METTI IV – Thermal Measurements and Inverse Techniques, Roscoff, France, June 13-18, 2011

Tutorial 12 :

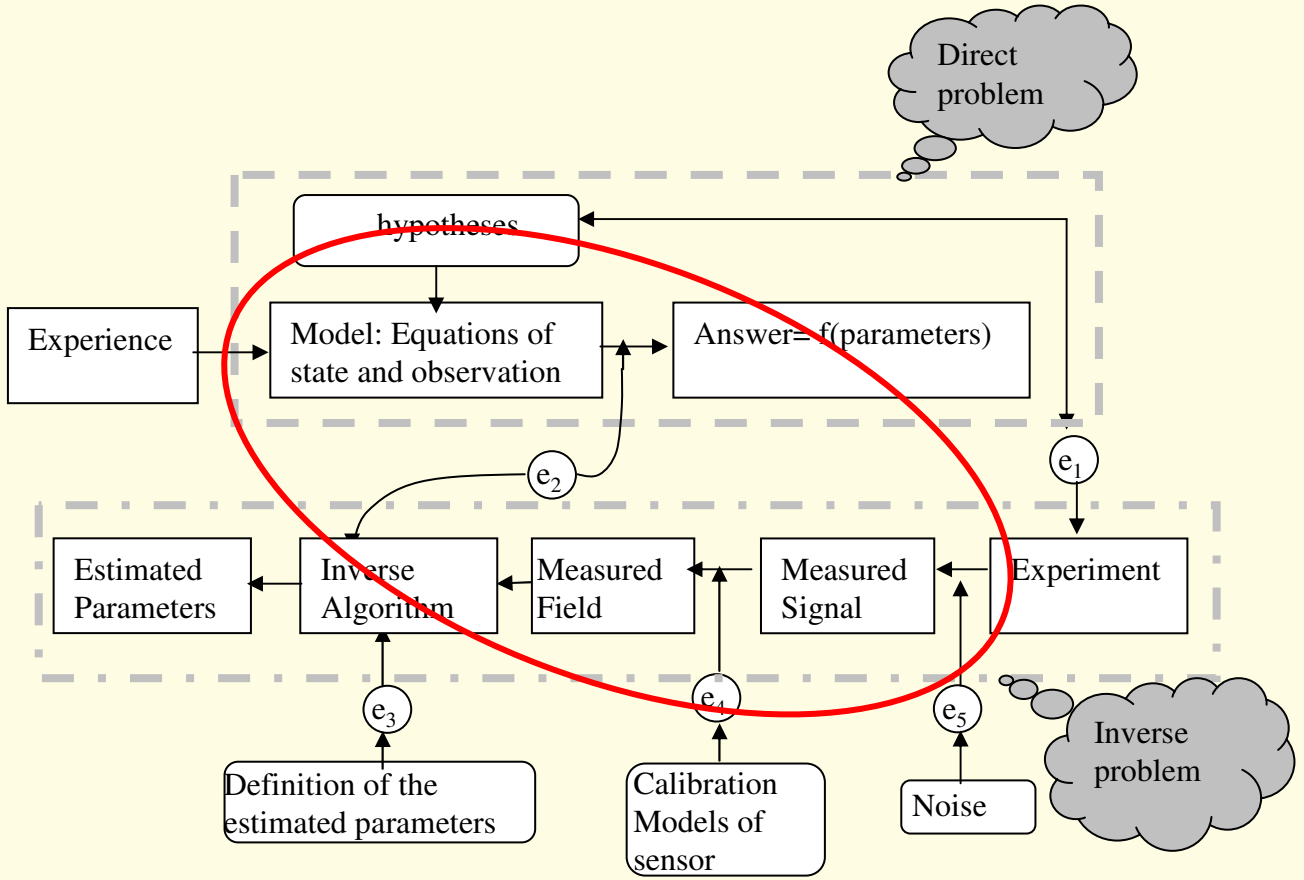
« Analysis of errors in measurements and inversion »

By Philippe LE MASSON° & Morgan DAL°

°Laboratoire d'Ingénierie des MATériaux de Bretagne. Université Européenne de Bretagne/ Université de Bretagne Sud; Centre de recherche de l'Université de Bretagne Sud, Rue saint Maudé, 56321 LORIENT Cédex.

philippe.le-masson@univ-ubs.fr; morgan.dal@univ-ubs.fr

The direct and inverse problems



Goals

- We use the Levenberg Marquardt method for the parameter estimation
- We use the software « Comsol Multiphysics » for the direct problem definitions
- We save this problem in a matlab file (« *.m »)
- We introduce the algorithm in a matlab file
- The resolution of the inverse problem is realised with Matlab.
- At last, we want to compare different measurement configurations for the estimation.

Outline

- Resolution of a direct welding problem with « Comsol Multiphysics »
- The Levenberg-Marquardt algorithm
- Resolution of the inverse problem with Matlab and Comsol Multiphysics
- Modelisation of the welding problem with thermocouples. Definition of the parameters.
- Resolution of the inverse problem with different measurement configurations
- Conclusions

The welding problem

- The governing equations :

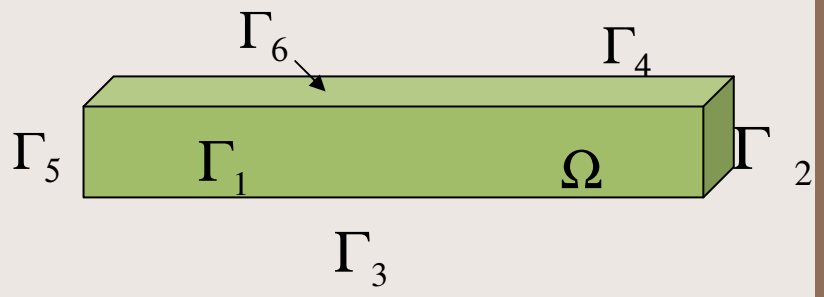
$$\rho C_p \frac{\partial T}{\partial t} - \nabla.(k \nabla T) = 0 \; in \; \Omega$$

$$-k \frac{\partial T}{\partial n} = 0 \; on \; \Gamma_1$$

$$-k \frac{\partial T}{\partial n} = h(T - T_{inf}) on \; all \; \Gamma_i \; for 2 \le i \le 5$$

$$-k \frac{\partial T}{\partial n} = h(T - T_{inf}) + q_0(x, y, t) on \; \Gamma_6$$

$$T(x, y, z, t = 0) = 20°C in \; \Omega$$



The welding problem

$q_0(x, y, t)$ is a Gaussian equivalent source:

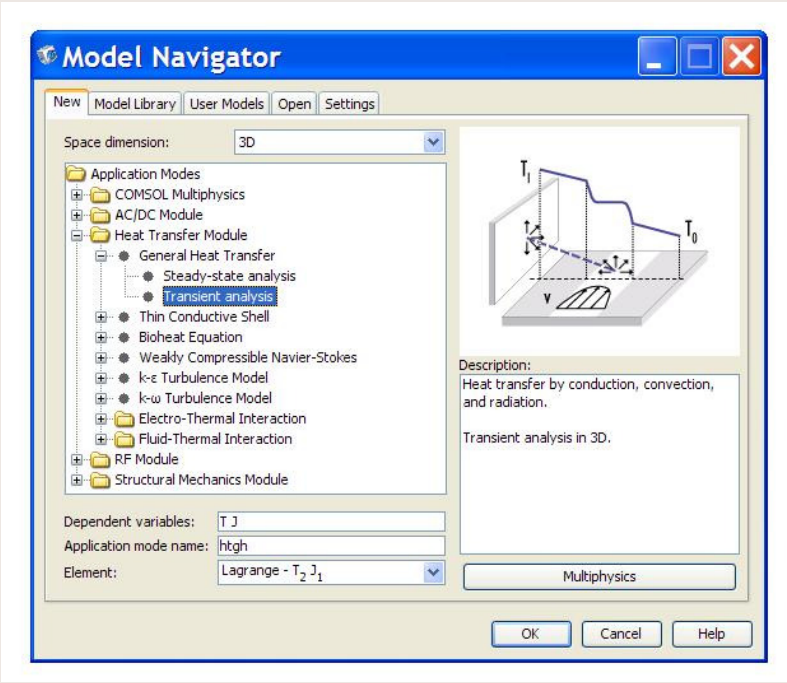
$$q_0(x,y,t) = \frac{Q}{2\pi r^2} exp\left(- \frac{\left(x^2 + (y - vt)^2\right)}{2r^2} \right)$$

The goal of the inverse method is to estimate Q.

The direct welding problem

- the resolution with comsol -

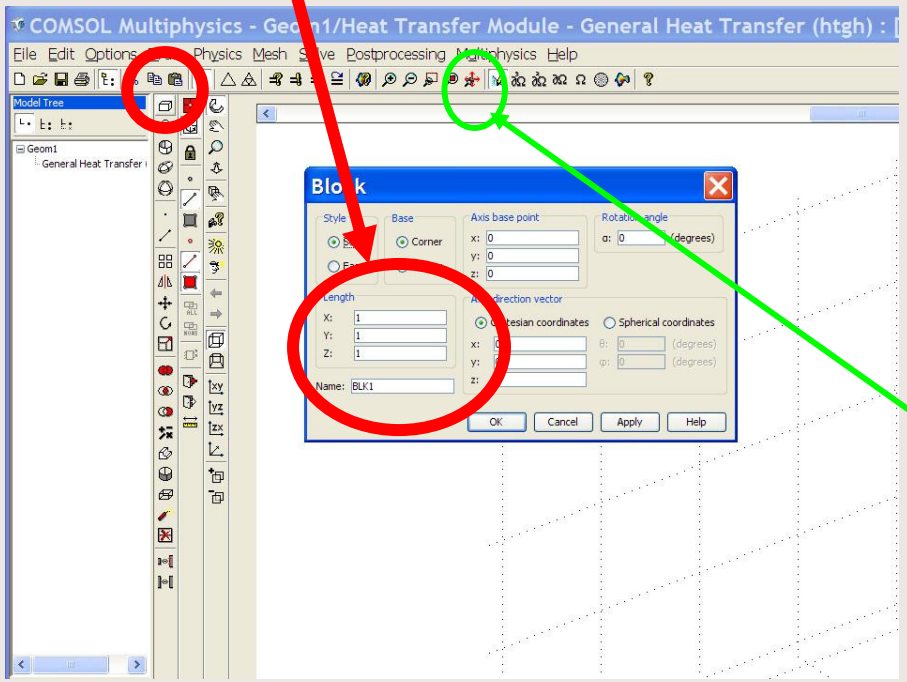
- Open comsol
- In the model navigator window, choose
- « Heat Transfer Module » … « 3D » …
- « General Heat Transfer» …
- « Transient analysis » …
- And click OK button



The direct welding problem

- the resolution with comsol -

- To draw a block of 30mm x 100mm x 10mm: Click on the « draw block icon »
- and insert the values in meter
- Draw a block in the draw window

