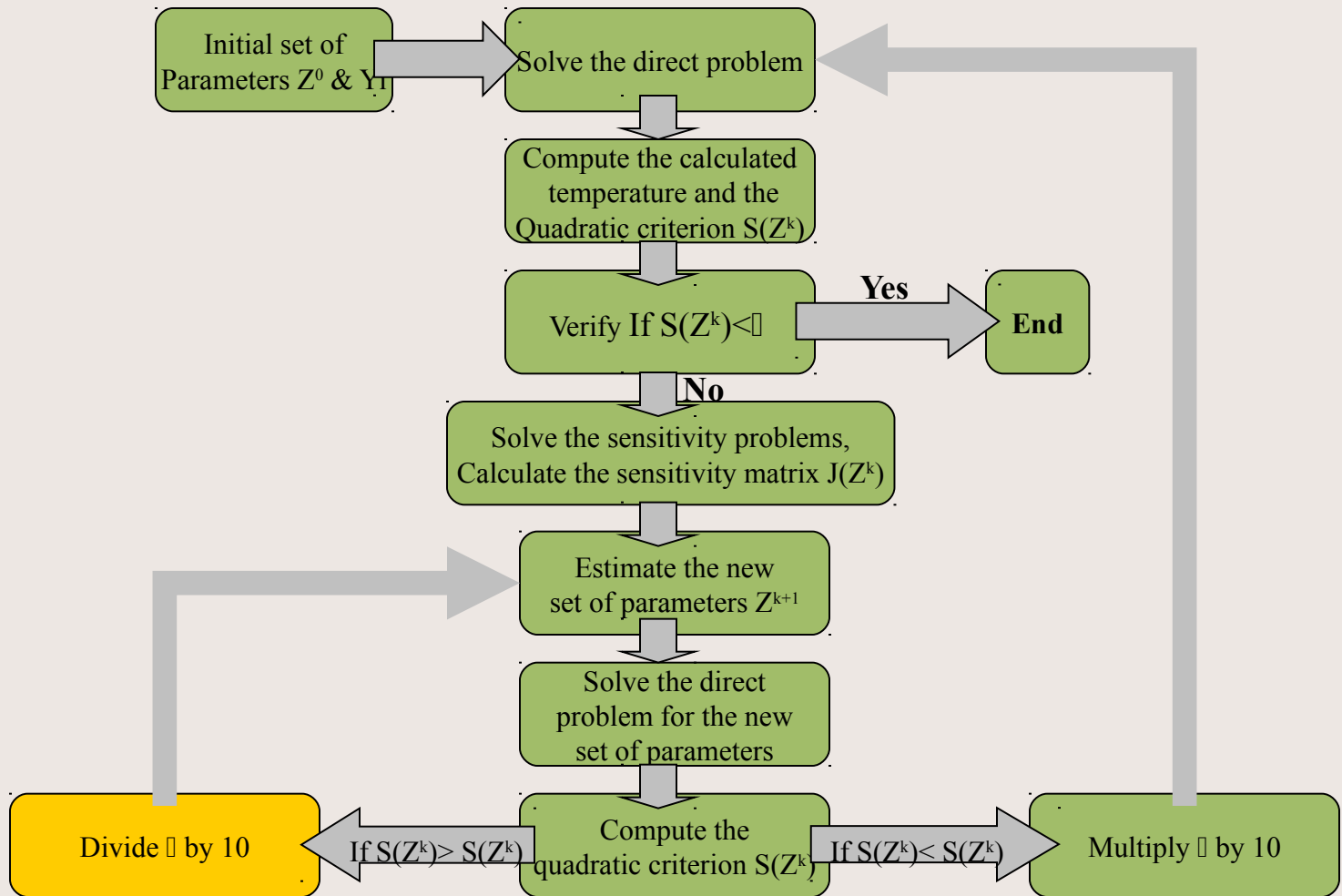
- building of the inverse problem program -

```
91 - else
92 -     if (somme<somprec)
93 -         %case somme<somprec (when the new calculated quadratic criterion is smaller t
94 -         lambda = 0.1*lambda;%divide by 10 the damping parameter
95 -         somprec=somme;
96 -         EVOLH(niter)=coefH;
97 -         EVOLE(niter)=coefE;
98 -         EVOLSOM(niter)=somme;
99 -
100 -        %interpolation of the sensitivities coefficients
101 -        [Jh]=postinterp(femquench,'T2',ncl,'solnum',1:length(fem.sol.tlist));
102 -        [Je]=postinterp(femquench,'T3',ncl,'solnum',1:length(fem.sol.tlist));
103 -        Jhe=[Jh(1:21) Je(1:21)];
104 -
105 -        %Estimation of the parameter coefH with Pk+1 = Pk + inv([Jhe'*WJ*Jhe+lambda(k
106 -        dcoef = inv((Jhe'*WJ*Jhe+lambda*OMEG))*(Jhe'*WJ*SOM(1:21)));
107 -        coefHprec = coefH;
108 -        coefEprec = coefE;
109 -        coefH = coefHprec + dcoef(1);
110 -        coefE2 = coefEprec + dcoef(2);
111 -
112 -        if ((coefE2>0)&(coefE2<1)) coefE=coefE2;
113 -        end
114 -
115 -        sprintf('New estimated values for coefH %8.1f and coefE %6.3f',coefH,coefE)
116 -
117 -        niter=niter+1;
118 -
```

