Documentation of welding process simulation code

J. Wijnen

Department of Engineering Science, University of Oxford

This documentation accompanies the welding process simulation code, consisting of ABAQUS user subroutines, of the pipeline structural integrity framework described in
"J. Wijnen, J. Parker, M. Gagliano, E. Martínez-Pañeda. A computational framework to predict weld integrity and microstructural heterogeneity: Application to hydrogen transmission. Materials & Design 2025."

In this documentation, only the welding process model is described. The model predicts the microstructural heterogeneity in terms of the phase fractions and resulting residual stresses. Two examples are included in this directory: i) A simple Jominy End Quench Test, where a small part of the bar is inserted replicating the deposition of a weld bead. ii) A welding simulation of a pipeline, as presented in the paper. Example i) is used throughout the document to explain how to setup a simulation.

The welding simulation is implemented sequentially. First a thermal-metallurgical simulation is performed to solve for the temperature field and accompanying phase transformations. Next, a separate simulation can be conducted which uses the results from the first simulation to solve for the deformation field, prediction residual stresses.

In addition, video tutorials on how to setup a weld simulation for a pipeline are available on Youtube:

- Tutorial on setting up the thermal-metallurgical part of the simulation:
  https://youtu.be/34Mb1INSRVc
- Tutorial on setting up the sequentially coupled residual stress simulation:
  https://youtu.be/Kw3ZCccEYM8

Finally, the results of the welding model can be used as an initial condition in a elasto-plastic phase field fracture simulation, which can optionally be coupled to hydrogen diffusion. This code, together with a documentation, can also be obtained from https://mechmat.web.ox.ac.uk/codes.

1 Software requirements

This documentation assumes that ABAQUS is setup properly such that the `abaqus` command can be used in the Command Prompt or Terminal. Additionally, the intel fortran compiler needs to be setup to be able to run user-subroutines.

Some python pre-/postprocessing scripts are provided that make use of the ABAQUS python scripting language. They can be run with the `abaqus python` command.

ABAQUS version 2022 is used in the accompanying .cae files, and can only be opened with this or later version. Nevertheless, the code itself should be able to run with any ABAQUS version, but this is not tested.

2 Thermal-metallurgical model

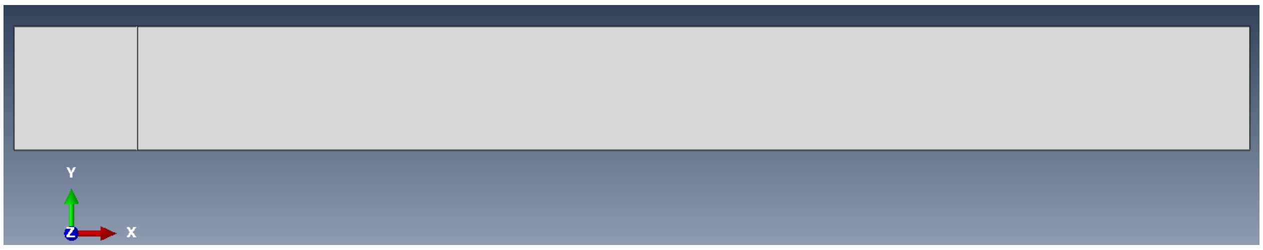
2.1 Abaqus CAE setup

Large part of the simulation can be setup in the ABAQUS CAE environment. This includes the geometry and the thermal model and its parameters. The Jominy bar is used as example. The most important steps in the setup are explained below.

2.1.1 Geometry, assembly and mesh

A 2D geometry is setup in the ABAQUS CAE environment. Here, we created a bar with dimensions 50×5. It is partitioned into two regions of lengths 5 and 45. As usual with ABAQUS geometries, an instance of the part is created under `Assembly`. Here, the part is meshed with a global seed size of 1. The element type needs to be set:

```
Element Type > Family: Heat Transfer
```



2.1.2 Model attributes

The absolute zero temperature and Stefan-Boltzmann constant need to be set.

```
Right-click on the model in the tree > Edit Attributes..
  > Absolute zero temperature: -273 > Stefan-Boltzmann constant: 5.67e-14
```

2.1.3 Material

The assigned materials need to have the following material behaviors assigned:

- `General > Density: 1`.
  (The density is handled internally in the user subroutine by using the volumetric heat capacity ρc)
- `General > User Material`
  `User material type: Thermal`.
  No `Thermal Constants` have to be defined.
- `General > Depvar`.
  `Number of solution-dependent state variables: 12`.
- `Mechanical > Expansion`.
  `General > Use Defined Field`.

