

# Low Cost Welding Simulation

## *Crash course- CalculiX Part I*

Dr.-Ing. Ossama Dreibati

[www.ing-ask.de](http://www.ing-ask.de)

Emmendingen, October 2016

# Introduction

In this part I will discuss the welding simulation with the open source application CalculiX:

- Weld simulation with open source codes
  - Structure of the input file
- Uncoupled thermomechanical calculation
  - Preprocessing with open source code
    - Definition of the heat source
    - Evaluation of the results
  - Sequential thermomechanical analysis
- Weld deposit / Element activation and deactivation

*....enjoy it*

# Low Cost Welding Simulation with open source codes

**Leading question may be put: is it possible to perform welding simulation with open source codes?**

The answer of this question is not trivial. Open source programs are generally “community” software which are created by active and creative persons for free. The participants add enhancements to the code regarding to their skills in the domain, and the community validates the code with other commercial systems. Anyway, supports from deferent sides boost the development of the code. The physical rules behind these codes are in general the same as the professional codes.

The code is available for free, and can be user-oriented improved to cover special cases of the handled problem. Open source codes find their major application in the F&E fields.

The main difficulty by the use of open source codes in industrial scale arise due the high customization time and the integration of the code in the development chain where other programs and different approaches are involved.

For the free codes there is no support or service to help in the integration. “You have to do all things by yourself”

# Low Cost Welding Simulation with open source codes

The open source code could open a new proportionality to get low cost simulation. For example, for those who have to use welding simulation as a stand alone process. All what they have to do is the first customization and the rest will be a standard work. The same is for the integration with other systems.

We will consider only the welding structure simulation as well as metal simulation

Also,

- Calculation of distortion and residual stresses
- Strength analysis
- Metallurgy calculation
- Crack susceptibility (hot and cold cracks)

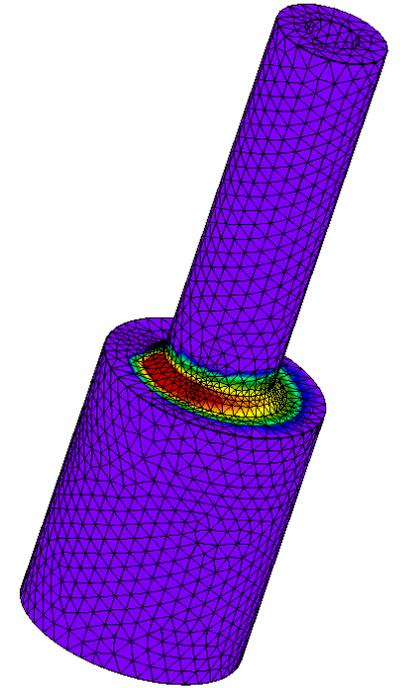
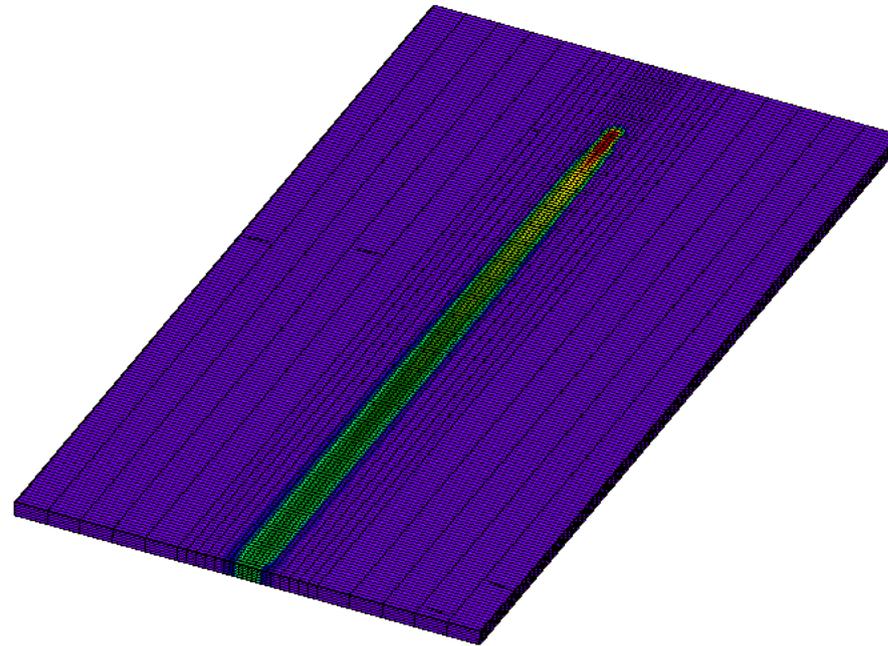
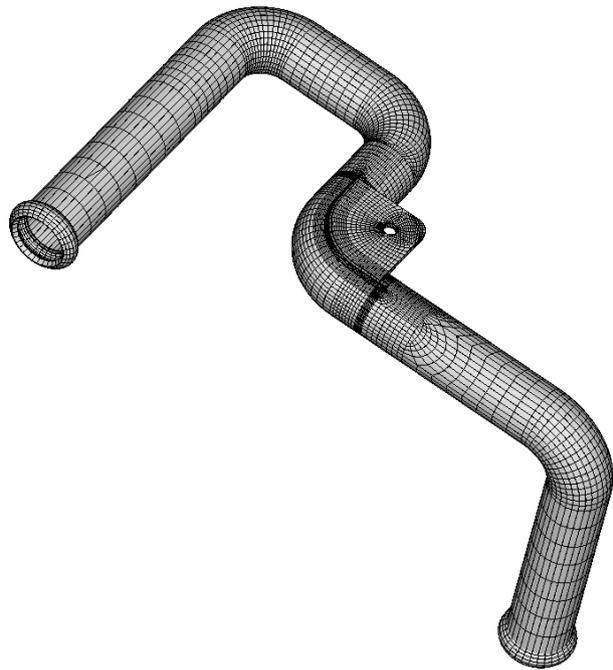
In this part, the calculation of distortion and residual stresses are considered

# Welding Simulation with open source codes

Examples:

Laser beam welding

Arc welding



Eight-node shell element  
S8

Eight-node brick elements  
C3D8

Four-node tetrahedral element  
C3D4

Modeling is done with Salome and the simulation with Calculix 2.11

# The free and open source finite element analysis application Calculix

Calculix is a free and open source finite element analysis application. Originally it was written for Linux but meanwhile there is a Windows version. It has an implicit and explicit solver. Calculix is written by Guido Dhondt and a pre and post processor (CGX) is written by Klaus Wittig.

With CalculiX Finite Element Models can be build, calculated and post-processed. The solver is able to do linear and non-linear calculations. Static, dynamic and thermal solutions are available, and it makes use of the abaqus input format.