- Case $S(Z^{k+1}) < S(Z^k)$

The inverse boundary problem resolution

- building of the inverse problem program -



The inverse boundary problem resolution

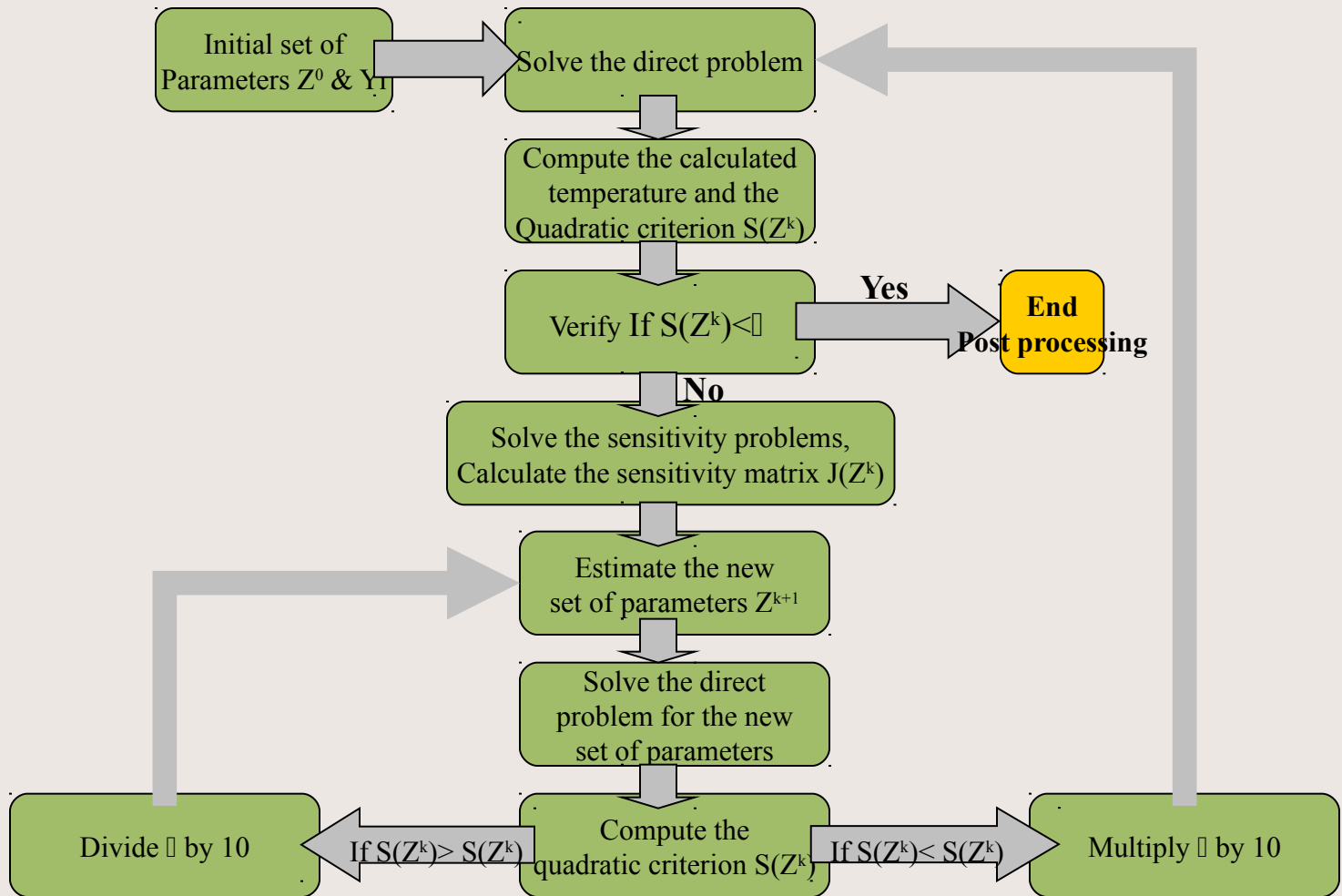
- building of the inverse problem program -

```
119 - else
120 -     %case somme>somprec (when the new calculated quadratic criterion is bigger than
121 -     lambda=10*lambda;%multiply by 10 the damping parameter
122 -
123 -     %Estimation of the parameter coefH with Pk+1 = Pk + inv([Jhe'*WJ*Jhe+lambda*I
124 -     dcoef = inv((Jhe'*WJ*Jhe+lambda*OMEG))*(Jhe'*WJ*SOM(1:21));
125 -     coefH = coefHprec + dcoef(1);
126 -     coefE2 = coefEprec + dcoef(2);
127 -
128 -     if ((coefE2>0)&(coefE2<1)) coefE=coefE2;
129 -     end
130 -
131 -     sprintf('New estimated values for coefH %8.1f and coefE %6.3f',coefH,coefE)
132 -
133 -     end
134 -
135 - end
136 -
137 - end%end of the optimization loop
138 -
```

- Case $S(Z^{k+1}) > S(Z^k)$
- And close all the loop

The inverse boundary problem resolution

- building of the inverse problem program -



The inverse boundary problem resolution

- building of the inverse problem program -

```
139 %post-processing step
140
141 - it=1:niter-1;
142 - EVOLHEXA(1:niter-1)=15000;
143 - EVOLEEXA(1:niter-1)=0.6;
144
145 - figure(1)
146 - plot(ITEMPS,SOM)
147 - grid
148 - xlabel('time (s)')
149 - ylabel('Temperature difference')
150 - legend('Evolution of difference between theoritical and calculated temperature',1)
151
152 - figure(3)
153 - plot(ITEMPS,Tcal,'s',ITEMPS,Yi,'-.')
154 - grid
155 - ylabel('Temperature (°C)')
156 - xlabel('time (s)')
157 - legend('Calculated temperature','Theoritical temperature',1)
158
159 - figure(4)
160 - semilogy(it,EVOLSOM)
161 - grid
162 - ylabel('Quadratic criterion')
163 - xlabel('Iteration number')
164 - legend('Evolution of the quadratic criterion in function of iteration number',3)
165
```

- To visualize the results of the numerical identification (1)