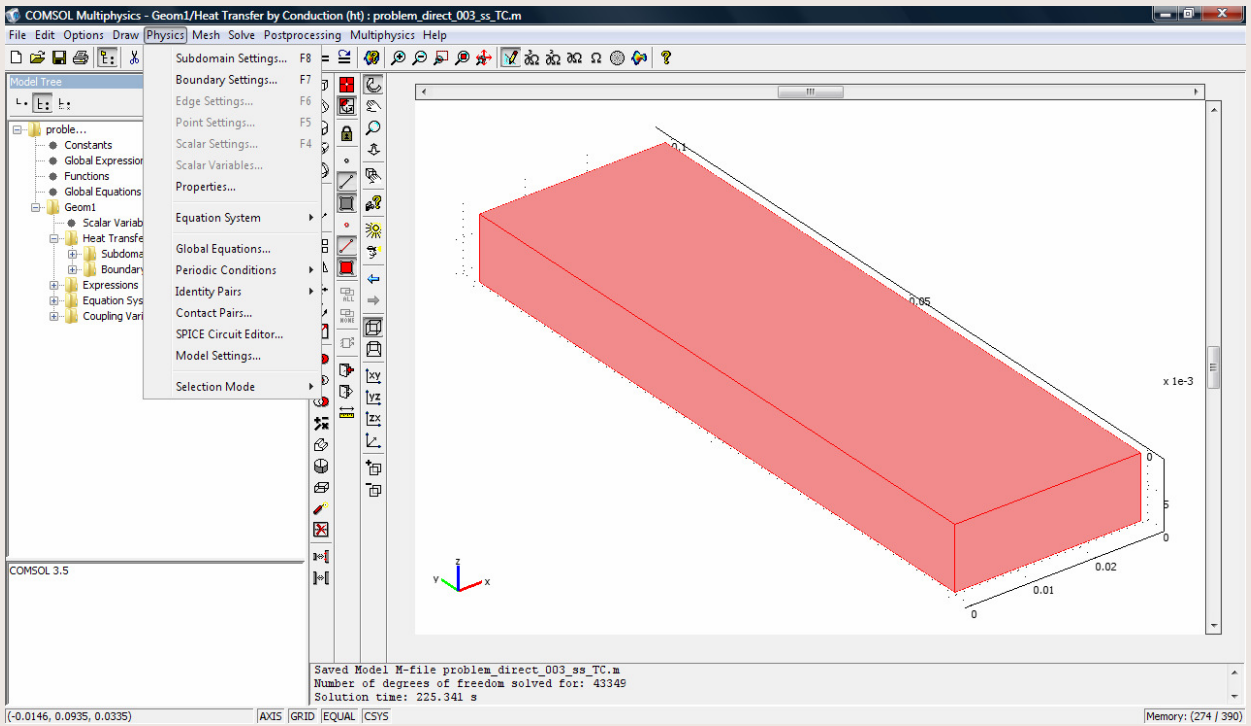


- Redimension the block with the « Zoom extents icon »

The direct welding problem

- the resolution with comsol -

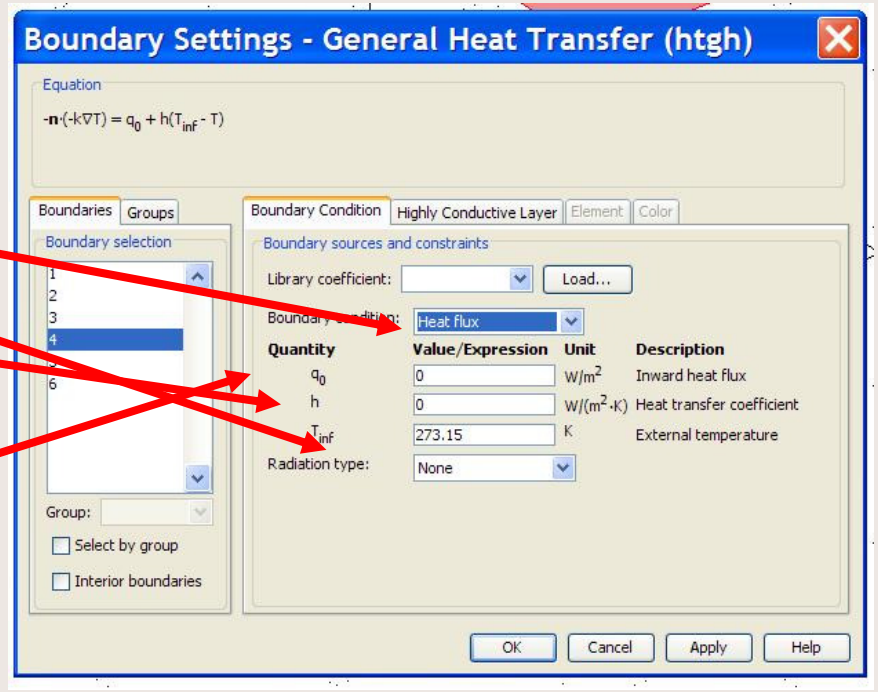
- In bar menu, choose « Physics » then « boundary settings » to define the boundary conditions.



The direct welding problem

- the resolution with comsol -

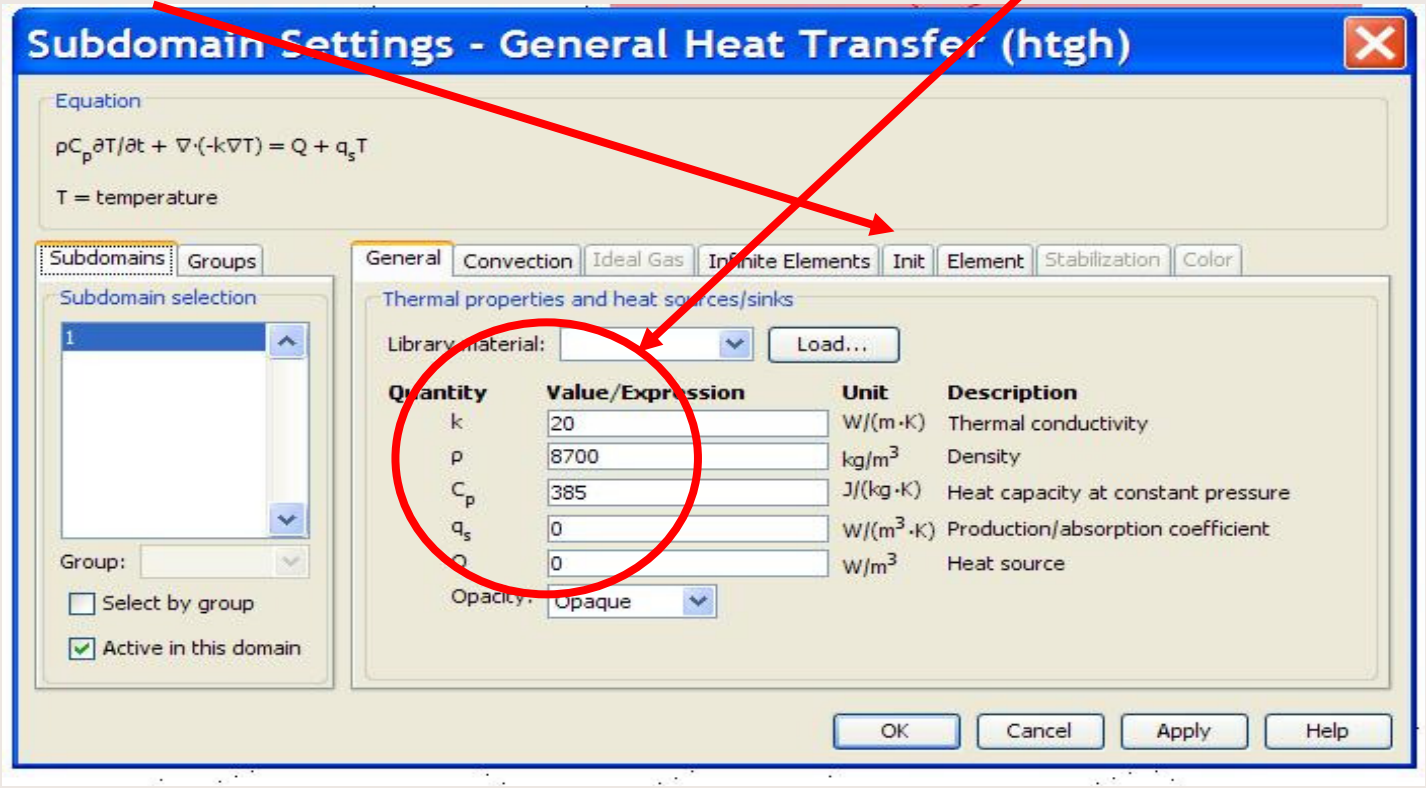
- Boundary 1: insulation / symmetry condition,
- Boundaries 2, 3, 5 and 6: select « heat flux » enter in « h » 10 and in « Tinf » box 20
- Boundary 4: select « heat flux » enter in « h » 10 and in « Tinf » box 20 and in « q0» box Gaussian
- APPLY and click OK



The direct welding problem

- the resolution with comsol -

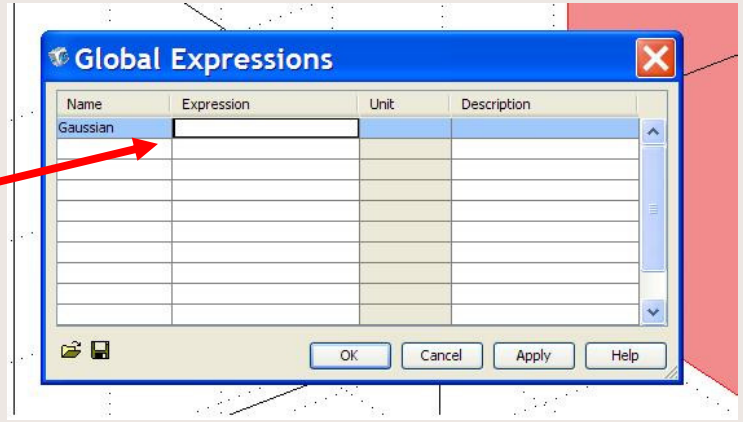
- In « Physics » menu bar, choose « subdomain settings » to define the material properties.
- The subdomain settings window appears and enter the properties and in the the init part, give the initial temperature .



The direct welding problem

- the resolution with comsol -

- Go to the « options » in the menu bar, choose « expressions » « global expressions » and define the expression: « Gaussian »
- In this expression, we have 3 constant parameters:
  - *Q =4000W*
  - *r =0.002 m*
  - *V =0.005m/s*

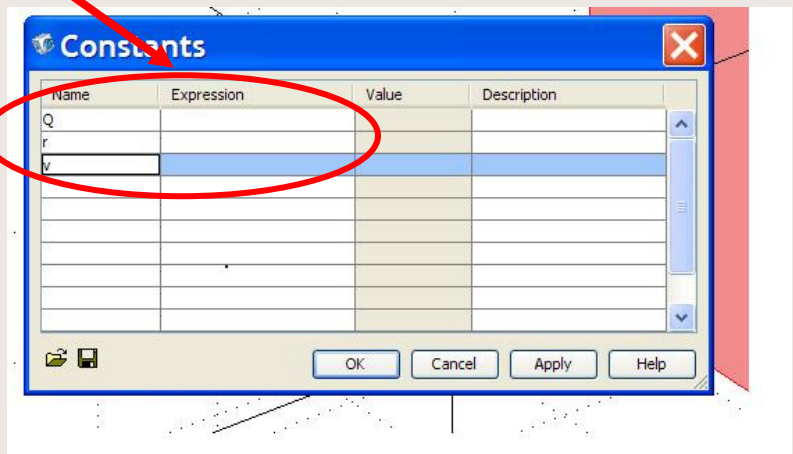


$$q_0(x,y,t) = \frac{Q}{2\pi r^2} exp\left(-\frac{\left(x^2 + (y - vt)^2\right)}{2r^2} \right)$$