The CalculiX is available in <http://www.calculix.de/>

A mailing list with name "calculix" has been setup. By this way, you can share problems and experiences with other CalculiX users. To join the group go to [groups.yahoo.com/group/calculix](http://groups.yahoo.com/group/calculix).

At the moment, the version 2.11 of Calculix is available.

# Thermomechanical Simulation with Calculix

In Calculix there are two ways to solve thermomechanical problems such as welding.

## \*COUPLED TEMPERATURE-DISPLACEMENT

The displacements and temperatures are simultaneously solved. The temperature influences the displacements through the thermal expansion, and the displacement effect on the temperature is limited to radiation.

## \*UNCOUPLED TEMPERATURE-DISPLACEMENT

A thermal analysis is performed first for each increment, and the resulting temperature field is applied as boundary load in the subsequent mechanical analysis for the same increment. Thereby, no feedback from mechanical analysis in the thermal one is performed.

Coupled analysis is very time-consuming comparing with uncoupled one.  
Uncoupled Analysis is mostly suitable for welding simulation

Free solvers available in Calculix

- ◆ Iterative solver (CHOLESKY or SCALING)
- ◆ SPOOLES

Spooles is the default solver and it is fast for the thermomechanical analysis. It doesn't support integer 8 and therefore, the the system size one can solve is limited.

# Structure of the input file

Calculix uses a similar input format to Abaqus. Typical file structure for welding simulation

## **\*\*Preprocessing**

\*HEADING ← start

\*NODE, NSET=Nname ← Definition of nodes and elements

\*ELEMENT, TYPE=type, ELSET=Ename ← Definition of nodes and elements

\*elset, elset=element set name ← Definition of node and element sets to arrange the properties and/or for post processing

\*nset, nset=node set name ← Definition of node and element sets to arrange the properties and/or for post processing

\*MATERIAL, NAME=material name ← Material definition

**\*\*data for mechanical analysis depends on the chosen material law**

    \*ELASTIC  
    E-modulus, poisson ratio, temperature1  
    . . .further values for next temperatures

    \*plastic, hardening=isotropic or kinematic  
    Stress (plastic), plastic strain1, temperature1  
    . . .further values for next plastic strains and temperatures

    \*EXPANSION  
    Expansion coefficient, temperature1  
    ...further values for next temperatures

**\*\*data for thermal analysis**

    \*CONDUCTIVITY  
    conductivity, temperature1  
    ...further values for next temperatures

    \*SPECIFIC HEAT  
    Specific heat, temperature1  
    ...further values for next temperatures

    \*DENSITY  
    density, temperature1  
    ...further values for next temperatures

This part must follow the keyword \*MATERIAL and must be given for every material

# Structure of the input file

- \*SOLID SECTION, MATERIAL= material name, ELSET= element set name  Assign of material to elements
- \*\* the assignment of materials to elements must be done for all elements (every element must be arranged to one material)
- \*INITIAL CONDITIONS, TYPE=TEMPERATURE  Initial conditions such as temperatures, pre-stresses or -stains  
Nname, 20.
- \*PHYSICAL CONSTANTS, ABSOLUTE ZERO=-273.15, STEFAN BOLTZMANN=5.669E-11  Properties for radiation
- \*\*please take care about the units in the whole analysis. The units must be consistent!
- \*BOUNDARY  Boundary condition, e.g. clamping  
Node or node set, first dof, last dof, displacement value  dof for x-axis =1, y-axis=2, z-axis= 3
- \*\*----- Calculation**
- \*STEP, INC=increment number  Max number of increments e.g. 1000
- \*UNCOUPLED TEMPERATURE-DISPLACEMENT  Keyword for uncoupled thermomechanical analysis  
Initial time increment, total step time, minimum time increment allowed, maximum time increment allowed
- \*\*
- \*FILM, OP=NEW  Convection through the surfaces  
Element number or element set name, film flux type label, sink temperature, film coefficient
- \*RADIATE, OP=NEW  Radiation through the surfaces  
Element number or element set name, radiation flux type label, sink temperature, emissivity.
- \*DFLUX, OP=NEW  Definition of the distributed heat input  
Element set name or element number, BFNU,1  BF for body flux and NU for nonuniform flux and user routine "dflux" must be provided
- \*\* Result for postprocessing**
- \*NODE PRINT, NSET=Nname  Save the nodal results in a DAT file,  
NT,U  NT = node temperature, U = displacement in x,y and z direction
- \*NODE FILE  Save the nodal results in a frd file,  
NT,U  NT = node temperature, U = displacement in x,y and z direction
- \*Element Output  Save the element results in a frd file,  
HFL,S  HFL = heat flux, S = stresses in x,y and z direction sowie von mieses and others
- \*END STEP
- \*\*step numbers are unlimited

# Structure of the input file

\*\*-----step 2 cooling stage  
\*STEP, INC=increment number ← Max number of increments e.g. 1000  
\*UNCOUPLED TEMPERATURE-DISPLACEMENT ← Keyword for uncoupled thermomechanical analysis  
Initial time increment, total step time, minimum time increment allowed, maximum time increment allowed  
\*\*  
\*FILM, OP=NEW ← Convection through the surfaces  
Element number or element set name. film flux type label. sink temperature. film coefficient  
\*RADIATE, OP=NEW ← Radiation through the surfaces  
Element number or element set name. radiation flux type label. sink temperature. emissivity.  
\*DFLUX, OP=NEW ← Delete the distributed heat input  
\*\*  
\*\*  
\*NODE PRINT, NSET=Nname ← Save the nodal results in a DAT file,  
NT,U NT = node temperature, U = displacement in x,y and z direction  
NT,U Save the nodal results in a frd file,  
\*NODE FILE ← NT = node temperature, U = displacement in x,y and z direction  
NT,U  
\*Element Output ← Save the element results in a frd file,  
HFL,S HFL = heat flux, S = stresses in x,y and z direction sowie von mieses and others  
\*END STEP

# Structure of the input file

\*\*-----last step (unclamping)

\*STEP, INC=increment number

\*UNCOUPLED TEMPERATURE-DISPLACEMENT

Initial time increment, total step time, minimum time increment allowed, maximum time increment allowed

\*\*

\*FILM, OP=NEW ← Convection through the surfaces

Element number or element set name, film flux type label, sink temperature, film coefficient

\*RADIATE, OP=NEW ← Radiation through the surfaces

Element number or element set name, radiation flux type label, sink temperature, emissivity.

\*\*

\*BOUNDARY, OP=NEW ← Delete the previous boundary conditions and redefine new one to ensure the equilibrium of the part.