The inverse boundary problem resolution

- building of the inverse problem program -

```
166 - figure(5)
167 - plot(it,EVOLHEXA,it,EVOLH,'-.s')
168 - grid
169 - ylabel('H parameter')
170 - xlabel('Iteration number')
171 - title('Evolution of the H parameter in function of iteration number')
172 - Legend('Theoretical Value','Calculated Value',4)
173
174 - figure(6)
175 - plot(it,EVOLEEXA,it,EVOLE,'-.s')
176 - grid
177 - ylabel('Emissivity parameter')
178 - xlabel('Iteration number')
179 - title('Evolution of the emissivity parameter in function of iteration number')
180 - Legend('Theoretical Value','Calculated Value',4)
```

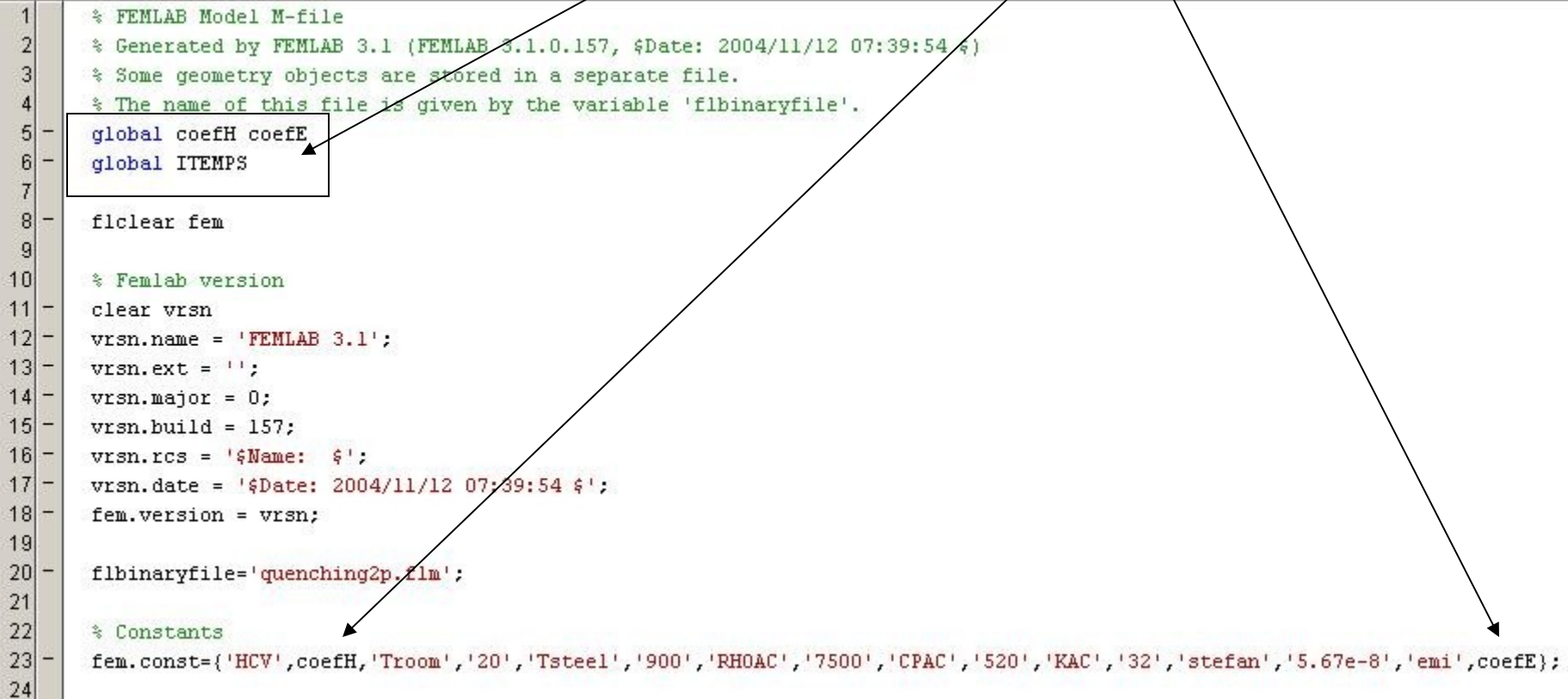
- To visualize the results of the numerical identification (2)