Multiple materials can be assigned to different sections of the geometry. This allows to define different chemical compositions and material properties per section. In this case, two materials are created named `Base` and `Weld` for the right and left parts of the geometry, respectively.

2.1.4 Steps

For the example considered here, 4 steps are created.

1. Initialization step
2. Heating step
3. Insert step
4. Cooling step

`Step-1 Initialization, Heat transfer step, time period=1e-7, increment size=1e-7`
This step is always required. During this step, the subroutine initializes some of its state variables (depending on some initial fields). No boundary conditions are applied. It can be a small time period, using a single increment.

`Step-2 Heating, Heat transfer step, time period=3`
In this step the temperature on the edge between the base metal and weld metal is increased. Make sure to select *"Ramp linearly over step"* in the *"Other"* tab.

`Step-3 Insert, Heat transfer step, time period=1e-7, increment size=1e-7`
This step is needed if weld metal needs to be inserted. In this step the state variables in the inserted region are initialized. This step can again be a small time period, using a single increment.

`Step-4 Cooling, Heat transfer step, time period=120`
In this step we let the weld cooldown over given time period. An small initial time increment is usually needed

since temperature decreases fast at the beginning of this step. It is best to use automatic time-stepping to slowly increase the time increment in this step.

2.1.5 Field output request

Select `NT (nodal temperature)`, `{SDV (solution dependent state variables)`, and `FV (Predefined field variab` to write relevant quantities to the output file. `FV` is required as input for the sequential mechanical simulation. To be able to read these values in the mechanical simulation, output needs to be written at nodal locations

```
Element output position: Averaged at nodes
```

2.1.6 Interactions

Create a *Model change* interaction in *Step-1*.

```
Create Interatactation > Step: Step-1 > Types for Selected Step: Model change
```

```
> Region: select weld metal > Activation state of region elements: Deactivated in this step
```

Once created, under `States` in the model tree, edit the interaction in `Step-3`:

```
Activation state of region elements: Reactivated in this step
```

Alternatively, the interaction manager can be used to (de-)activate interaction in certain steps.

To model heat loss to the environment, a `Surface film condition` interaction and a `Surface radiation` interaction are setup on the outer surface.

2.1.7 Boundary conditions

Create a temperature boundary condition in `Step-1`. Select the edge in between the weld and the base regions. Here, we apply a temperature of 1500 degrees. Make sure the `Amplitude > Ramp`. This is only possible if in the step settings *"Ramp linearly over step"* is selected.

We want to deactivate the boundary condition in the cooldown step. This can be done by right clicking on `Step-3` under `States` and select `Deactivate` or in the boundary condition manager.

2.1.8 Predefined fields

Here , some initial conditions are set. First, create 2 *Temperature* predefined fields in step *Initial*. Set the initial temperature in the weld region to 1500, and that in the base region to 20. To make sure that the nodes on the boundary between the two regions has an initial temperature of 20, the predefined field for the base region needs to be lower in the list than the predefined field for the weld region. This can be achieved by starting the name of the weld predefined field with the number 1 and the name of the base predefined field with the number 2.

Next, the model uses the following user defined fields:

```
! ** FIELDS
!  1: xf (phase fraction ferrite)
!  2: xp (phase fraction pearlite)
!  3: xb (phase fraction bainite)
!  4: xm (phase fraction martensite)
!  5: xa (phase fraction austenite)
!  6: Gsize [mum] (average grain size [mm])
!  7: Hardness (Vickers hardness)
```

To make sure that ABAQUS allocates memory for all the user defined fields, at least one `Predefined field` needs to be defined. The field variable number needs to be set to 7. Its initial magnitude doesn't and is not used. To set initial values of fields 1-7, `Predefined fields` can be defined. For these fields, the initial value is taken into account. If no initial phase fractions are defined or the sum of the defined phase fractions is lower than 1 this will be corrected by the austenite phase fraction, which is automatically set to $x_a = 1 - x_f - x_p - x_b - x_m$. Additionally, phase fractions x_f and x_p will be set to their equilibrium fractions if their initial value exceeds it.