\*\*Rigid body motion

Node 1, rigid in x-, y- and z-direction ↑

Node 2, rigid in y- and z-direction ↓

Node 3, rigid in z-direction

Rigid body motion

\*\*

\*NODE PRINT, NSET=Nname

NT,U

\*NODE FILE

NT,U

\*Element Output

HFL,S

\*END STEP

# Preprocessing

In the following, the procedure to get the main data for the input file is described.

## Modeling

To generate the mesh of the considered problem, the free ware SALOME can be used. The CAD file in STEP format can be imported therein (take care about the units!). The geometry can be modeled as well as in salome using the direct modeling tools available in it. The mesh can be then exported in unv format, which can be converted to Calculix format using the free ware unical.

Alternatively, the mesh can exported in ASCII format, which contain the node coordinate and the element definition. The ASCII file can be manually reformatted to a inp format in a text editor or in Excel, LibreOffice Calc, etc.!. These free wares require a high skill level to get a good meshed model. **Since the analysis results depends strongly on the mesh quality, it is highly recommended to check the mesh quality before analysis start!** Because the solver makes use of the ABAQUS input format it is possible to use commercial pre-processors as well.

When the mesh is exported to a unv format such as Mesh\_1.unv, the converting into inp format can be done in a terminal as follow

*Unical Mesh\_1*

Then the file Mesh\_1.inp will be generated

SALOME is available in <http://www.salome-platform.org/>

Unical is free tool available in the forum of the salome

# Preprocessing

## Organization of the mesh data

To have an organized input file, I recommend to make a structured input file!

Firstly, load the input file in the post-processor cgx. This is done via the command (in a terminal opened in the working directory)

```
cgx -c Mesh_1.inp
```

Then via the commands

```
send all abq
```

The file all.msh, which contains the node and element definition, is generated. Export all the groups, which is made originally in SALOME, such as clamping nodes, Parts elements, weld elements, evaluation sections etc.

```
send group abq nam (with the keyword nam, only the elements number or nodes number will be exported)
```

A new organized input file can be then generated in a text editor e.g. input\_file.inp

```
*HEADING
```

```
*INCLUDE,INPUT=all.msh
```

```
*INCLUDE,INPUT=basis.nam
```

```
*INCLUDE,INPUT=weld.nam
```

```
*INCLUDE,INPUT=clamp.nam
```

```
....
```

# Preprocessing

## Material definition

In Calculix there is no material model deals with the metallurgy. Furthermore, the influence of maximum temperatures, cooling rate or heating rate on the thermomechanical properties is not included. BUT Calculix offers a user routines which can be redefined to deal with arbitrary material behavior.

I recommend to write the material data in a separate files such as basis.mat, weld.mat etc. The content of the material definition is given previously. Also each material file must have

```
*MATERIAL, NAME=material_name_1
*ELASTIC
    E-modulus, poisson ratio, temperature1
    ...further values for next temperatures
*plastic, hardening=isotropic or kinematic
    Stress (plastic), plastic strain1, temperature1
    ...further values for next plastic strains and temperatures
*EXPANSION
    Expansion coefficient, temperature1
    ...further values for next temperatures
    **data for thermal analysis
*CONDUCTIVITY
    conductivity, temperature1
    ...further values for next temperatures
*SPECIFIC HEAT
    Specific heat, temperature1
    ...further values for next temperatures
*DENSITY
    density, temperature1
    ...further values for next temperatures
```

## Material 1

```
*MATERIAL, NAME=material_name_2
*ELASTIC
    E-modulus, poisson ratio, temperature1
    ...further values for next temperatures
*plastic, hardening=isotropic or kinematic
    Stress (plastic), plastic strain1, temperature1
    ...further values for next plastic strains and temperatures
*EXPANSION
    Expansion coefficient, temperature1
    ...further values for next temperatures
    **data for thermal analysis
*CONDUCTIVITY
    conductivity, temperature1
    ...further values for next temperatures
*SPECIFIC HEAT
    Specific heat, temperature1
    ...further values for next temperatures
*DENSITY
    density, temperature1
    ...further values for next temperatures
```

## Material 2

# Preprocessing

The arrangement of material to the different parts in the model takes place through the keyword *\*SOLID SECTION, MATERIAL= material name, ELSET= element set name*

## For example

*\*SOLID SECTION, MATERIAL= material\_name\_1, ELSET= Ebasis*

*\*SOLID SECTION, MATERIAL= material\_name\_2, ELSET= Eweld*

The initial conditions and parameters for radiation can be then defined

*\*INITIAL CONDITIONS, TYPE=TEMPERATURE*

*Nall, 20. (Nall is a group of all nodes in the model)*

*\*PHYSICAL CONSTANTS, ABSOLUTE ZERO=-273.15, STEFAN BOLTZMANN=5.669E-11*

With the mechanical boundary conditions, the preprocessing can be finished. For example

*\*BOUNDARY*

*Nclamp, 1, 3, 0*

*\*\*rigid clamping in all directions for all nodes of the nodes group Nclamp*

# Calculation

The calculation part start with the keyword \*STEP. For example

*\*STEP, INC=1000*

And the thermomechanical calculation is triggered with

*\*UNCOUPLED TEMPERATURE-DISPLACEMENT*

*Initial time icrement, total step time, minimum time increment allowed, maximum time increment allowed*

With uncoupled thermal and mechanical calculation, there is no feedback from mechanical sub-step to the thermal sub-step. After that the convection and radiation between the parts and the environment must be defined. This is done though

*\*FILM, OP=NEW*

*Element number or element set name, film flux type label, sink temperature, film coefficient*

*\*RADIATE, OP=NEW*

*Element number or element set name, radiation flux type label, sink temperature, emissivity.*

The film and radiation flux type label depend on the element definition and the element orientation in the model. In the following, the derivation of these data is discussed.

# Calculation

## Determination of the film and radiation flux type label

The input file must be loaded in the cgx using

```
cgx -c Mesh_1.inp
```

Then via the command (without Quotation marks)

```
send all abq film "sink temperature" "film coefficient"
```

```
send all abq rad "sink temperature" "radiation coefficient"
```

The file all.flm containing the element numbers, film flux type label, sink temperature and film coefficient and the file all.rad containing the element numbers, radiation flux type label, sink temperature and radiation coefficient are generated.

The content of the files all.flm and all.rad can be copied in the input file underneath the keywords *\*FILM,OP=NEW* and *\*RADIATE,OP=NEW*.

To keep the input file clear, I recommend to include the contents of these file in the input file instead of copying it to the input file

```
*FILM, OP=NEW
```

```
*INCLUDE,INPUT=all.flm
```

```
*RADIATE, OP=NEW
```

```
*INCLUDE,INPUT=all.rad
```

# Calculation

## Definition of the heat input

By welding the heat flux is inhomogeneous distributed. I'll consider in the following only the volume flux in bodies (energy per unit of volume per unit of time).