The inverse boundary problem resolution

- building of the inverse problem program -

- We have to modify the « quenching2p.m » file (the file obtained from femlab)
- Add these lines and keywords : coefH & coefE

```
1  % FEMLAB Model M-file
2  % Generated by FEMLAB 3.1 (FEMLAB 3.1.0.157, $Date: 2004/11/12 07:39:54 $)
3  % Some geometry objects are stored in a separate file.
4  % The name of this file is given by the variable 'flbinaryfile'.
5  global coefH coefE
6  global ITEMPS
7
8  flclear fem
9
10 % Femlab version
11 clear vrsn
12 vrsn.name = 'FEMLAB 3.1';
13 vrsn.ext = '';
14 vrsn.major = 0;
15 vrsn.build = 157;
16 vrsn.rcs = '$Name: $';
17 vrsn.date = '$Date: 2004/11/12 07:39:54 $';
18 fem.version = vrsn;
19
20 flbinaryfile='quenching2p.flm';
21
22 % Constants
23 fem.const={'HCV',coefH,'Troom','20','Tsteel','900','RHOAC','7500','CPAC','520','KAC','32','stefan','5.67e-8','emi',coefE};
24
```



The inverse boundary problem resolution

- building of the inverse problem program -

At the end of the file:

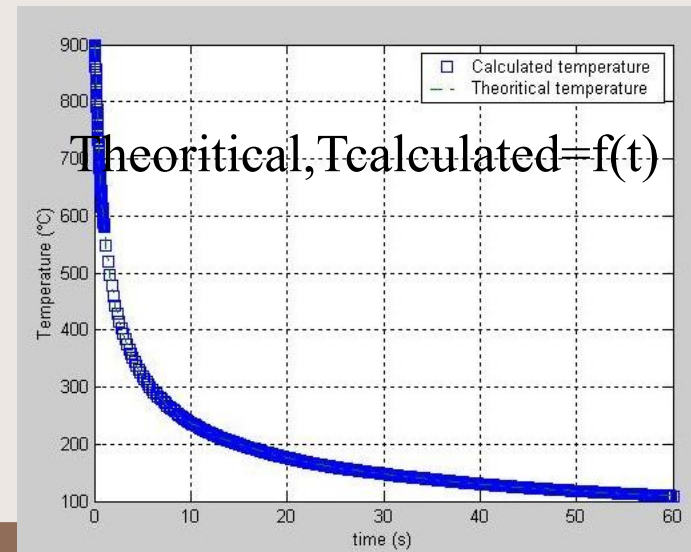
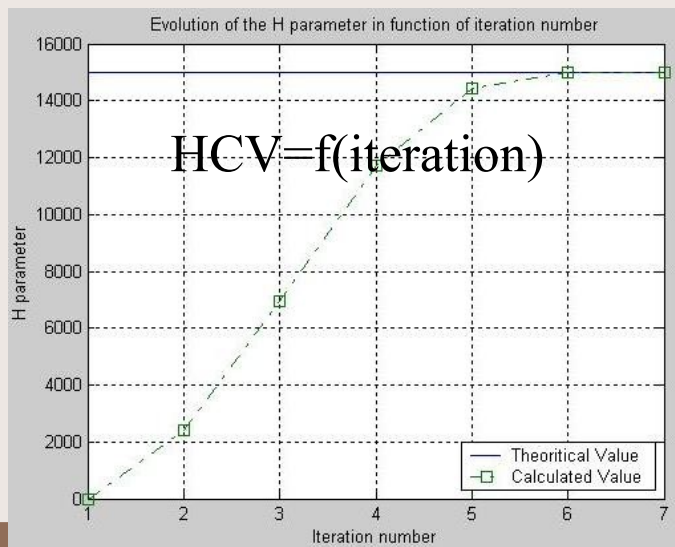