Here we created the following four user defined fields, by selecting the `Field` type in the `Create Predefined Field` dialog. We assume a ferrtite-pearlite microstructure in the base metal.

- Ferrite fraction in the base material

```
Create predefined field > Step: Initial > Types for selected step: Field
> Select base region > Field variable number: 1 > Magnitude 1
```

- Pearlite fraction in the base material

```
Create predefined field > Step: Initial > Types for selected step: Field
> Select base region > Field variable number: 2 > Magnitude 1
```

- Initial grain size in the base material

```
Create predefined field > Step: Initial > Types for selected step: Field
> Select base region > Field variable number: 6 > Magnitude 20
```

- Field 7 to ensure fields all are allocated

```
Create predefined field > Step: Initial > Types for selected step: Field
> Select whole geometry > Field variable number: 7 > Magnitude 150
```

2.1.9 Analysis

Create a job. Here we named it thermal. To write the simulation input file of the job:

```
Right-click on the job in the tree > Write Input
```

2.2 Additional input files

2.2.1 set_parameters_thermal.f90

The chemical composition needs to be specified in a file *set_parameters_thermal.f90* in the working directory of the simulation. An example of *set_parameters_thermal.f90* is:

```
param%matnames = 'BASE'

param%C = 0.07
param%Si = 0.235
param%Mn = 1.26
param%P = 0.015
param%V = 0.002
param%Ti = 0.021
param%Cr = 0.015
param%Ni = 0.02
param%Mo = 0.002
param%As = 0.002
param%Al = 0.024
param%Gsize_min = 10.0
```

Here, the chemical composition of the material in the ABAQUS CAE model with name *BASE* is specified. If a chemical element is not defined, it is assumed to be 0.

The model automatically calculates phase transformation temperatures based on empirical formula's. However, they can also be specified, which overrides the automatically calculated temperatures:

```
! Transformation temperatures
param%Ae3 = 823.6
param%Ae1 = 706.9
param%Bs = 605.7
param%Ms = 418.0
```

If different materials are used, they can be specified as follows:

```
param%matnames(1) = 'BASE'
param%C(1) = 0.07

param%matnames(2) = 'WELD'
param%C(2) = 0.14
```

Here, the weld has a higher carbon content than the base metal. If no index is used, the parameter is assigned to all materials.

Finally, a properties file, with temperature and phase dependent density, conductivity, and heat capacity needs to be specified

```
param%propfile = 'path_to_directory\WeldModel\examples\1_jominy\properties.inp'
```

2.2.2 properties file

Three properties need to be specified in the properties file, namely, the density, the conductivity and the specific heat. An example of a density specification is

```
*density
2
0.0078 0.008 20.0
0.0072 0.007 1500.0
```

On the first line, the property is defined. The properties have keywords `*density`, `*conductivity`, and `*specificheat`. On the second line, the number of temperature data points is specified. The last rows specify the properties at temperature data points. The first column specifies the properties of ferrite, pearlite, bainite and martensite, which use the same properties. The second column specifies the austenite property. The final column specifies the temperature.

2.2.3 State variable output (Optional)

The quantities that are stored in the state variables have the following numbering

```
! ** STATE VARIABLES
!  1: xf (ferrite phase fraction)
!  2: xp
!  3: xb
!  4: xm
!  5: xa
!  6: Gsize (mean)
!  7: Gsize
!  8: nucf
!  9: nucp
! 10: nucb
! 11: dT_dt at 700 degrees
! 12: temp_max
```

By default, the results in the output file will be names `SDV1`, `SDV1`, etc... To use more descriptive names, the following lines can be inserted in the .inp file:

```
*Depvar
12,
1, Xf, XF
2, XP, XP
3, XB, XB
4, XM, XM
5, XA, XA
6, GSIZE, GSIZE
7, GGROW, GGROW
8, NUCF, NUCF
9, NUCP, NUCP
10, NUCB, NUCB
11, DT700, DT700
12, TMAX, TMAX
```

A python script `thermal_input.py` is provided which automatically replaces these lines in the input file.

```
abaqus python thermal_input.py thermal.inp
```

This creates a new file `thermal_mesh.inp` where the state variable names are inserted. Here, the file is called `thermal.inp`, but this is dependent on the Job name that is assigned.