The heat flux generation is triggered through the command

*\*DFLUX, OP=NEW*

*Element set name or element number, BFNU,1*

*\*\*BF for body flux and NU for nonuniform flux, 1 scale factor*

The routine dflux.f must be provided. An example of a dflux routine is provided with the Calculix but it doesn't cover the welding processes. I'll give guidance in the following to write a user specific dflux routine for welding.

dflux routine is written in Fortran. By standard installation of Calculix under Linux, It is located in the /user/local/Calculix/ccx\_2.11/scr

When the routine dflux is written, a new ccx must be made. For Linux:

Change to the home directory "/user/local/Calculix/ccx\_2.11/scr" and type *make* in the terminal. Note that the new ccx must be then copied to the bin directory and the accesses permission must be changed through *chmod a+rx ccx*

# Calculation

## Structure of the dflux routine

```
! user subroutine dflux
  subroutine dflux(flux,sol,kstep,kinc,time,noel,npt,coords,
&  jltyp,temp,press,loadtype,area,vold,co,lakonl,konl,
&  ipompc,nodempc,coefmpc,nmpc,ikmpc,ilmpc,yscale,mi)
!
```

This routine will be activated for each integration point of all elements written underneath the keyword `*dflux` in the input file. Also, keep the elements number small to save calculation time. That means, only the elements which take part in the weld (weld pool and some adjacent elements) should be considered.

Global coordinates of the integration point are given through `coords(1...3)` and the calculated heat flux through `flux(1)`

Depending on the welding procedure, the heat flux distribution is defined. I'll consider herein only the Laser beam welding procedure.

# Calculation

For the heat flux function the local coordination of the integration point  $XX$ ,  $YY$  and  $ZZ$  as well as the touch angel  $AY$  must be defined!. To do this I recommend to define a trajectory as a group of nodes in the weld center /e.g. in the surface of the weld). These nodes must be arranged in sequence started in the welding start point and finished where the weld will finish.

Then the coordinates of the integration points must be transformed to the local coordinates ( $X0, Y0, Z0$ )

By start the local coordinate are the coordinate of node  $n1$

When the welding torch passes through node  $n2$  then the origin local coordinates is shifted to the coordinates of node  $n2$  and so on.

The welding direction is the local  $y$  axis

Global coordinates of the integration point

$XX = \text{coords}(1)$

$YY = \text{coords}(2)$

$ZZ = \text{coords}(3)$

The coordinates of trajectory nodes (origin of the local coordinate) are defined through

$X0 = \text{co}(1, \text{node number})$  ! in global  $x$  axis

$Y0 = \text{co}(2, \text{node number})$  ! in global  $y$  axis

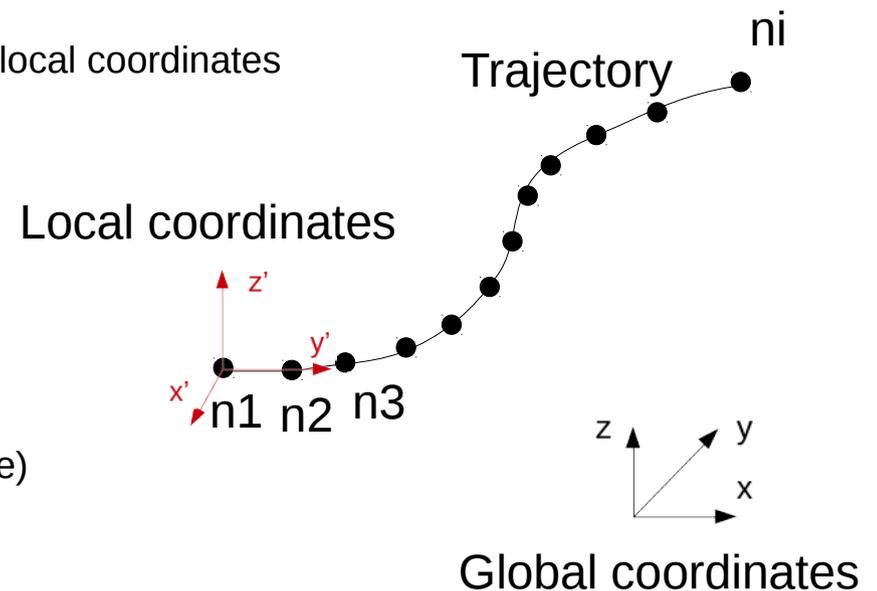
$Z0 = \text{co}(3, \text{node number})$  ! in global  $z$  axis

Transformation from global coordinate system to the local one (translation of the origin point)

$XX = XX - X0$

$YY = YY - Y0$

$ZZ = ZZ - Z0$



# Calculation

To get the coordinates of integration point oriented to the local coordinates they must be rotated about the angles of the line defined between two adjacent trajectory nodes (such as n1 and n2) and the global coordinates.

Rotation angle about z axis

$$\begin{aligned} \text{gama} &= \text{atan}(\frac{\text{co}(2, \text{node\_trj}(ii+1)) - \text{co}(2, \text{node\_trj}(ii))}{\text{co}(1, \text{node\_trj}(ii+1)) - \text{co}(1, \text{node\_trj}(ii))}) \\ \text{gama} &= \text{gama} - 1.570796327 \end{aligned}$$

Rotation angle about y axis (use this rotation about y axis only when you find it is necessary!)

$$\begin{aligned} \text{beta} &= \text{atan}(\frac{\text{co}(3, \text{node\_trj}(ii+1)) - \text{co}(3, \text{node\_trj}(ii))}{\text{co}(1, \text{node\_trj}(ii+1)) - \text{co}(1, \text{node\_trj}(ii))}) \\ \text{beta} &= \text{beta} - 1.570796327 \end{aligned}$$

Rotation angle about x axis

$$\begin{aligned} \text{alpha} &= \text{atan}(\frac{\text{co}(3, \text{node\_trj}(ii+1)) - \text{co}(3, \text{node\_trj}(ii))}{\text{co}(2, \text{node\_trj}(ii+1)) - \text{co}(2, \text{node\_trj}(ii))}) \\ \text{alpha} &= \text{alpha} - 1.570796327 \end{aligned}$$

Here ii take value from 1 to the maximum number of trajectory nodes -1.

!.....ROTATION ABOUT Z

$$\begin{aligned} X_ &= XX \\ Y_ &= YY \\ Z_ &= ZZ \\ XX &= (X_ * \cos(\text{gama})) + (Y_ * \sin(\text{gama})) \\ YY &= (-1 * X_ * \sin(\text{gama})) + (Y_ * \cos(\text{gama})) \\ ZZ &= Z_ \end{aligned}$$

And so on for other axes. In most cases, it is enough to rotate the coordinated about z and x axis!. Note that the list of the trajectory nodes must be given within the routine and don't use the read function from external file! due to the errors my occurs when using multi core simulation.