The direct welding problem

- the resolution with comsol -

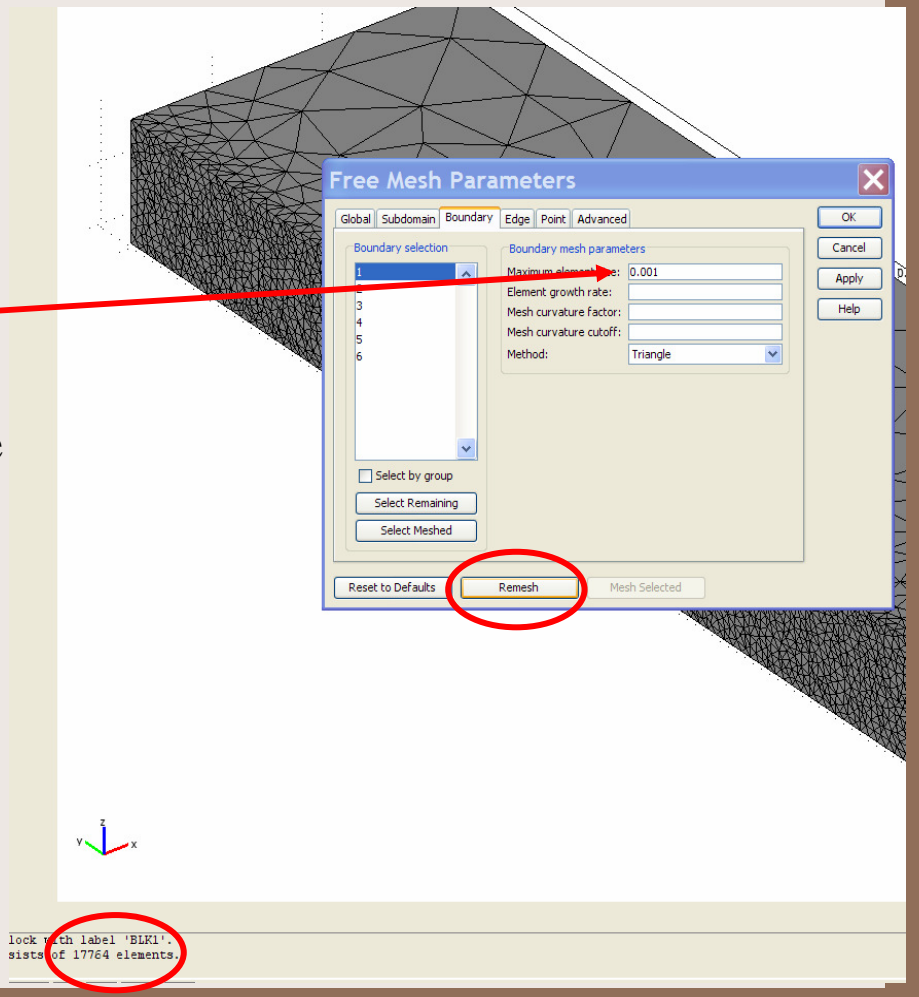
- Go to the « options » in the menu bar, choose « constants » to define all the parameters and their values.
  - *Q =4000W*
  - *r =0.002 m*
  - *V =0.005m/s*



The direct welding problem

- the resolution with Comsol -

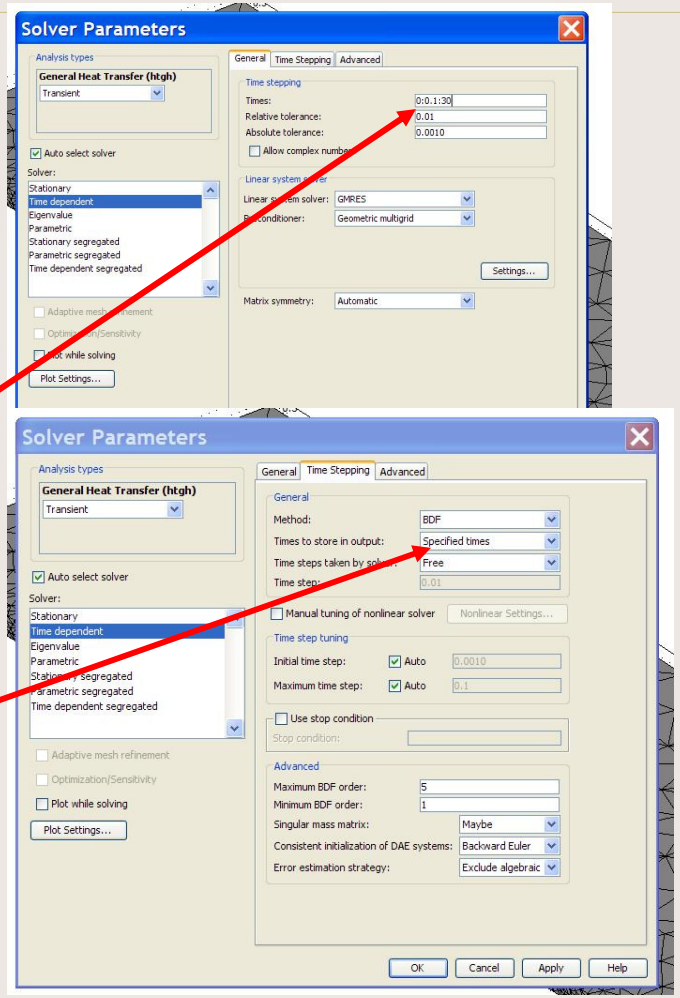
- Mesh step
- In the menu, select « mesh » then « Free mesh parameters » to open the mesh parameters window
- On the boundary 1, define the maximum element size and remesh
- At last, we have the number of the elements (you can change the maximum element size 0.001m)



The direct welding problem

- the resolution with Comsol -

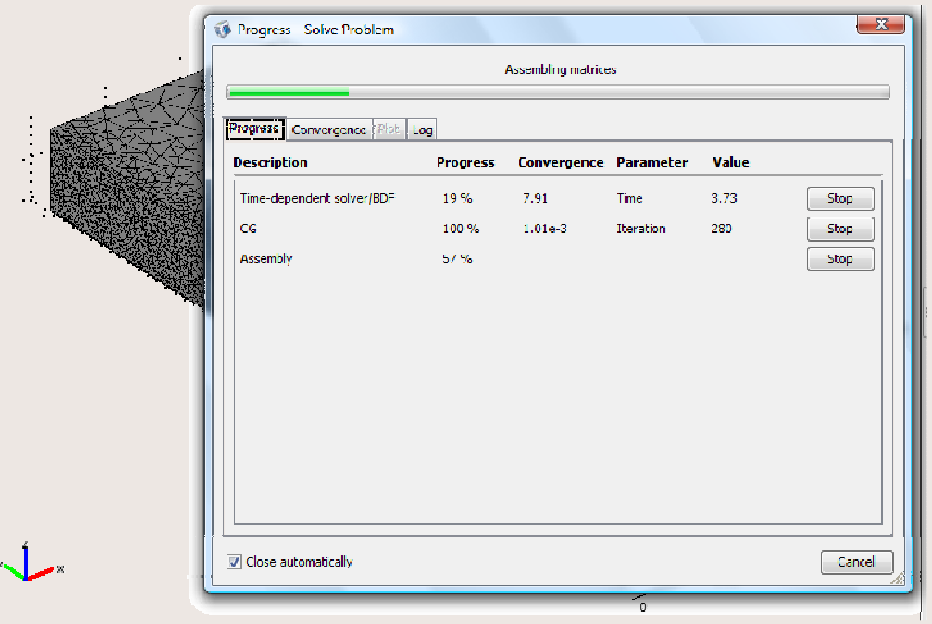
- Select « solve » in the bar menu,
- Then « Solver parameters » and click
- The solver parameters window appears
- In the « general » menu, verify that is a ‘time dependent’ problem in « solver type » and define « times » (0:0.1:20)
- Go to the « timestepping » menu and verify that we have « specified times » in « Times to store in ouput » menu.
- Click apply and OK



The direct welding problem

- the resolution with Comsol -

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



- We obtain the temperature field at the final time

The direct welding problem

- the resolution with Comsol -

- In the bar menu, choose « file » then « reset M-file » before solving again the direct problem
- Solve the direct welding problem by using the « solve » icon (symbol equal)
- We obtain the temperature field at the final time

- Save the data in a M-file.
- Go to « file » in the menu, choose « Save As » then « Model M-file ».
- The name is ‘direct’
- The program generates a direct.m file.

The direct problem

- Open your file ‘direct.m’

The direct welding problem

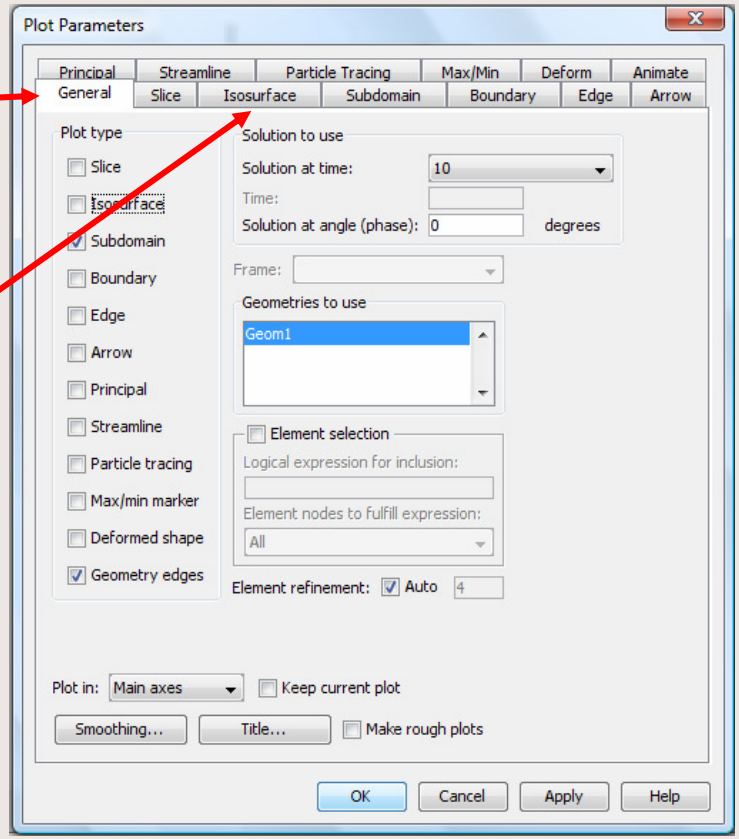
- the resolution with Comsol -

- Before the introduction of the Levenberg-Marquardt method, we define the measurement points where the temperatures still less than 1200°C.

The direct welding problem

- the resolution with Comsol -

- Take the « Postprocessing » menu
- « plot parameters… »
- In « General » check the « subdomain » and take a middle t = 10s
- Apply and look the thermal field.

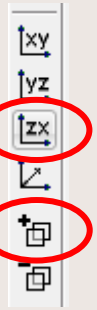


- In the “Postprocessing” menu
- Uncheck “Subdomain” but select “Isosurface”.
- Define in the “isosurface” menu three temperatures in “vector with isolelvels” :
  - 1450°C → limit of the fused zone
  - 1200°C → temperature measurement limit
  - 1100°C

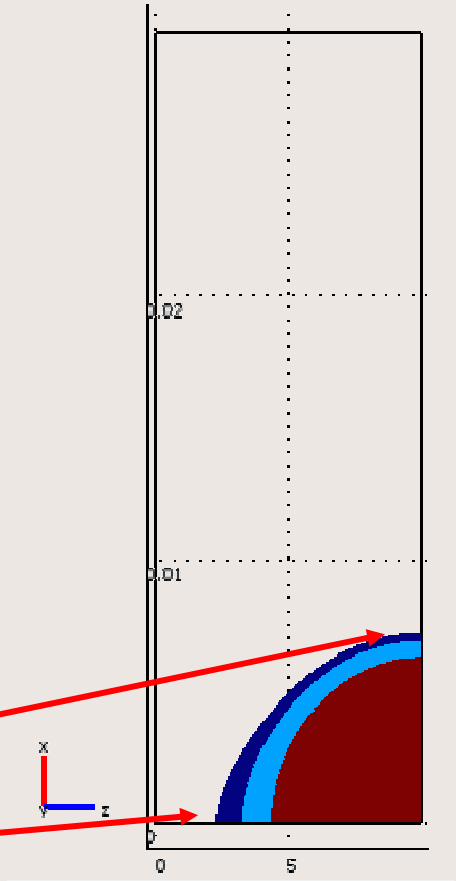
The direct welding problem

- the resolution with Comsol -

- In the tool bar Select “Go to ZX view”
- Click the “Increase transparency” icon



- We have in this case the three thermal levels.
- So, we can chose the measurement points :
  - (0.00634, 0.05, 0.008)
  - (0, 0.05, 0.0035)



The Levenberg-Marquardt method

- the inverse boundary problem formulation -

- The inverse boundary problem formulation [3]:

  Find the parameter Z={Q} which minimizes the quadratic criterion S(Z,T) :

$$S(Z,T) = \left[Y_i - T_i(Z) \right]^T W \left[Y_i - T_i(Z) \right]$$

  With Yi is the measurements, Ti 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. In fact here, W=I (we don't have noisy data)

- At each iteration, the parameters are calculated by [4,5]:

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

  where J is the sensitivity matrix, λ is the damping parameter and Ω is a diagonal matrix equal here 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 Levenberg-Marquardt method

- the sensitivity matrix -

- Sensitivity coefficients calculus [6]:

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

The expression of the sensitivity matrix becomes:

$$J_{\{Q\}} = \left[\frac{\partial T_1}{\partial Q} \; \frac{\partial T_2}{\partial Q} \; \frac{\partial T_3}{\partial Q} \; \frac{\partial T_4}{\partial Q} \; \ldots\ldots \frac{\partial T_I}{\partial Q} \right]^T$$

Stopping criterion: $S\left(Z^k, T\right) \le \varepsilon$

[6] M.N. Osizik, H.R.B. Orlande, Inverse heat transfer: fundamentals and applications, Taylor and Francis, New York, 2000.