2.3 Running the simulation

The simulation can be run with (`job` and `input` can be different)

```
abaqus job=thermal input=thermal_mesh.inp user=sim_thermal.f90 interactive
```

In the final result, the inserted weld region has a microstructure consisting of bainite, ferrite and pearlite, which can be seen in Figure 1.

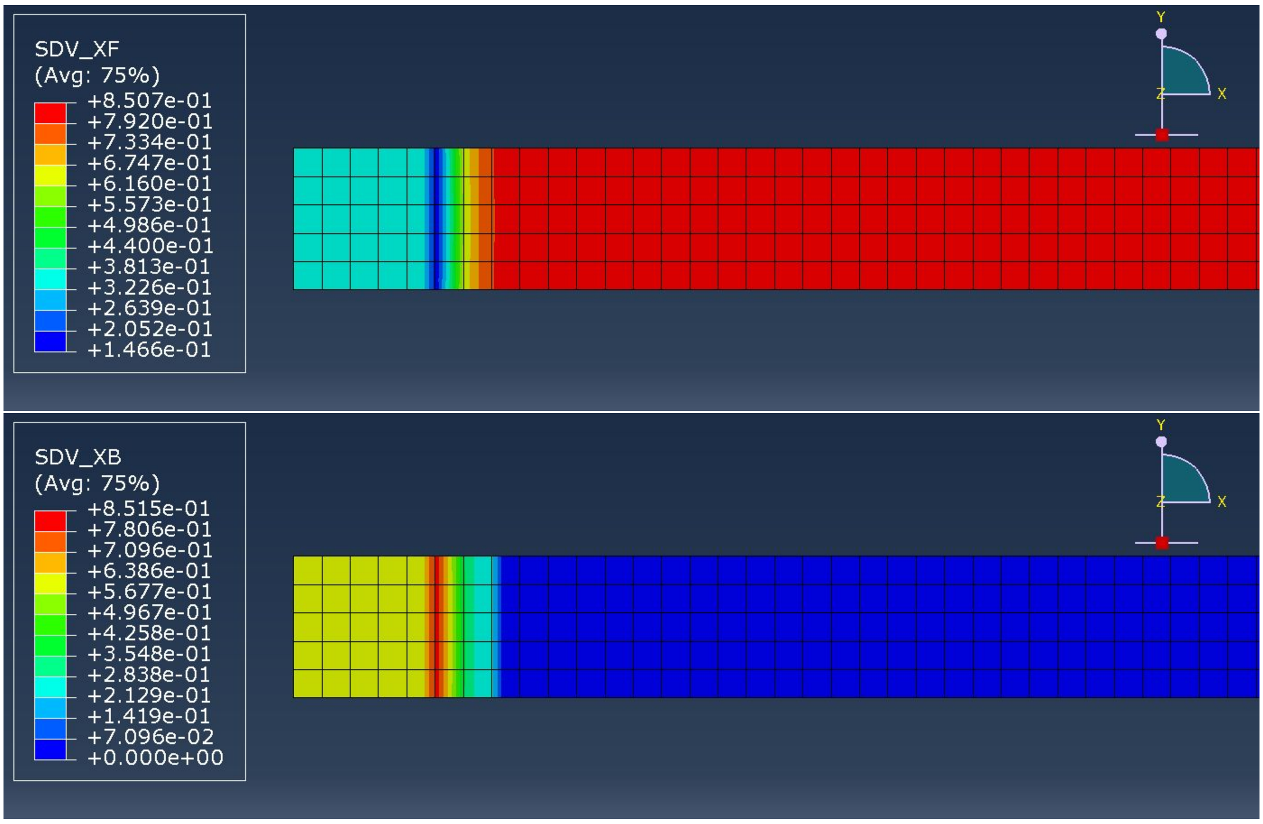


Figure 1: Results of the thermal model.