The location of the torch on the trajectory can be calculated by means of welding speed VY and global time (time(2)).

$YY = YY - (VY * \text{time}(2))$  this is given in the definition of heat source and not in the transformation!

# Calculation

When the global coordinate is rotated only about z and x axis, the angel AY of the torch in the xz plane can be given in radian when it is constant over the welding process. If it is not constant, I recommend to define a group of nodes at a reference line, which is parallel to the trajectory, and make pairs of nodes of the both trajectory and reference. Then calculate the angel between z axis and the line defined with each pair of nodes. The number of reference node must be equal to that one of trajectory. Using the arm function between two adjacent pairs, the angel can be calculated in each position.

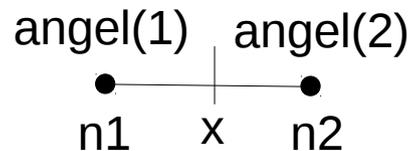
The angel for the first pair:

$$\text{angel}(1) = \frac{\text{co}(3, \text{node\_trj}(ii)) - \text{co}(3, \text{node\_ref}(ii))}{\text{co}(1, \text{node\_trj}(ii)) - \text{co}(1, \text{node\_ref}(ii))}$$

The angel for the second pair:

$$\text{angel}(2) = \frac{\text{co}(3, \text{node\_trj}(ii+1)) - \text{co}(3, \text{node\_ref}(ii+1))}{\text{co}(1, \text{node\_trj}(ii+1)) - \text{co}(1, \text{node\_ref}(ii+1))}$$

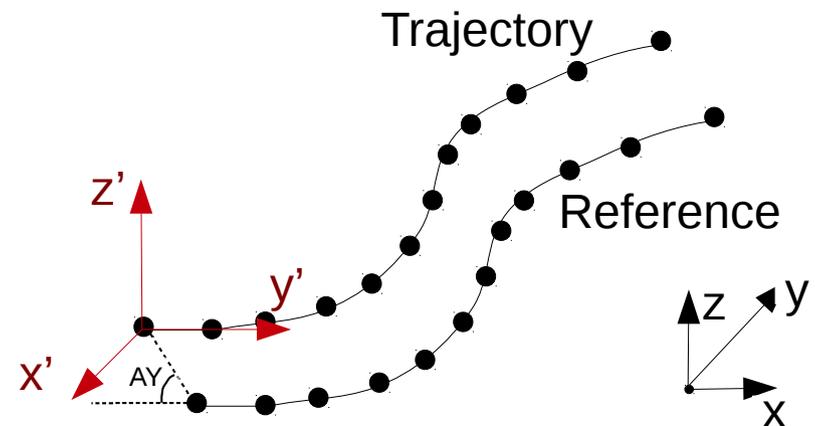
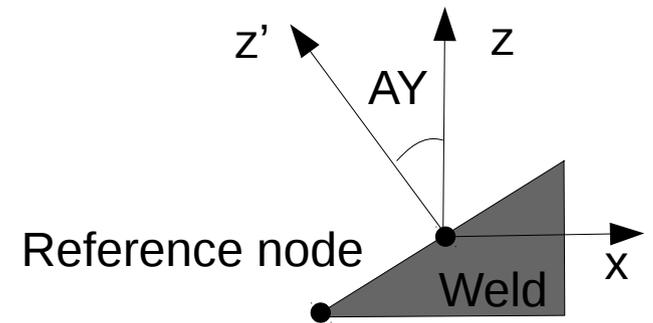
Here ii take value from 1 to the maximum number of trajectory nodes-1, node\_trj trajectory node and node\_ref reference node.



The angel in position x:

$$\text{angle} = \frac{\text{angel}(1) * (n2 - x) + \text{angel}(2) * (x - n1)}{(n2 - n1)}$$

Arithmetic distance



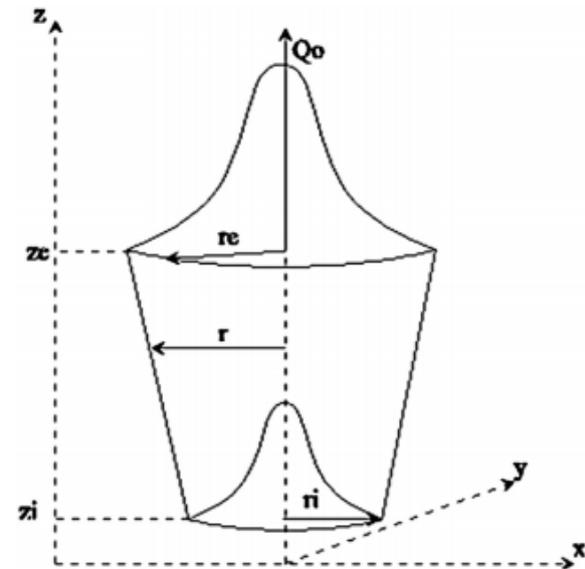
# Calculation

The heat flux of Laser beam welding is generally modeled by normal distributed function with reduced density in the depth to involve the keyhole effect in the model

```
!-----function
! FLUX = Q0 * exp( - R^2 / R0^2 ) where
! R^2 = ( XX-X0 )^2 + ( YY-Y0-VY*T )^2
! R0 = RE - ( RE-RI )*( ZE-ZZ+Z0 )/( ZE-ZI )
!-----
```

$Q0 = \text{power} * \text{effi} * 1000.0$  !input power depends on the units used  
 $RE = re$   
 $RI = ri$   
 $ZE = ze$   
 $ZI = zi$   
 $Xs = 0.0$   
 $Ys = 0.0$   
 $Zs = 0.0$   
 $VY = \text{speed}$   
 $AY = \text{angel (torch angel in radian)}$

Calibration parameter  
of heat source



3D Conical gaussian heat source model

# Calculation

! -----Shifting of the local coordinates  
! -----and rotation about y axis

$XD = XX - Xs$   
 $YD = VY * TT$   
 $YD = YD + Ys$   
 $ZD = ZZ - Zs$

!

$SA = \text{SIN}(AY)$   
 $SA = - SA$   
 $CA = \text{COS}(AY)$   
 $A1 = XD * CA$   
 $A2 = ZD * SA$   
 $XL = A1 + A2$   
 $YL = YY - YD$   
 $A1 = ZD * CA$   
 $A2 = XD * SA$   
 $ZL = A1 - A2$

Origin of y axis as  
a function of time

! ----- End shifting and rotation about y axis

! -----Start comutation

$DE = ZL - ZE$   
 $DI = ZL - ZI$   
 $\text{IF}( DE .GT. 0.00001 ) \text{ RETURN}$   
 $A1 = DI + RI$

!