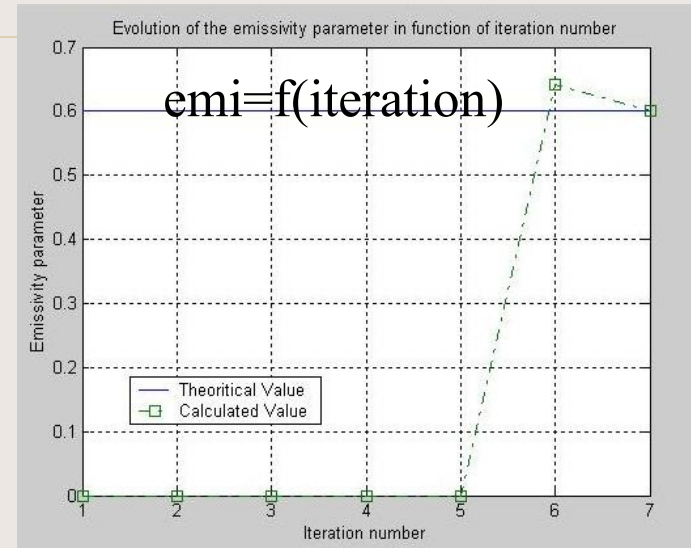
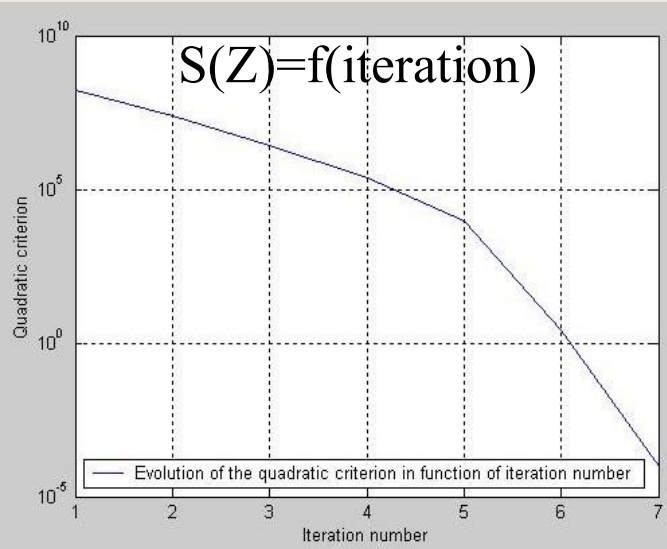
- In the solver, replace time vector by the vector named 'ITEMPS'
- Comment the postplot process

```
120 % Solve problem
121 fem.sol=fetime(fem, ...
122     'solcomp',{'T3','T','T2'}, ...
123     'outcomp',{'T3','T','T2'}, ...
124     'tlist',ITEMPS, ...
125     'tout','tlist', ...
126     'linsolver','cg');
127
128 % Save current fem structure for restart purposes
129 femquench=fem;
130
131 % Plot solution
132 postplot(fem, ...
133     'tridata',{'T','cont','internal'}, ...
134     'trimap','jet(1024)', ...
135     'solnum',601, ...
136     'title','Time=60    Surface: Temperature', ...
137     'refine',3, ...
138     'axis',[-0.088128699538072,0.100628699724337,-0.0050000000745058
139
```

You can run the program

The inverse boundary problem resolution

- results of the inverse problem program -



Thanks for your attention

And a special thanks to my colleagues for their councils :
Philippe Le MASSON and Tahar LOULOU

Thanks to Comsol support for their help today

Have a nice Eurotherm Winter School