3 Mechanical model

The mechanical model uses the result obtained from the thermal-metallurgical model.

3.1 Abaqus CAE setup

3.1.1 Geometry, assembly and mesh

Probably the easiest way to start setting up the mechanical model is to copy the thermal model in the model tree. This ensures that the same geometry and mesh as in the thermal model is used. Note that this is not required. Data from the thermal model will be transferred to the mechanical model via predefined fields. ABAQUS has the option to use interpolation between different meshes. Nevertheless, we opt for the copying the previously created thermal model.

The element type needs to be changed

```
Element Type > Family: Plane Strain
```

3.1.2 Material

The assigned materials need to have the following material behaviors assigned:

- `General > User Material`.
  `User material type: Mechanical`.
  No `Mechanical Constants` have to be defined.
- `General > Depvar`.
  `Number of solution-dependent state variables: 30`.
- `Mechanical > Expansion`.
  `Type: Isotropic`.
  Check `Use user subroutine UEXPAN`.

3.1.3 Steps

The same steps as in the thermal model need to be created, but with

```
Procedure type: Static, General
```

The old heat transfer steps can be deleted. For the example considered here, the 4 steps were

1. Initialization step
2. Heating step
3. Insert step
4. Cooling step

The `Time Period` of the steps can be arbitrary, since no time-dependent phenomena are involved in this simulation. The Initialization and Insert step can again be done in a single increment.

3.1.4 Field output request

Relevant variables that can be written to the output file are `S (Stress)`, `E (Total strain)`, `U (Displacement)`, and `SDV (Solution dependent state variables)`.

3.1.5 Interactions

Similar as in the thermal simulations, create a *Model change* interaction in *Step-1*.

```
Create Interatction > Step: Step-1 > Types for Selected Step: Model change
> Region: select weld metal > Activation state of region elements: Deactivated in this step
```

Once created, under `States` in the model tree, edit the interaction in `Step-3`:

```
Activation state of region elements: Reactivated in this step
```

Alternatively, the interaction manager can be used to (de-)activate interaction in certain steps.

3.1.6 Boundary conditions

Here, we constrain (set to 0) the left and right sides of the geometry in x-direction (`U1`). A single point is constrained in the y-direction (`U2`) to suppress rigid body motion.

3.1.7 Predefined fields

Clear the fields that were defined in the thermal model.

The temperature and phase fraction fields of the thermal-metallurgical simulation need to be read in as `Predefined fields`. However, ABAQUS can only read in the nodal temperature from an ODB file. Therefore, the `thermal.odb` file will be split into multiple files where the phase fractions have the nodal temperature label. This is done in an ABAQUS python script *thermal_postproc.py*, and can be run by the following command

```
abaqus python thermal_postproc.py thermal.odb
```

This creates 10 additional files named `thermal\_i.odb` or `thermal\_i.prt` with i replaced by the phase identifier (f,p,b,m,a).

Next, the predefined temperature field is created:

```
Create Predefined Field > Step: Initial > Category: Other > Types for Selected Step: Temperature
Select entire geometry
Distribution: From results or output database file > File name: thermal.odb > Step: 1 > Increment: 0
```

To make sure all the steps are read correctly, expand `States` of the just created field in the model tree. Edit `Step-2`

```
Status: Modified > Begin step: 2
```

Repeat this for `Step 3` and `Step 4`, where `Begin step` needs to be set to the corresponding step.

The ferrite phase fraction fields also need to be read from an .odb file.

```
Create Predefined Field > Step: Initial > Category: Other > Types for Selected Step: Field
Select entire geometry
Distribution: From results or output database file > Field variable number: 1
> File name: thermal_f.odb > Output variable: NT > Step: 1 > Increment: 0
```

Similar to the temperature field, modify all steps to read the correct step number. Repeat above steps for all fractions, where the `File name` needs to be set to `thermal_p.odb`, `thermal_b.odb`, `thermal_m.odb` and `thermal_a.odb` for `Field variable number` 2, 3, 4, and 5, respectively.