$\text{IF}( A1 .LT. 0.00001 ) \text{ RETURN}$

! Continue

$A1 = XL * XL$   
 $A2 = YL * YL$   
 $R2 = A1 + A2$   
 $A3 = DI * DI$   
 $\text{IF}( ZL .LE. ZI ) R2 = R2 + A3 + A3$

!

$A1 = RE - RI$   
 $A2 = ZE - ZI$   
 $A3 = ZE - ZL$   
 $R0 = A3 / A2$   
 $R0 = R0 * A1$   
 $R0 = RE - R0$   
 $\text{IF}( ZL .LE. ZI ) R0 = RI$   
 $R02 = R0 * R0$

!

$\text{IF}( R2 .GT. R02 ) \text{ RETURN}$   
 $A1 = R2 / R02$   
 $A1 = M1 * A1$   
 $A1 = \text{EXP}( A1 )$

!!----Heat flux

$\text{FLUX}(1) = Q0 * A1$   
Return  
END

# Postprocessing and evaluation of the simulation results

Postprocessing take place in the cgx

Load the results saved in a frd file with all groups defined in the input file

*cgx\_results\_file.frd input\_file.inp*

Check the temperature field and compare the temperature cycles with temperature cycle measurements.

Therefore, make sure that the node location in the model corresponds to the location of the thermocouple location in the measurement.

The shape of the weld pool must be compared with micro graph of real welds. I recommend to define a section (group of elements) of the model correspondent to the section of the micro graph already in the preprocessing stage. Then it is available to plot the temperature field in the section at the moment when the section has got maximum temperature.

In the terminal:

*Plot e section\_name*

And then choose in the cgx window the time → Ntemp → Entity → T

To show the weld pool shape in the section type in the terminal

*Max 1500* (1500 is the melt temperature of steel, change corresponding the the material used)

*Min 20*

Then you get the weld pool shape in the section. The length of the weld pool can be got using the same way by plotting the hole model and check the length on the surface. The temperature cycles can be got from the result\_file.DAT.

The plotting of the temperature cycles can be done using tabulating programs such as Excel

# Postprocessing and evaluation of the simulation results

Postprocessing take place in the cgx

To plot the distortion and residual stresses in the model

*plot e all*

And then choose in the cgx window

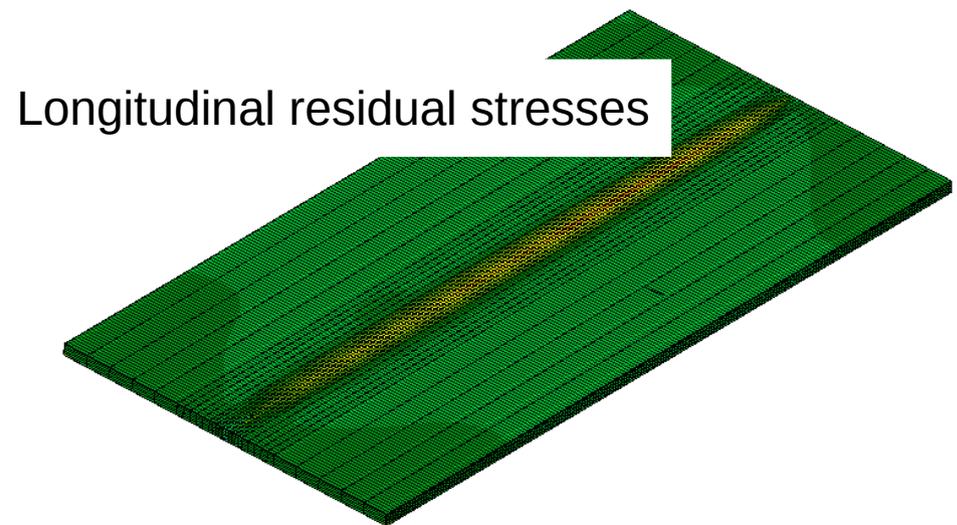
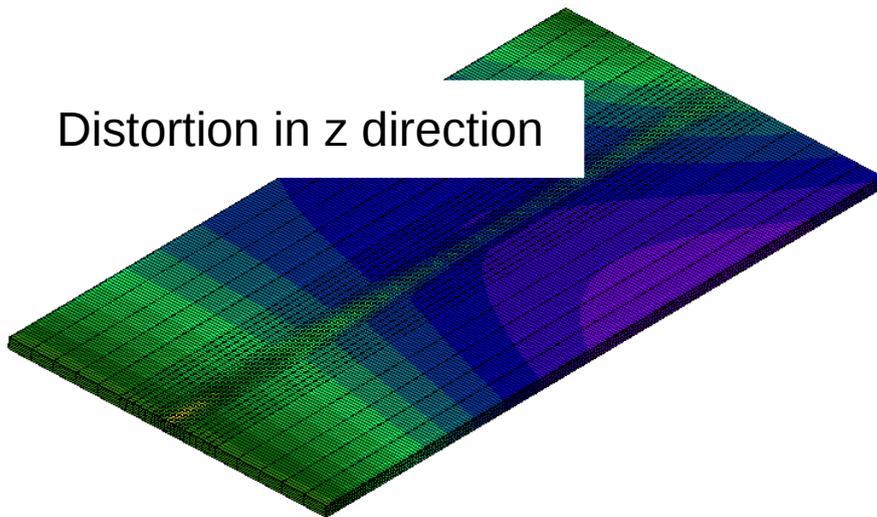
the last calculation step time → disp → Entity CD1,D2,D3 or all

For Stresses

Choose in the cgx window the

last calculation step time → STRESS → Entity → SXX,SYY,SZZ,Miseses etc.

For more details consult the cgx documentation



# Sequential thermomechanical analysis

For some reason may you wish to separate the thermomechanical analysis to a thermal analysis followed sequential with a mechanical calculation. To do this you have to perform a transient heat transfer analysis first. This is done by use \*HEAT TRANSFER instead of \*UNCOUPLED TEMPERATURE-DISPLACEMENT in the calculation

*\*STEP, INC=1000000*

*\*HEAT TRANSFER*

*Initial time step, Total time step, , Maximum time step*

\*\*keep the maximum time step small to get a fine resolved mechanical calculation when you start the mechanical analysis. The preprocessing remains the same as describe previously. Boundary conditions may be removed.

The nodes temperatures must be saved in a result\_file.DAT

*\*NODE PRINT, NSET=Na11*

*NT*

\*\*Na11 means all model modes

When the thermal analysis is finished and the mechanical calculation can be started.