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 $T\left(Z^k\right)$.

2-Compute $S\left(Z^k, T\left(Z^k\right)\right)$.

3-Compute the sensitivity matrix $J\left(Z^k\right)$ and the matrix Ω^k.

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

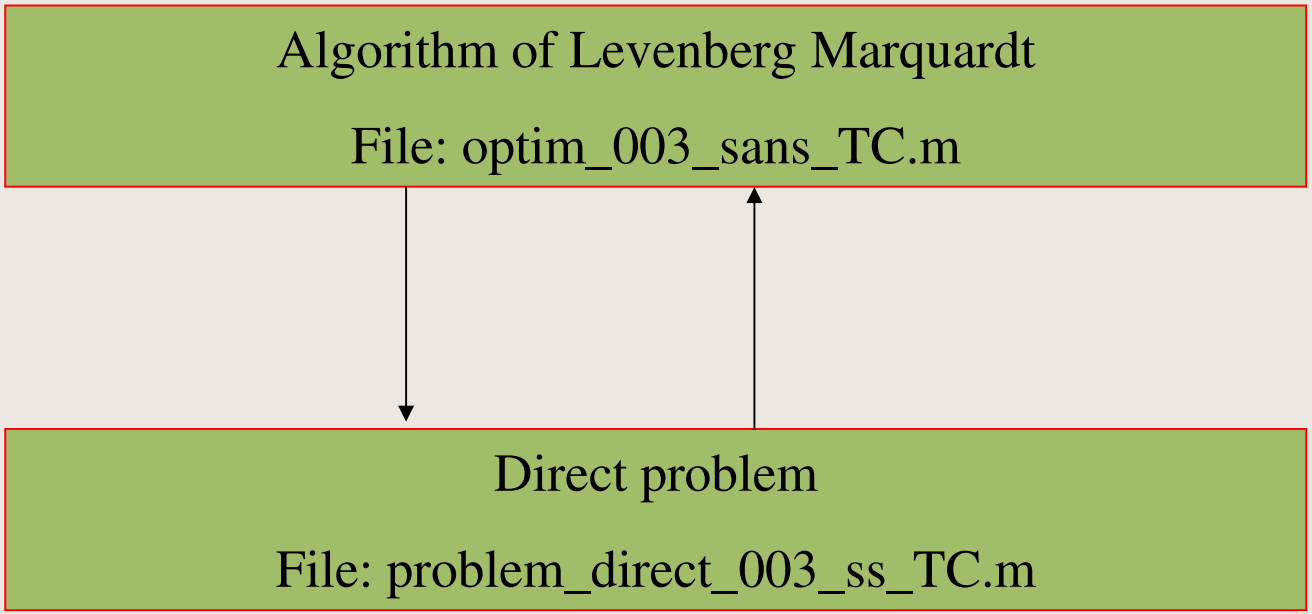
5-Solve the direct problem with Z^{k+1}, Compute $S\left(Z^{k+1}, T\left(Z^{k+1}\right)\right)$.

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

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

[6] M.N. Osizik, H.R.B. Orlande, Inverse heat transfer: fundamentals and applications, Taylor and Francis, New York, 2000.

The inverse algorithm



Open the two files: problem_direct_003_ss_TC.m

and

optim_003_sans_TC.m

Execution of the estimation

- Now we can run the estimation:
  - Execute: optim_003_sans_TC.m
  - The execution shows at each iteration:
    - The criterion
    - The parameter Q
  - Now, we can examine the three files:
    - ‘direct.m’
    - ‘problem_direct_003_ss_TC.m’
    - ‘optim_003_sans_TC.m’

Modification of the “direct.m” file

In this direct file (problem_direct_003_ss_TC.m), we have only the parameter Q which is modified:

We add in the first line:

“global P1” for the modification of Q during the estimation

And

Replace ‘4000’ by P1

```
% COMSOL Multiphysics Model M-file
% Generated by COMSOL 3.5 (COMSOL 3.5.0.494, $Date: 2008/09/19 16:09:48 $)
global p1
clear fem

% COMSOL version
clear vrsn
vrsn.name = 'COMSOL 3.5';
vrsn.ext = '';
vrsn.major = 0;
vrsn.build = 494;
vrsn.rcs = '$Name:  $';
vrsn.date = '$Date: 2008/09/19 16:09:48 $';
fem.version = vrsn;

% Geometry
g1=block3('0.03','0.1','0.01','base','corner','pos',{'0','0','0'},'axis',{'0','0',

% Analyzed geometry
clear s
s.objs={g1};
s.name={'BLK1'};
s.tags={'g1'};

fem.draw=struct('s',s);
fem.geom=geomcsg(fem);

% Constants
fem.const = {'P',p1, ...
  'r0','1e-3'};

% Initialize mesh
fem.mesh=meshinit(fem, ...
                  'hauto',5, ...
                  'hmaxedg',[4,0], ...
                  'hmaxfac',[1,0.001], ...
                  'hgradsub',[1,1.3], ...
                  'hmaxsub',[1,1e-2]);

% (Default values are not included)

% Application mode 1
clear appl
appl.mode.class = 'HeatTransfer';
appl.sshape = 2;
appl.assignsuffix = '_ht';
clear bnd
bnd.q0 = {0,0,'q0'};
```

The Levenberg Marquardt file

Open the optim_003_sans_TC.m

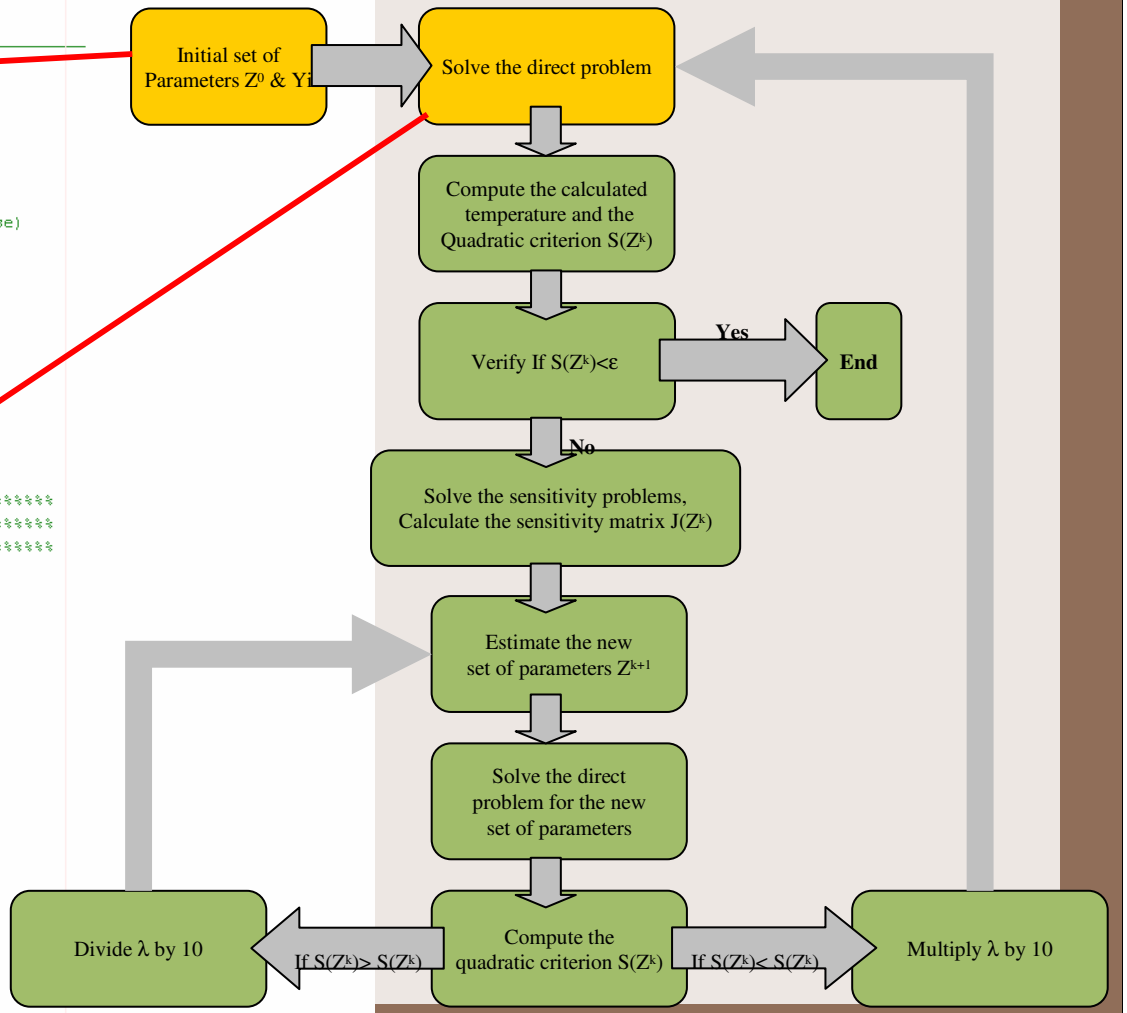
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% the estimated parameter is:
% p1    :   the gaussian amplitude ref = 4000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all; clear all; clc

global p1

epsv=1e-4;
% Definition of the measured points %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mx=[0 0.00634];
my=[0.05 0.05];
mz=[0.0035 0.008];
pp=[mx;my;mz];

% the theoretical value of Q=4000

P1=4000;

% Resolution of the direct problem ____________________________________

problem_direct_003_ss_TC

% the theoretical temperatures

TCme=postinterp(fem,'T',pp,'solnum','all');
for j=1:length(TCme(1,:))
    for i=1:length(TCme(:,1))
        TCmep(i+length(TCme(:,1))*(j-1))=TCme(i,j);
    end
end