Firstly, make a backup copy of the result\_file and rename it to Weld.dat

Secondly, change the Weld.dat file to get a file with following structure

\*\*

*\*STEP, INC=1000*

*\*STATIC*

*Initial time step, Total time step*

\*\*Total time step is the same time step from thermal calculation

*\*TEMPERATURE*

*Node number, temperature*

..

Thermal analysis



Mechanical analysis

Thermomechanical analysis

# Sequential thermomechanical analysis

I made a small Fortran program which prepares the file "Weld.dat" to the mechanical calculation, it looks like:

```
PROGRAM input_file
integer iso_12,iso_11,i,io_r,io_rr, stp, kt,node_f,f1
real*8 time_t,time,dtime,temp_t
character*51 textt,namee,filename
logical control_l,control_g
C
time = 0.0
stp = 0
f1=12
!=====
filename = 'Weld.dat'
! The nodal temperatures from the transient heat transfer analysis
! should be saved in the file
! Weld.dat. For this one have to use the
! *NODE PRINT, NSET=Nall
! NT
! in the thermal calculation and then rename the resulted file to "Weld.dat"
!
open(11,FILE=filename,IOSTAT=iso_11)
open(12,FILE='input.inp',IOSTAT=iso_12)
open(13,FILE='input2.inp',IOSTAT=iso_12)
if(iso_11 .GT. 0) then
write(*,*) 'error can not read the file'
go to 111
endif
!=====
do while(io_r .eq. 0)
read(11,'(A51)', IOSTAT=io_r) textt
if(textt .eq. ".") cycle
if(textt(2:14) .eq. 'temperatures') then
stp = stp + 1
if(stp.gt.200) f1=13
if(time .gt. 0.0) then
write(f1,"(A)") '*NODE OUTPUT'
write(f1,"(A)") 'NT,U'
write(f1,"(A)") '*Element Output'
write(f1,"(A)") 'S'
write(f1,"(A)") '*END STEP'
endif
read(textt(38:51),*) time_t
dtime = time_t - time
time = time_t
write(f1,"(A,I8)") '**STEP',stp
write(f1,"(A)") '*STEP, INC=1000'
write(f1,"(A)") '*STATIC'
write(f1,"(A,f12.3)") '** TOTAL TIME', time_t
write(f1,"(f12.3,A,f12.3)") dtime,',',dtime
if(stp.eq. 1) then
write(f1,"(A)") '*CONTROLS,PARAMETERS=TIME INCREMENTATION'
write(f1,"(A)") '10,20,21,'
write(f1,"(A)") '0.3'
write(f1,"(A)") '*CONTROLS,PARAMETERS=FIELD'
write(f1,"(A)") '5.,12.,0.01,,10,1.e-5,1.e-3,1.e-8'
end if
write(f1,"(A)") '*TEMPERATURE'
else
read(textt(2:11),*, IOSTAT=io_rr) node_f
read(textt(13:25),*, IOSTAT=io_rr) temp_t
write(f1,"(I8,A,f12.3)") node_f,',',temp_t
endif
end do
close(UNIT=11)
write(f1,"(A)") '*NODE OUTPUT'
write(f1,"(A)") 'NT,U'
write(f1,"(A)") '*Element Output'
write(f1,"(A)") 'S'
write(f1,"(A)") '*END STEP'
111 continue
close(UNIT=12)
close(UNIT=13)
END PROGRAM input_file
```

# Sequential thermomechanical analysis

Two files “input.inp” and “input2.inp” are generated. Two files to keep each file size small and you can use arbitrary file number. They contain all data required for the mechanical calculation and the output statements. To include these files in the mechanical analysis:

```
*INCLUDE, INPUT=input.inp  
*INCLUDE, INPUT=input2.inp  
**
```

The preprocessing remains the same as describe previously. Mechanical boundary conditions must be included, too. To unclamp the model we add the last step:

```
*STEP 1000  
*STATIC  
** TOTAL TIME 0.025  
1.0, 1.0  
*TEMPERATURE  
Nall,20.  
*BOUNDARY,OP=NEW  
Node 1,1,3,0  
Node 2,1,3,0  
Node 3,2,3,0  
*NODE OUTPUT  
NT,U  
*Element Output  
S  
*END STEP
```

Here node 1 through node 3 correspond the the three point principle

# Deposit material

For the weld simulation the geometry of all parts including weld geometry is modeled. By using of deposit material especially for multi pass welds there is a need to deactivate the elements not jet deposited. Calculix offers the possibility to change the material of a group during the calculation step. But this possibility is not suitable for welding, where the element of the deposit geometry only during cooling stage must be activated (also, with a time depending function).

To make time depending activation/deactivation of elements possible for thermal calculation, I recommend the following approach:

1- open the file "resultstherm.f" from /user/local/Calculix/ccx\_2.11/scr

2- search the line "call materialdata\_th(cocon,...)" and "call thermmodel(amat,...)" and insert the following code between them

```
!=====ELEMENT DEACTIVATION/ACTIVATION
  if(imat .eq.2) then
    fa=xstateini(1,kk,i)
    if(time.lt. Start time of pass 1) fa=0.000000000001
    if(fa.lt. 1.0)then
      if((ttime+time).gt. Start time of pass 2)then
        do j=1,3
          do i1=1,nope
            coords_(j,i1)=co(j,konl(i1))
          enddo
        enddo
        call weld_deposit(ttime+time,coords_,cogauss,co,nope,fa)
      end if
    end if
    do ii=1,6
      coconloc(ii)=coconloc(ii)*fa
    end do
    if(fa .lt. 1.d0) rho= 0.d0
    xstate(1,kk,i)=fa
  end if
!=====
```

# Deposit material

In the declaration add:

```
real*8 fa,coords_(3,20),cogauss(3)
```

! coords\_ the coordinates of the element nodes (max 20/element)

! cogauss(3) for the coordinates of integration point kk

The state of the fa is saved in the state variable 1. to make the state variable 1 active in the calculation use

```
*DEPVAR
```

```
1
```

After the material definition of the deposit weld in the input file.

When the element is deactivated then the conductivity is multiplied with the factor  $fa=0.000000000001$  and the density is zero. No heat can be generated in the deactivated elements.

The subroutine “weld\_deposit(ttime+time,coords\_,cogauss,co,nope,fa)” doesn’t exist yet and we have to great it.

Generally, the element must be initially deactivated and the activation function is similar to that one of welding.

```
subroutine weld_deposit(ttime,coords,cogauss,co,nope,fa)
```

```
... input here the same procedure to define the original of the local coordinate system as the routine dflux and the location of the torch
```

```
...in the example below the element in the cylinder defined with z axis and radius of 5 are activated
```

```
YY= YY -(VY*TT)
```

```
if((YY*YY + XX*XX).lt. 25.0) then
```

```
fa= 1.d0
```

```
jj=jj+1
```

```
end if
```

```
end do
```

```
if(jj .gt.3) then ! When more than 3 node of the element locate in the cylinder then the element is activated, exemplary!
```

```
fa= 1.d0
```

```
end if
```

```
RETURN
```

```
END
```

When this routine is finished then ccx has to be rebuild again with make etc. To include this routine in the make this routine must be included in the file Makefile.inc. This is done by adding

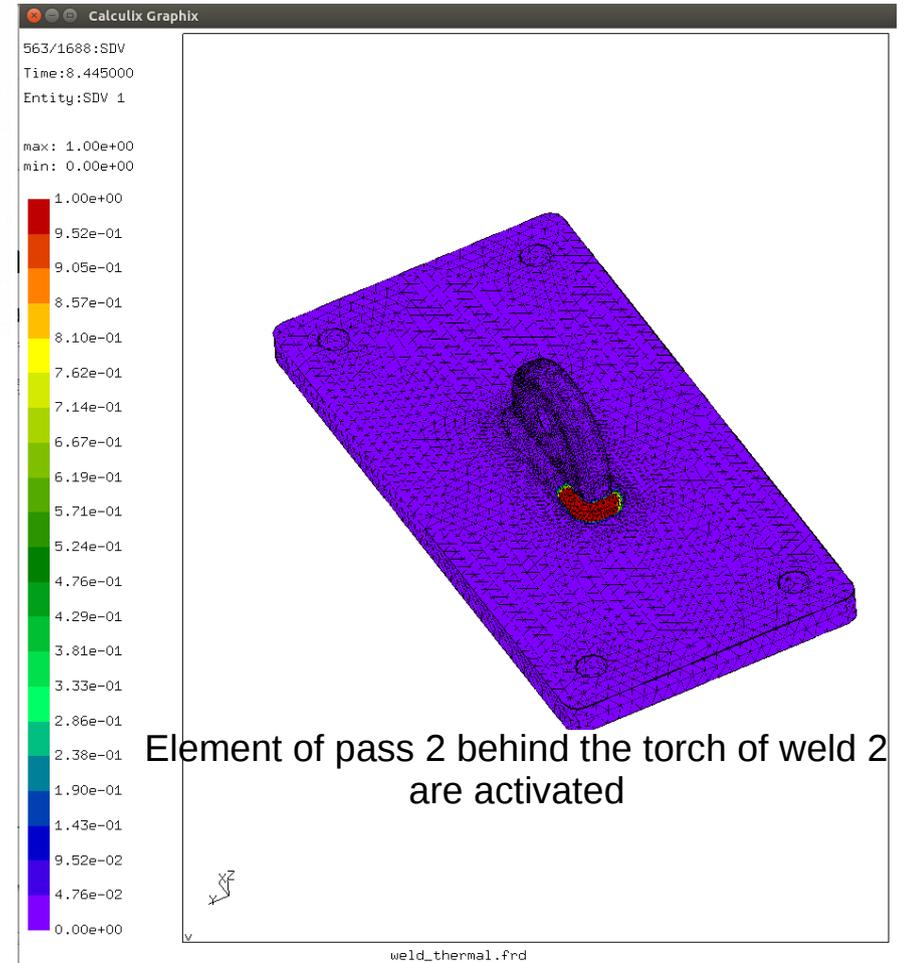
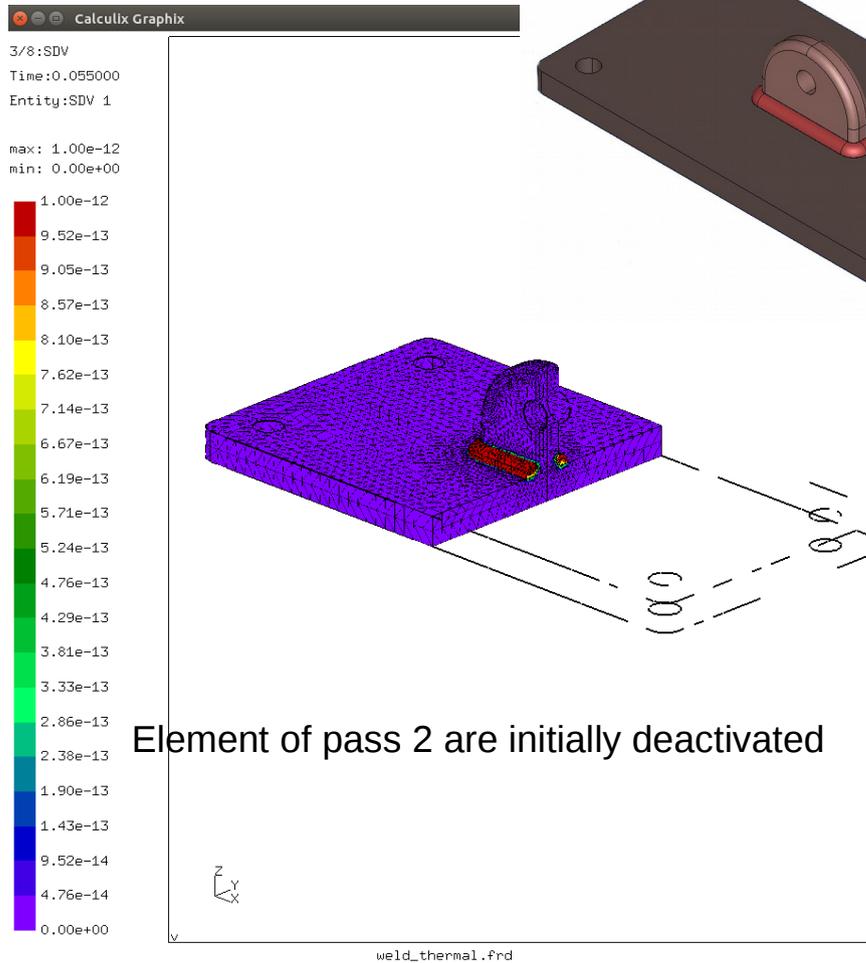
```
weld_deposit.f \
```

# Deposit material

Example:

Two pass weld of a ledge and plate

The second weld is deactivated initially. The reactivation take place time dependent



# Outlook

In **part II** I will discuss the user material and material laws. Also, tips and steps for the following themes:

- Influence of the peak temperature and cooling time on the mechanical calculation (user defined stress-strain curves and thermal strains)
- Writing a user material law for welding
- Calculation of the phase transformation (include the CCT diagrams)

***....Wish you an efficient simulation***