TCmes=TCmep';

```

$$J_Z = \frac{T(Z+\varepsilon Z) - T(Z-\varepsilon Z)}{2\varepsilon Z}$$

The Levenberg Marquardt file

Open the optim_003_sans_TC.m

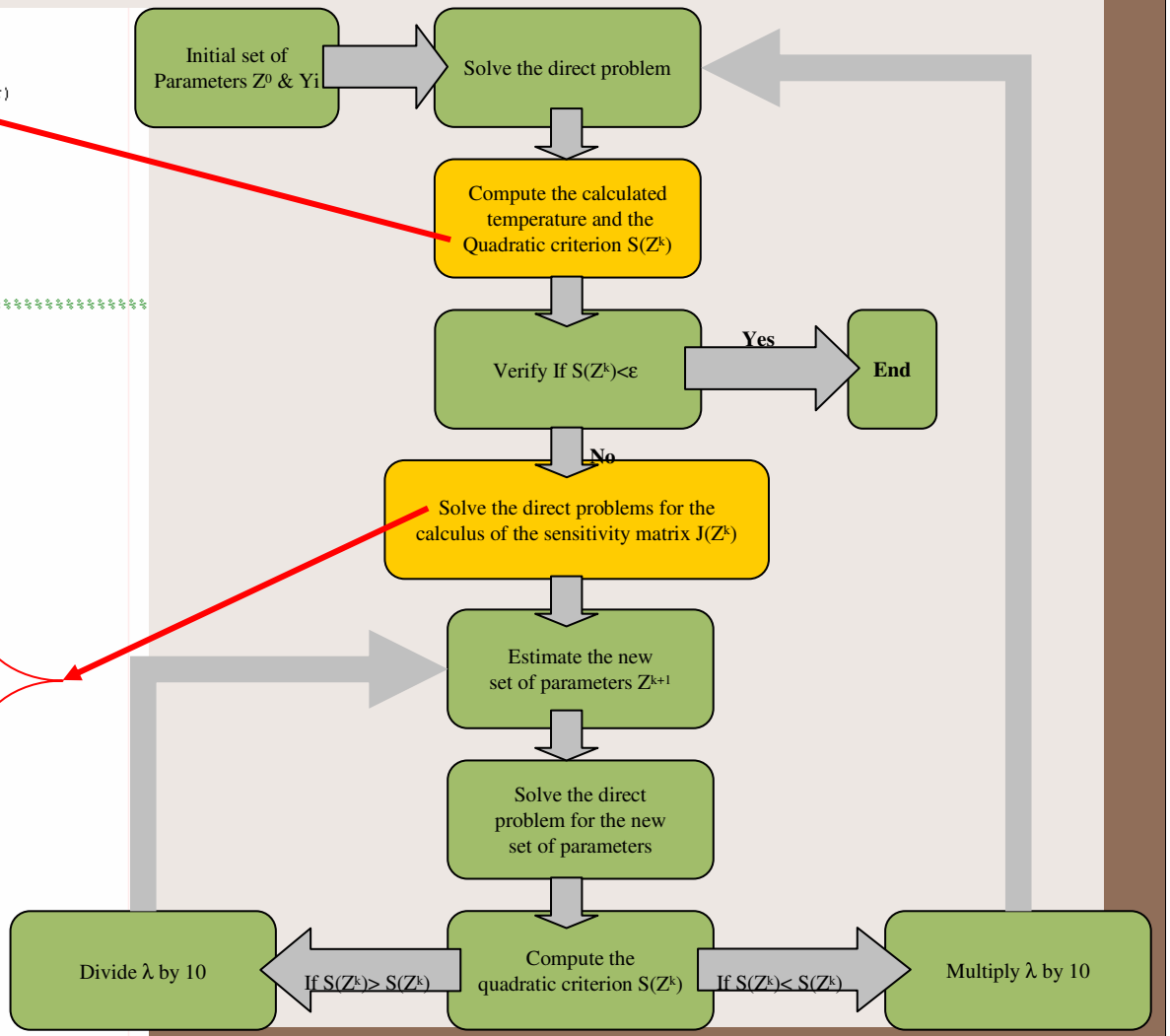
```
% ************************ beginning of the inverse procedure ****************************
% the levenberg-Marquardt method: Pk+1 = Pk + inv{[Jh'*WJ*Jh+lambda(k)*OMEG(k)]}*{Jh'*WJ*[Yi-Tcal(Pk)]}

% Initialisation of the unknown parameter Q _____________________________

p1=100;
p1prec=p1;

% Initialisation of the damping parameter of the Levenberg Marquardt
lambda=0.001;

% Creation of the matrix WJ egal to the identity matrix (whitout noise)
WJ=eye(1);

% Creation of the matrix OMEG egal to the identity matrix
OMEG=eye(1);

% Initialisation of the quadratic criterion
somme=(norm(TCmes)*norm(TCmes));

% first iteration
niter=1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%% beginning of the optimisation loop %%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

while ((somme>1e-3)&(niter<100))

    % Resolution of the direct problem with the new parameter Q

    problem_direct_003_ss_TC

    % Definition of the calculated temperatures

    clear TCsi
    clear TCsim

    TCsi=postinterp(fem,'T',pp,'solnum','all');
    for j=1:length(TCsi(1,:))
        for i=1:length(TCsi(:,1))
            TCsip(i+length(TCsi(:,1))*(j-1))=TCsi(i,j);
        end
    end

    TCsim=TCsip';
```



The Levenberg Marquardt file

Open the optim_003_sans_TC.m

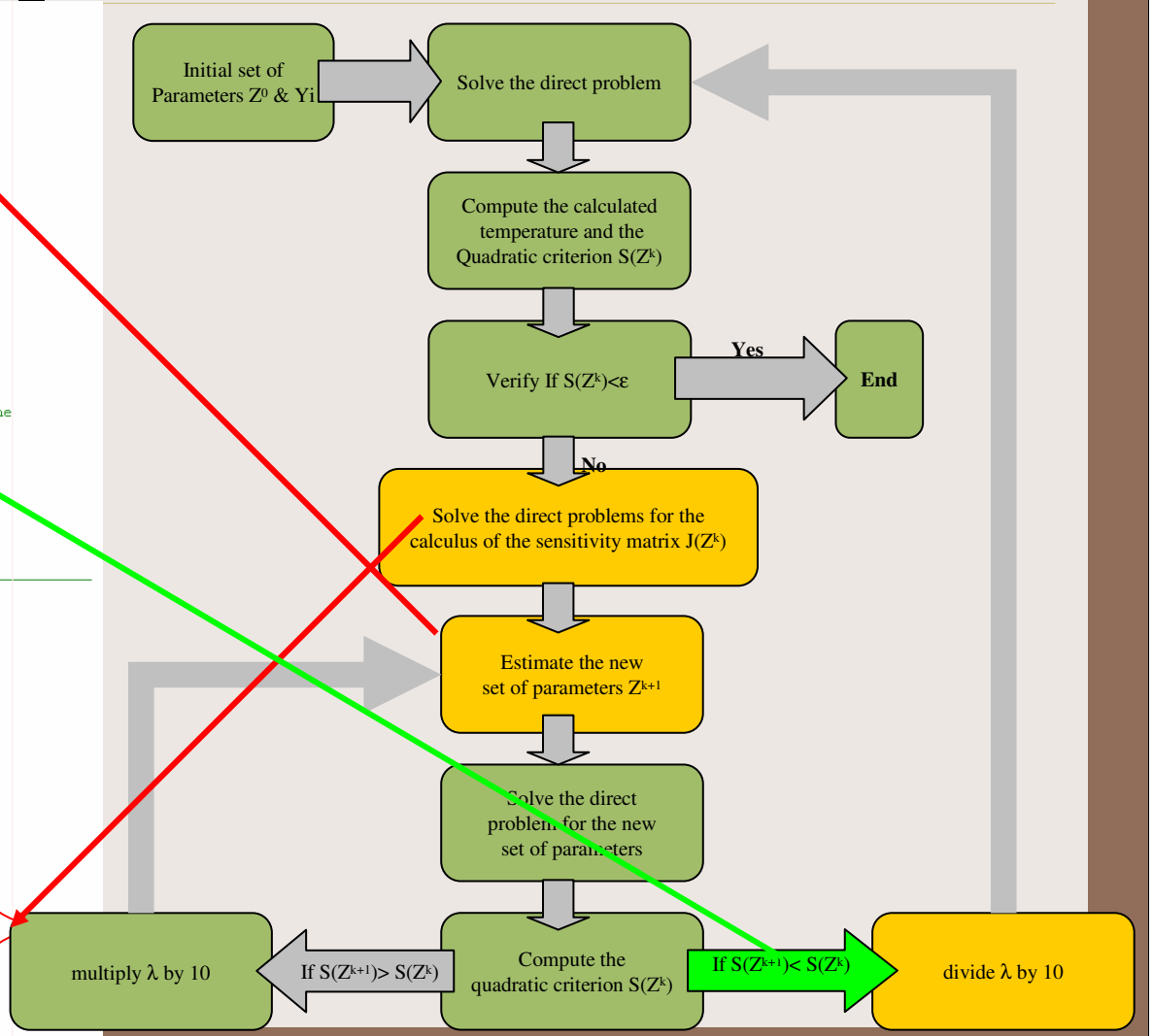
```
84      % Calculus of the quadratic criterion
85 -    SOM=TCmes-TCsim;
86 -    somme=norm(SOM)*norm(SOM);
87 -    sprintf('critere %18.3f at the iteration %1.1f',somme,niter)
88
89 -    if (niter==1)
90
91          % Cas niter=1 (first itération)
92 -        somprec=somme;
93 -        EVOLp1(niter)=p1;
94 -        EVOLSOM(niter)=somme;
95
96  %%%%%%%%% Calculus of the sensitivity coefficients %%%%%%%%%%%%%%%%%%%%%%%%%
97
98          % Parameter 1
99 -        p1=p1*(1+epsv);
100 -       problem_direct_003_ss_TC
101 -       clear TCsip
102 -       clear TCsimplus
103 -       TCsi=postinterp(fem,'T',pp,'solnum','all');
104 -       for j=1:length(TCsi(1,:))
105 -           for i=1:length(TCsi(:,1))
106 -               TCsip(i+length(TCsi(:,1))*(j-1))=TCsi(i,j);
107 -           end
108 -       end
109 -       TCsimplus=TCsip';
110 -       p1=EVOLp1(niter);
111
112 -       p1=p1*(1-epsv);
113 -       problem_direct_003_ss_TC
114 -       clear TCsip
115 -       clear TCsimmins
116 -       TCsi=postinterp(fem,'T',pp,'solnum','all');
117 -       for j=1:length(TCsi(1,:))
118 -           for i=1:length(TCsi(:,1))
119 -               TCsip(i+length(TCsi(:,1))*(j-1))=TCsi(i,j);
120 -           end
121 -       end
122 -       TCsimmins=TCsip';
123 -       p1=EVOLp1(niter);
124
125 -       Ja=(TCsimplus-TCsimmins)/(2*epsv*p1);
126 -       Av=1  ;
127
128 -       J=[Ja];
```



The Levenberg Marquardt file

Open the optim_003_sans_TC.m

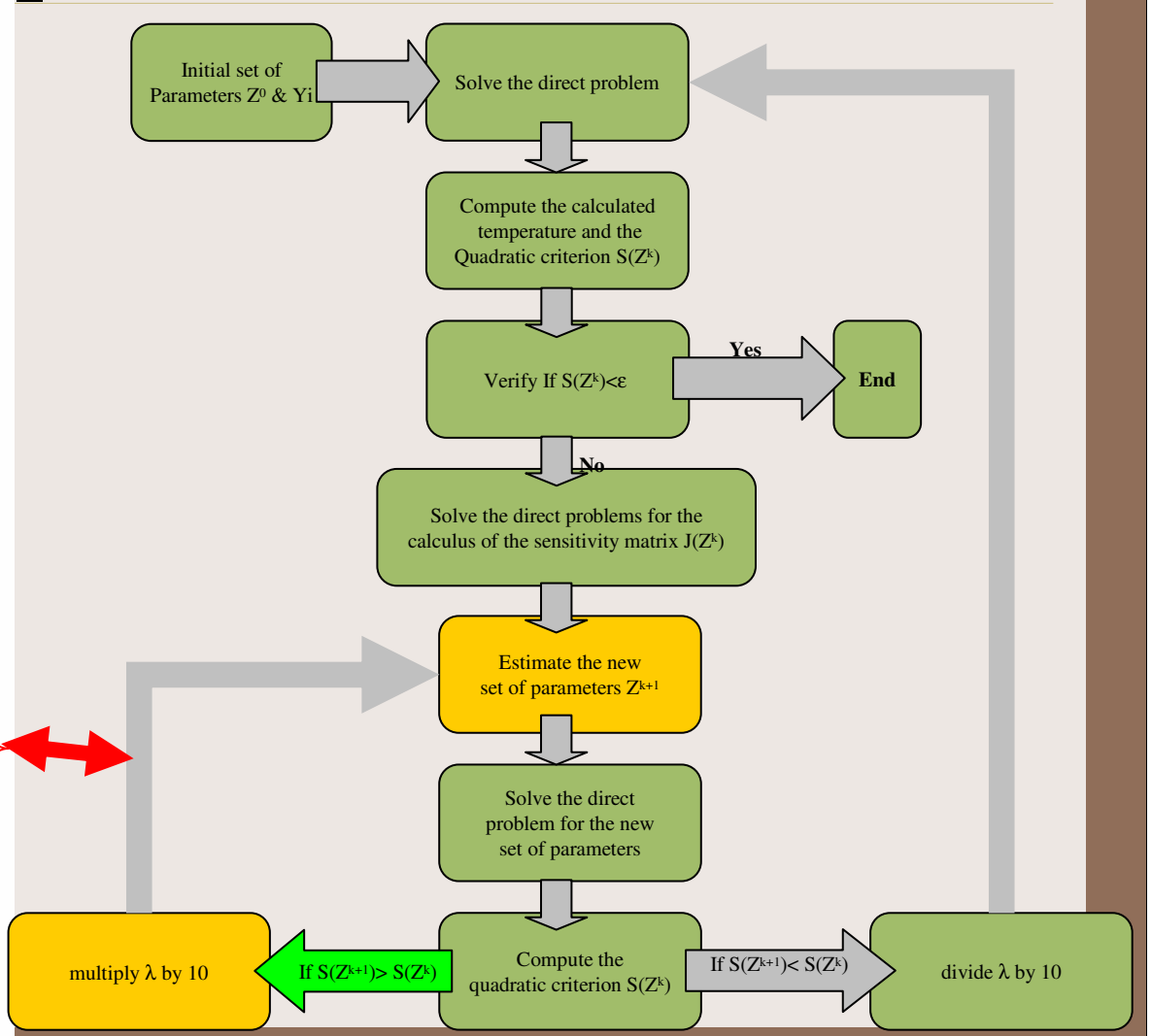
```
130         % Estimation of the parameter variation
131
132 -       dcoef=inv((J'*WJ*J+lambda*OMEG))*(J'*WJ*SOM(1:length(TCmes)));
133
134 -       p1=p1prec+dcoef(1);
135 -       if p1<1
136 -           p1=EVOLp1(niter);
137 -       elseif p1>50000
138 -           p1=EVOLp1(niter);
139 -       end
140
141 -       sprintf('the new estimated value \nfor p1: %6.5f\n',p1)
142
143 -       niter=niter+1;
144 -       p1prec=p1;
145
146 -   else
147
148 -       if (somme<somprec)
149
150             % Case where somme<somprec (the new criterion is lower than the
151             % next. The damping parameter is divided by 10
152
153 -           lambda = 0.1*lambda
154 -           somprec=somme;
155 -           EVOLp1(niter)=p1;
156 -           EVOLSOM(niter)=somme;
157
158             % Calculus of the sensitivity coefficients_____________________
159             % Parameter 1
160 -           p1=p1*(1+epsv);
161 -           problem_direct_003_ss_TC
162 -           clear TCsip
163 -           clear TCsimplus
164 -           TCsi=postinterp(fem,'T',pp,'solnum','all');
165 -           for j=1:length(TCsi(1,:))
166 -               for i=1:length(TCsi(:,1))
167 -                   TCsip(i+length(TCsi(:,1))*(j-1))=TCsi(i,j);
168 -               end
169 -           end
170 -           TCsimplus=TCsip';
171 -           p1=EVOLp1(niter);
172
173 -           p1=p1*(1-epsv);
174 -           problem_direct_003_ss_TC
175 -           clear TCsip
176 -           clear TCsimmins
177 -           TCsi=postinterp(fem,'T',pp,'solnum','all');
178 -           for j=1:length(TCsi(1,:))
179 -               for i=1:length(TCsi(:,1))
180 -                   TCsip(i+length(TCsi(:,1))*(j-1))=TCsi(i,j);
181 -               end
182 -           end
183 -           TCsimmins=TCsip';
```



The Levenberg Marquardt file

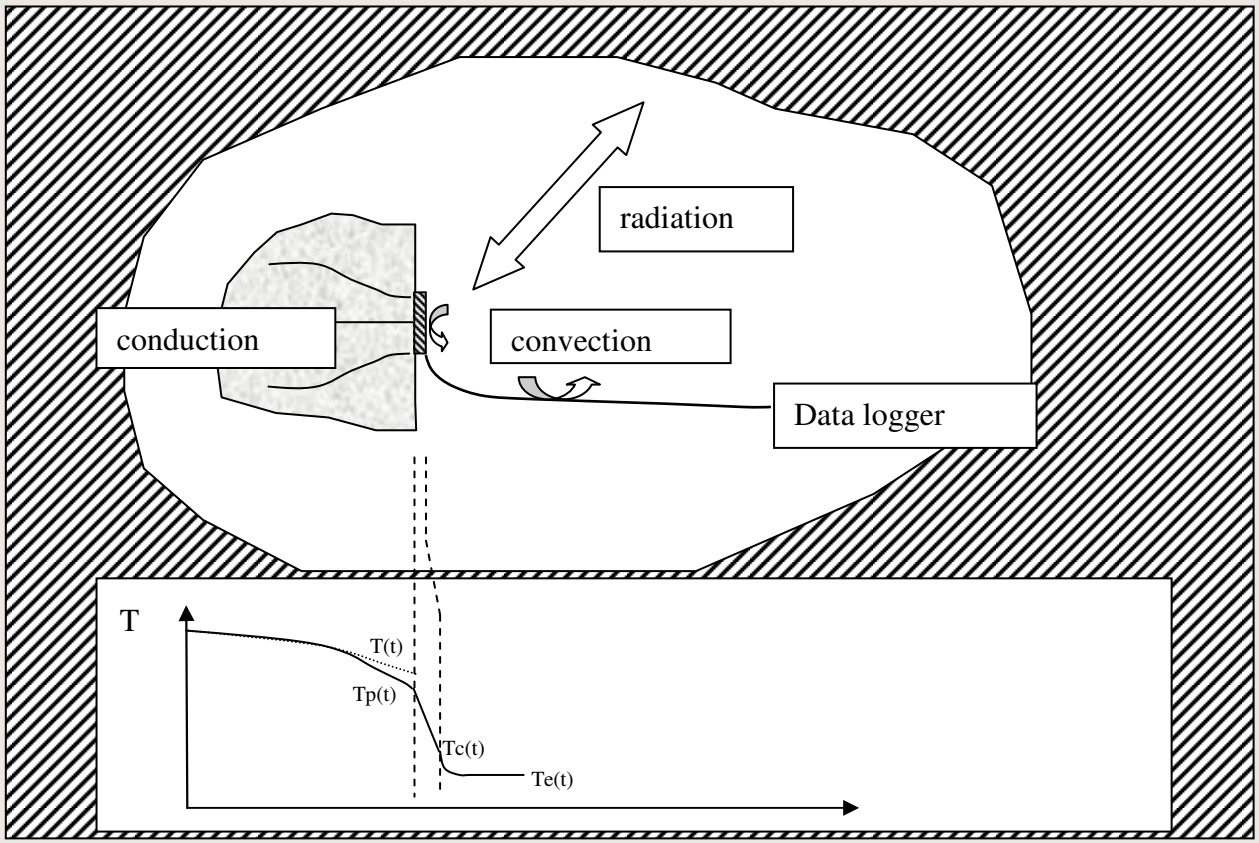
Open the optim_003_sans_TC.m

```
183 -               TCsimmins=TCsip';
184 -               p1=EVOLp1(niter);
185
186 -               Ja=(TCsimplus-TCsimmins)/(2*epsv*p1);
187
188 -               J=[Ja];
189
190           % Estimation of the parameter variation
191
192 -         dcoef=inv((J'*WJ*J+lambda*OMEG))*(J'*WJ*SOM(1:length(TCmes)));
193
194 -         p1=p1prec+dcoef(1);
195 -         if p1<1
196 -             p1=EVOLp1(niter);
197 -         elseif p1>50000
198 -             p1=EVOLp1(niter);
199 -         end
200
201 -         sprintf('the new estimated value \nfor p1: %6.5f\n',p1)
202
203 -         niter=niter+1;
204 -         p1prec=p1;
205
206 -         else
207
208               % Case where somme>somprec (the new criterion is upper than the
209               % next. The damping parameter is multiplied by 10
210
211 -             lambda=10*lambda
212
213 -             somprec=somme;
214 -             EVOLp1(niter)=p1;
215
216 -             EVOLSOM(niter)=somme;
217
218           % Estimation of the parameter variation
219 -         dcoef=inv((J'*WJ*J+lambda*OMEG))*(J'*WJ*SOM(1:length(TCmes)));
220
221 -         p1=p1prec+dcoef(1);
222 -         if p1<1
223 -             p1=EVOLp1(niter);
224 -         elseif p1>50000
225 -             p1=EVOLp1(niter);
226 -         end
227
228 -         sprintf('the new estimated value \nfor  p1: %6.5f\n',p1)
229
230 -         niter=niter+1;
231 -         p1prec=p1;
232
233 -         end
234
235 -     end
236
237 - end
```



The modelisation of the error

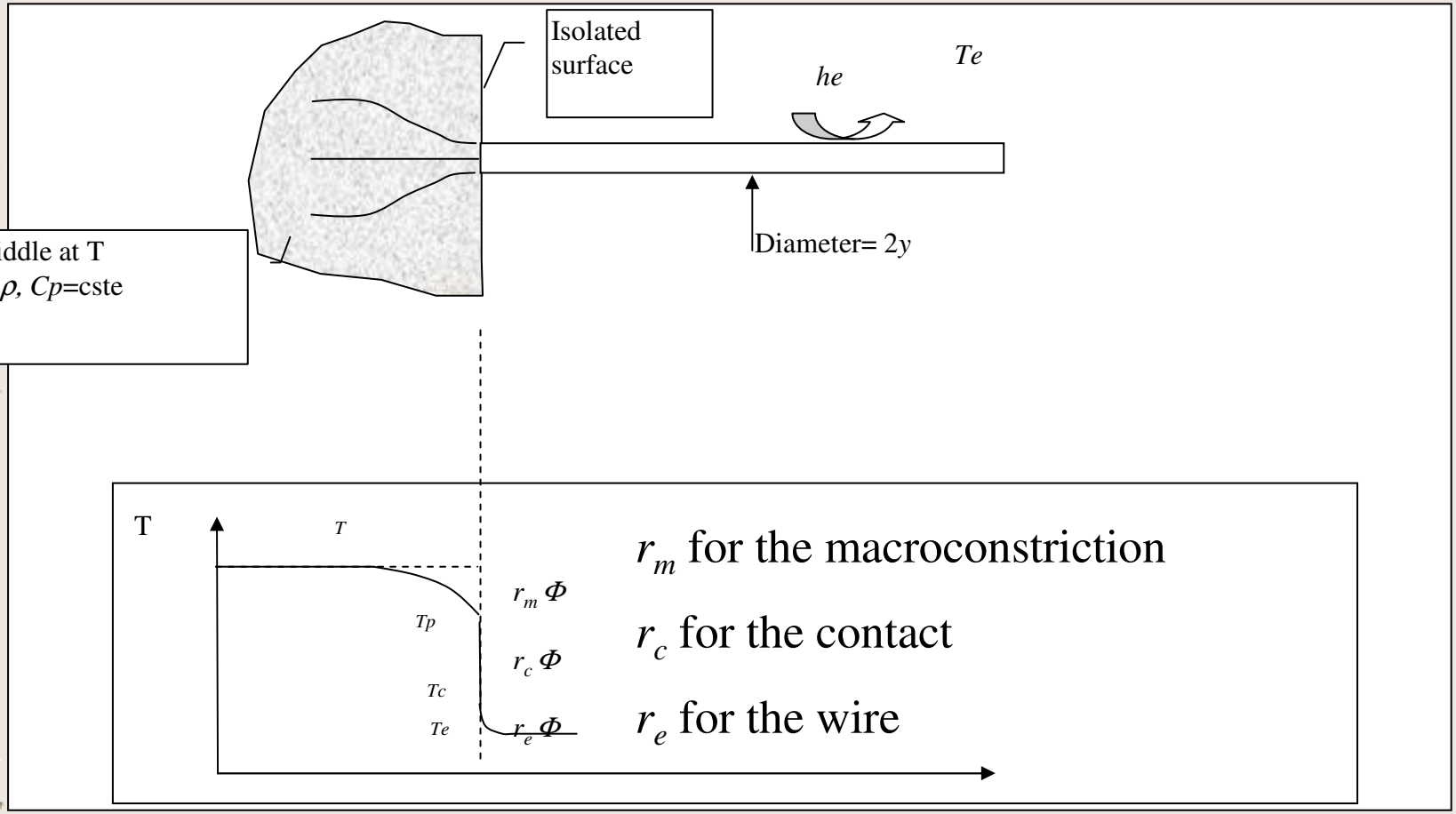
- In a steady state case and for a thermocouple on a surface, we have this scheme: $\varepsilon(t) = T(t) - Tc(t)$



[7] Bardon J.P., Cassagne B. "Température de surface: mesures par contact" Techniques de l'Ingénieur, Paris R2732, 1-22, 1981

The modelisation of the error

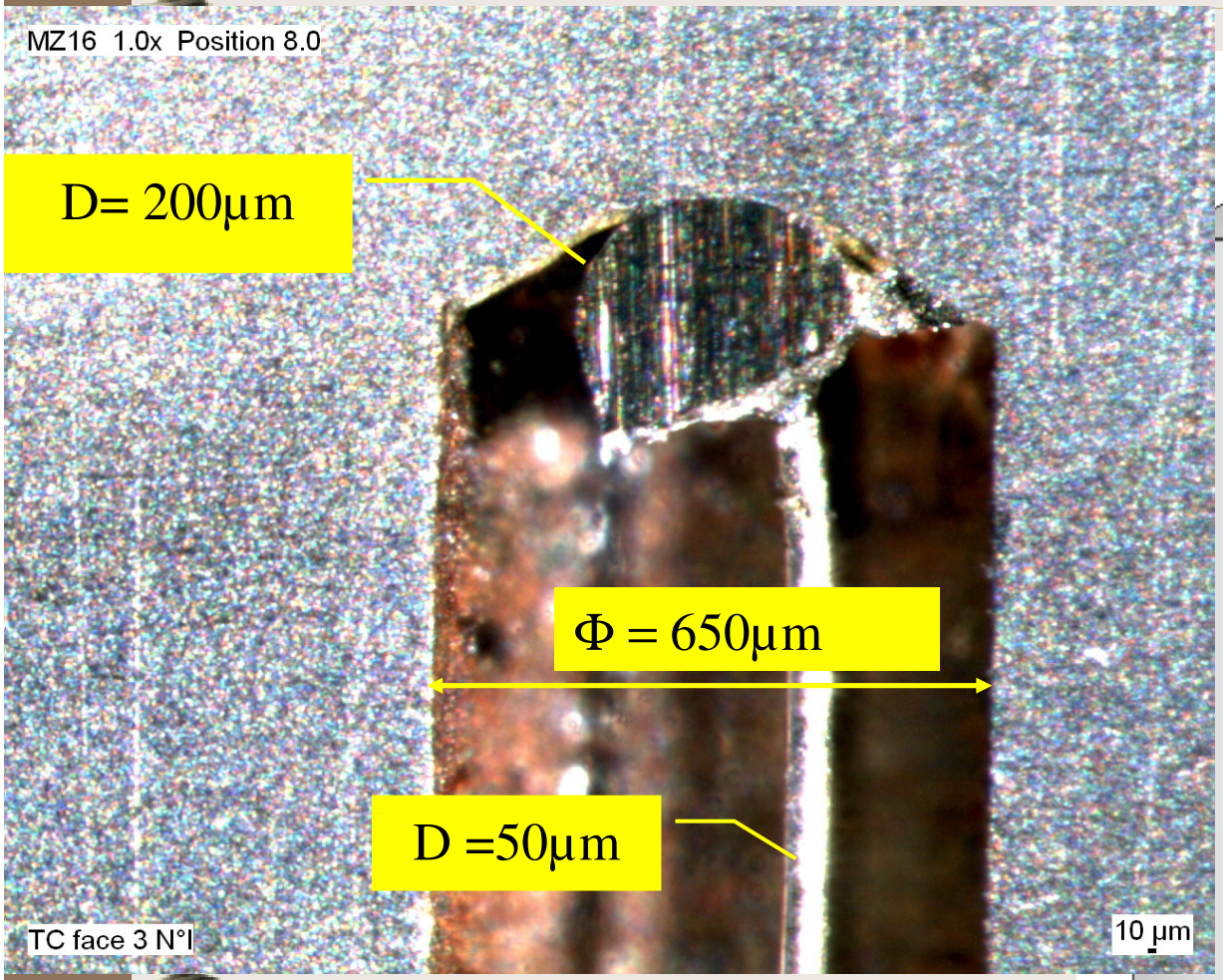
- For this case, in the model in steady state, we define three resistances

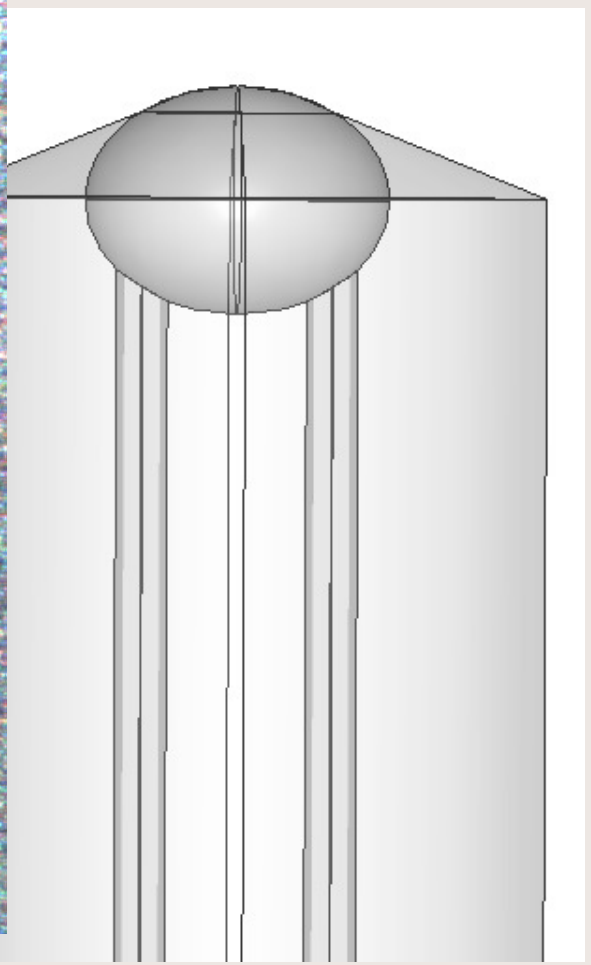


Modelisation of the welding problem with thermocouples

- With the software « Comsol Multiphysics », we realize a simulation of the welding problem. But now, we modelise the thermocouples.
- Two configurations are studied:
  - For the first one, the holes for the thermocouples are perpendiculars of the heat flux and the fused zone.
  - For the second, the holes are parallels of the fused zone.
- Moreover, we compute different contact resistances between the thermocouples and the material ($Rc = e/\lambda$, $\lambda = 0.025$ W/m/K):
  - $Rc = 10^{-3}$ or 10^{-4} m²K/W for a bad contact (e = 25µm or 2.5µm)
  - $Rc = 10^{-5}$ or 10^{-6} m²K/W for a mean contact (e = 0.25µm or 0.025µm)
  - $Rc = 10^{-7}$ m²K/W for a good contact (e = 0.0025µm)

The modelisation of the thermocouples





Results of the first estimation

- Analyze the results:
  - The criterion decreases
  - After the first iteration, we have:

iteration	Q	Criterion
Initial values	100	187 10^6
First	3988	1800
second	4000	0.001

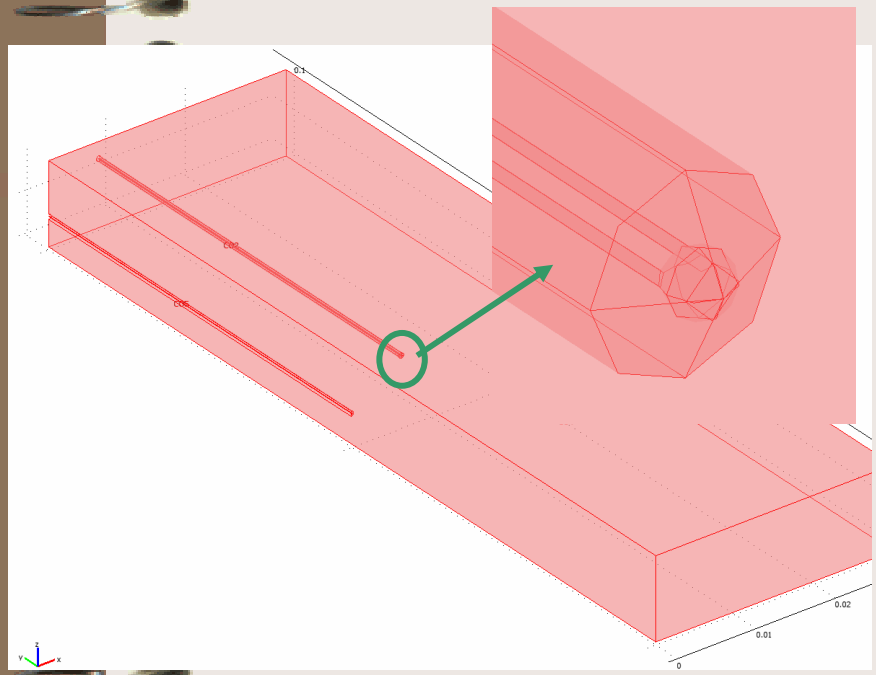
  - After 2 iterations, we obtain the good value Q = 4000

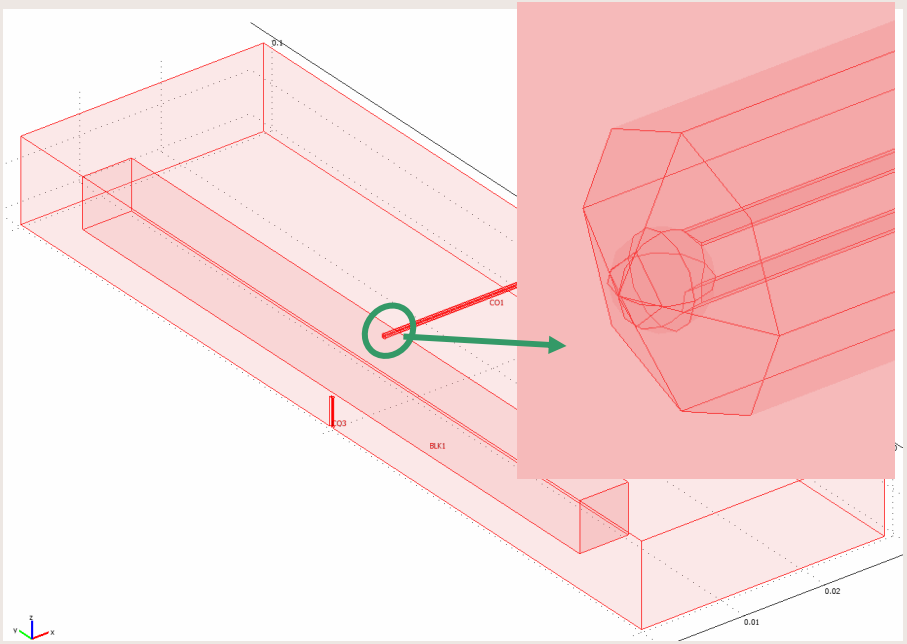
Modelisation of the welding problem with thermocouples

Open with “Comsol Multiphysics” the two configurations.

Parallele.mph

Perpendicular.mph





Modelisation of the welding problem with thermocouples

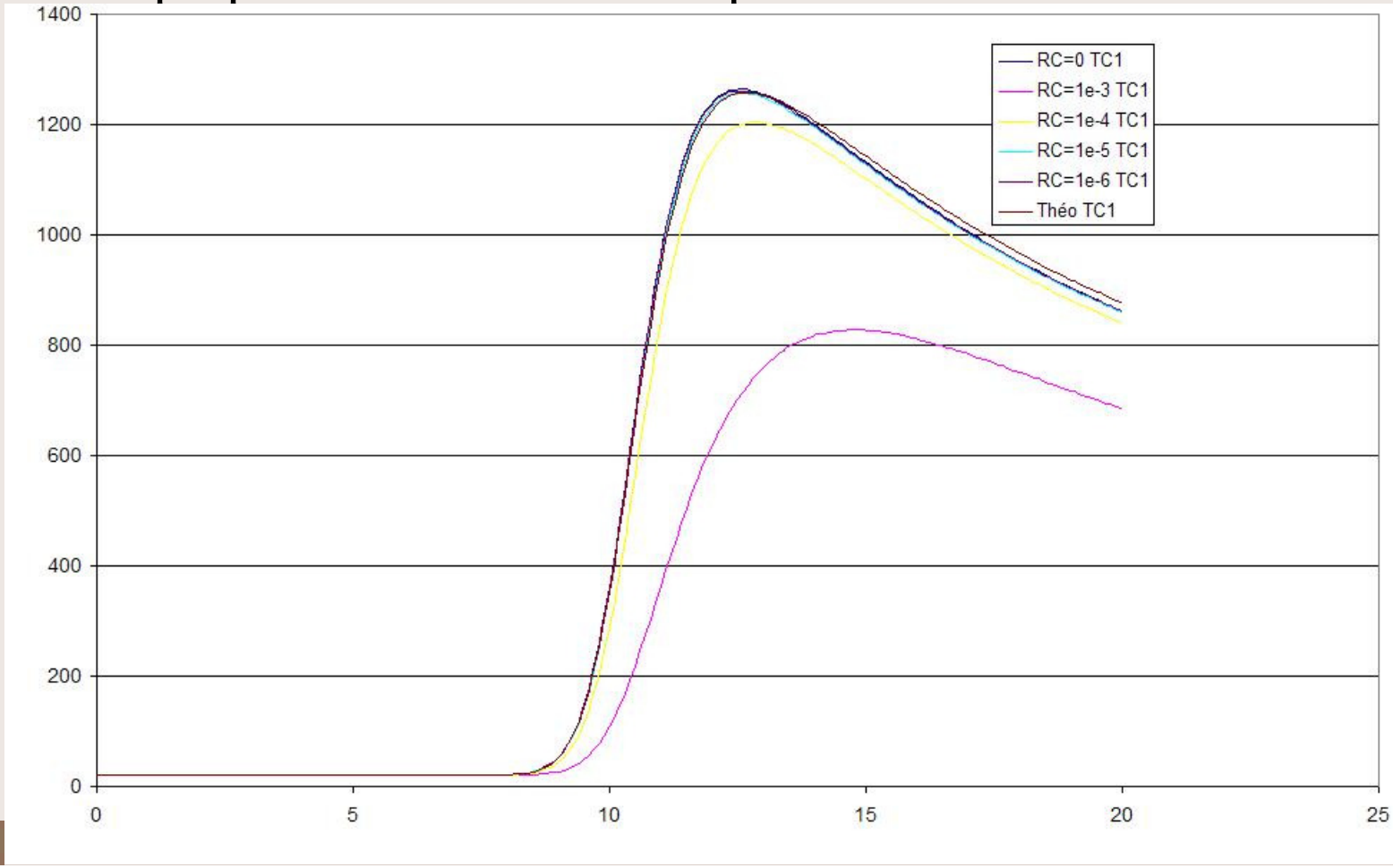
We execute these configurations with different contact resistances and we use the thermogrammes in the first optimisation loop with a direct problem without the thermocouples.

With this work, we can underline:

- The measurement errors
- The estimation errors of Q

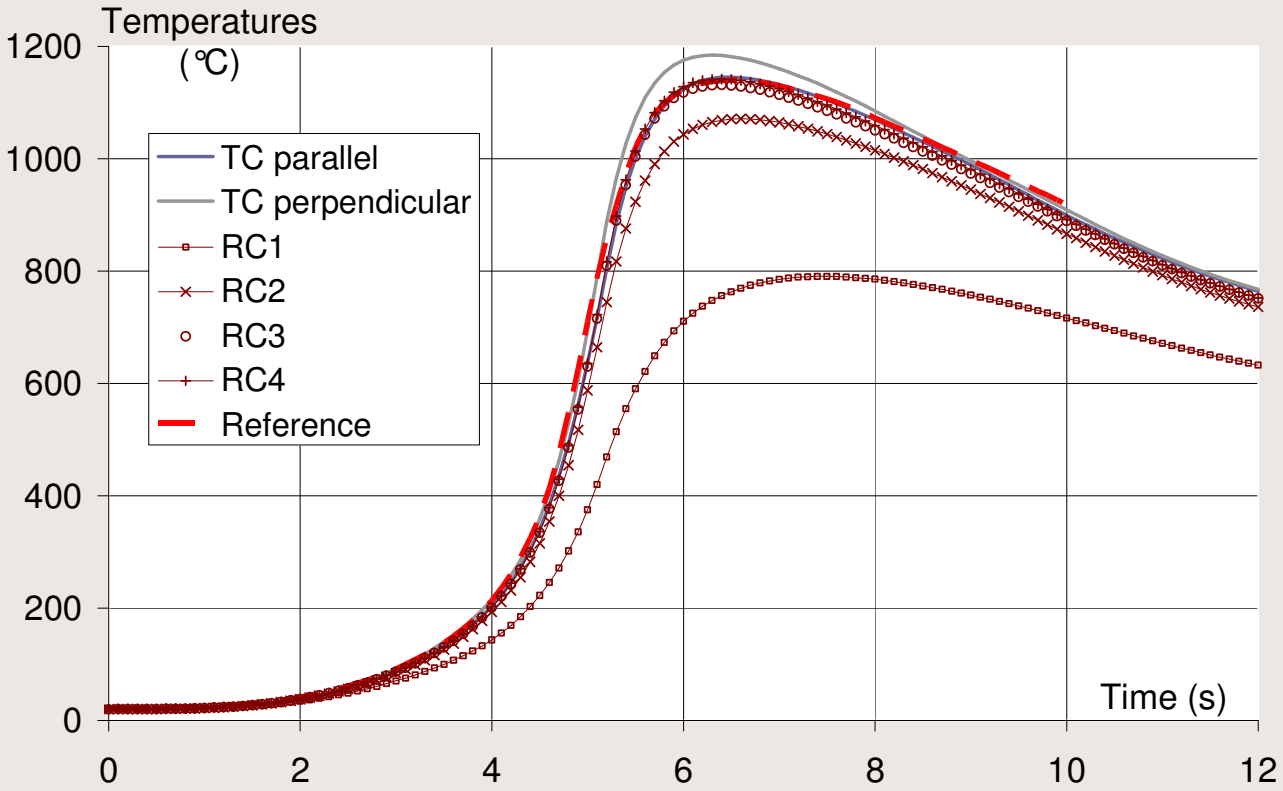
Modelisation of the welding problem with thermocouples

Visualisation of the measurements errors for perpendicular thermocouples

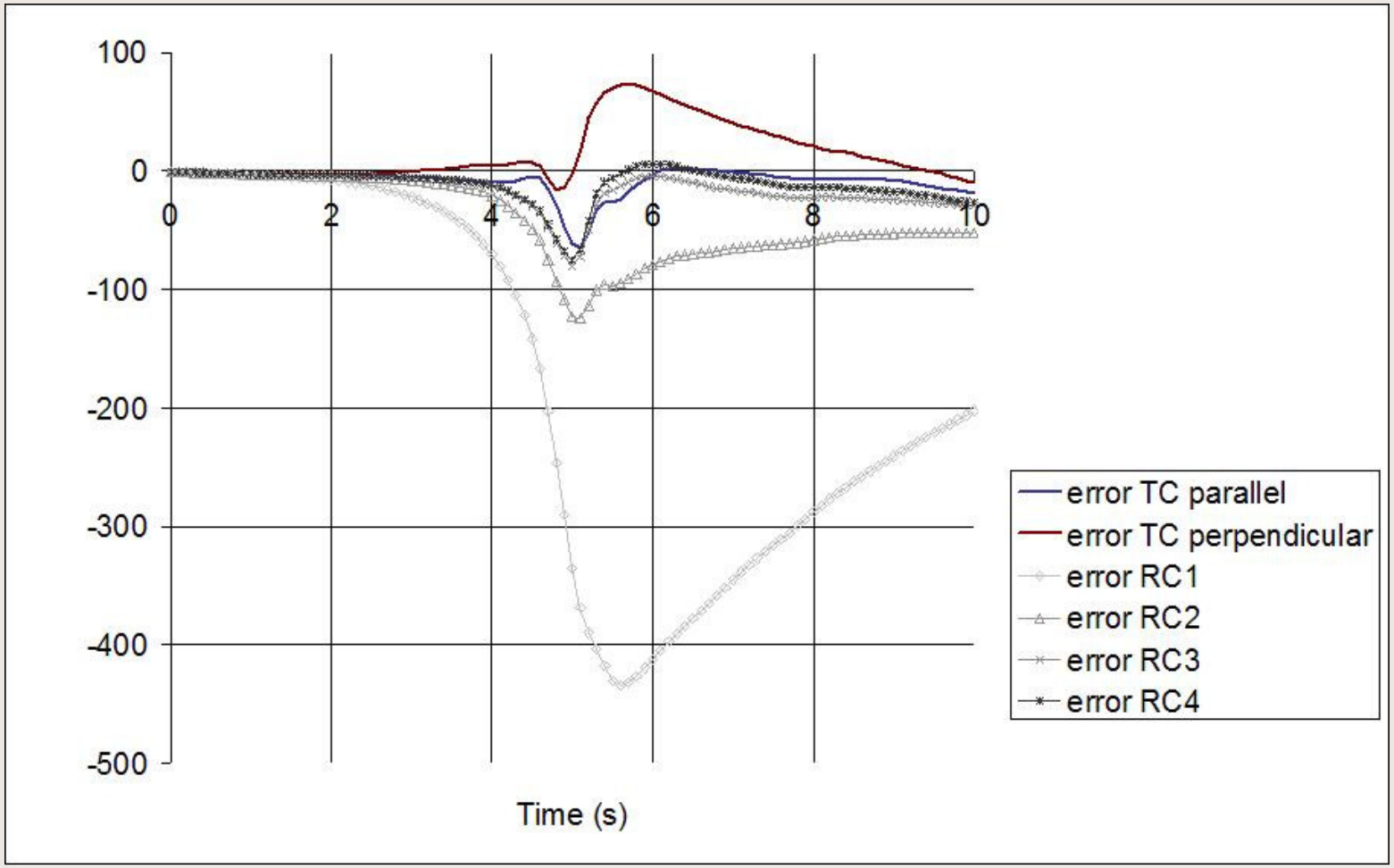


Modelisation of the welding problem with thermocouples

Visualisation of the measurements errors and comparisons between the two configurations



Modelisation of the welding problem with thermocouples



Modelisation of the welding problem with thermocouples

Conclusions for the two configurations

1- With the thermocouples in an isotherm, we have less errors.

2- It's very important to have a good contact between the thermocouple and the material

3- We must define correctly the space domain to have the less errors.

The Levenberg Marquardt file

Open the optim_003_sans_TC.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% the estimated parameter is:
% p1     :    the gaussian amplitude ref = 4000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all; clear all; clc

global p1

epsv=1e-4;
% Definition of the measured points %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mx=[0 0.00634];
my=[0.05 0.05];
mz=[0.0035 0.008];
pp=[mx;my;mz];

% the theoretical value of Q=4000

P1=4000;

% Resolution of the direct problem

problem_direct_003_ss_TC

% the theoretical temperatures

TCme=postinterp(fem,'T',pp,'solnum','all');
for j=1:length(TCme(1,:))
    for i=1:length(TCme(:,1))
        TCmep(i+length(TCme(:,1))*(j-1))=TCme(i,j);
    end
end

TCmes=TCmep';
```

Change the direct problem: Introduce the name of the file for the problem with thermocouples

Modelisation of the welding problem with thermocouples

for 7 iterations	RC= 1e-3		RC= 1e-4		RC= 1e-5		RC= 1e-6		RC= 0	
TC perpendicular	2907	37,60%	3788	5,60%	3923	1,96%	4064	1,57%	4061	1,50%
TC parallel	2600	53,85%	3854	3,79%	3987	0,33%	4010	0,25%	3985	0,38%

Modelisation of the welding problem with thermocouples

Conclusions for the two configurations

1- An estimation which don't take into account the real instrumentation leads to an error.

2- This error can be higher if we have bigger thermocouples (here the diameter of the wire is 50µm). It's impossible to define the characteristic time for the thermocouple. In fact, we study the interaction between the thermocouple with the domain

3- At last, if we use thermocouples, we must analyze the transfer between the thermocouple and the material (Rc and heat transfer coefficient between the wires and the environment). And, we must try to use a real experimental direct problem in the optimization loop. Or eventually, we must quantify the measurement corrections

Thanks for your attention

Thanks to Comsol support for their help

Have a nice METTI School