3.1.8 Analysis

Create a job. Here we named it mechanical. To write the simulation input file of the job:

```
Right-click on the job in the tree > Write Input
```

3.2 Additional input files

3.2.1 set_parameters_mechanical.f90

A file with input parameters `set_parameters_mechanical.f90` needs to be created in the working directory of the simulation. An example of `set_parameters_mechanical.f90` is:

```
! Hardening exponent
param%n = 0.1

! Annealing temperature
param%T_anneal = 1300

! Absolute path to prop file
param%propfile = 'path_to_directory\WeldModel\examples\1_jominy\properties_mechanical.inp'
```

3.2.2 properties file

The temperature and phase dependent properties are specified in a separate properties file. Five properties need to be specified in the properties file, namely, the initial yield stress, the shear modulus, the bulk modulus, the density, and the trip parameters. An example of yield stress specification is

```
*yield
2
182 661 667 1428 207   10
 55 308 316  603  71 1500
```

On the first line, the property is defined. The properties have keywords `*yield`, `*shear`, `*bulk`, `*density`, and `*trip`. On the second line, the number of temperature data points is specified. The last rows specify the properties at temperature data points. The first column specifies the properties of ferrite, the second column of pearlite, the third of bainite, the fourth of martensite, and the fifth of austenite. The final column specifies the temperature.

3.2.3 State variable output (Optional)

The quantities that are stored in the state variables have the following numbering

```
! ** STATE VARIABLES
!  1:     THE   (thermal strain)
!  2:     PEEQ  (eq total plastic strain; plastic+TRIP)
!  3:     PPEEQ (eq plastic strain)
!  4:     TPEEQ (eq TRIP strain)
!  5-8:   PE    (total plastic strain)
!  9:12:  PPE   (plastic strain)
!  13:16: TPE   (TRIP strain)
!  17:20: EE    (elastic strain)
!  21:    SPD   (plastic dissipation)
!  22:    Y     (yield stress, including hardening)
!  23:    Y0    (initial yield stress)
!  24:    K     (bulk modulus)
!  25:    G     (shear modulus)
!  26:    RHO   (density)
!  27:    A     (TRIP parameter)
!  28:    RHOR  (reference density)
!  29:    SYF   (base factor for sy)
!  30:    PEEQT (plastic strain total, not annealed)
```

By default, the results in the output file will be names `SDV1`, `SDV1`, etc... To use more descriptive names, the following lines can be inserted in the .inp file:

```
*Depvar
30,
1, THE, THE
2, PEEQ, PEEQ
3, PPEEQ, PPEEQ
4, TPEEQ, TPEEQ
5, PE11, PE11
6, PE22, PE22
7, PE33, PE33
8, PE12, PE12
9, PPE11, PE11
10, PPE22, PE22
11, PPE33, PE33
12, PPE12, PE12
13, TPE11, TPE11
14, TPE22, TPE22
15, TPE33, TPE33
16, TPE12, TPE12
17, EE11, EE11
```

```
18, EE22, EE22
19, EE33, EE33
20, EE12, EE12
21, SPD, SPD
22, Y, Y
23, Y0, Y0
24, K, K
25, G, G
26, RHO, RHO
27, KTP, KTP
28, RHOR, RHOR
29, SYF, SYF
30, PEEQT, PEEQT
```

A python script `mechanical_input.py` is provided which automatically replaces these lines in the input file.

```
abaqus python mechanical_input.py mechanical.inp
```

This creates a new file `mechanical_mesh.inp` where the state variable names are inserted. Here, the file is called `mechanical.inp`, but this is dependent on the Job name that is assigned.

3.2.4 Running the simulation

To run the model, the `sim_mechanical.f90` file needs to be linked to abaqus. This can be done by running the following command:

```
abaqus job=mechanical input=mechanical_mesh.inp user=sim_mechanical.f90 interactive
```

Figure 2 shows the equivalent plastic strain and the residual stress in the x-direction.

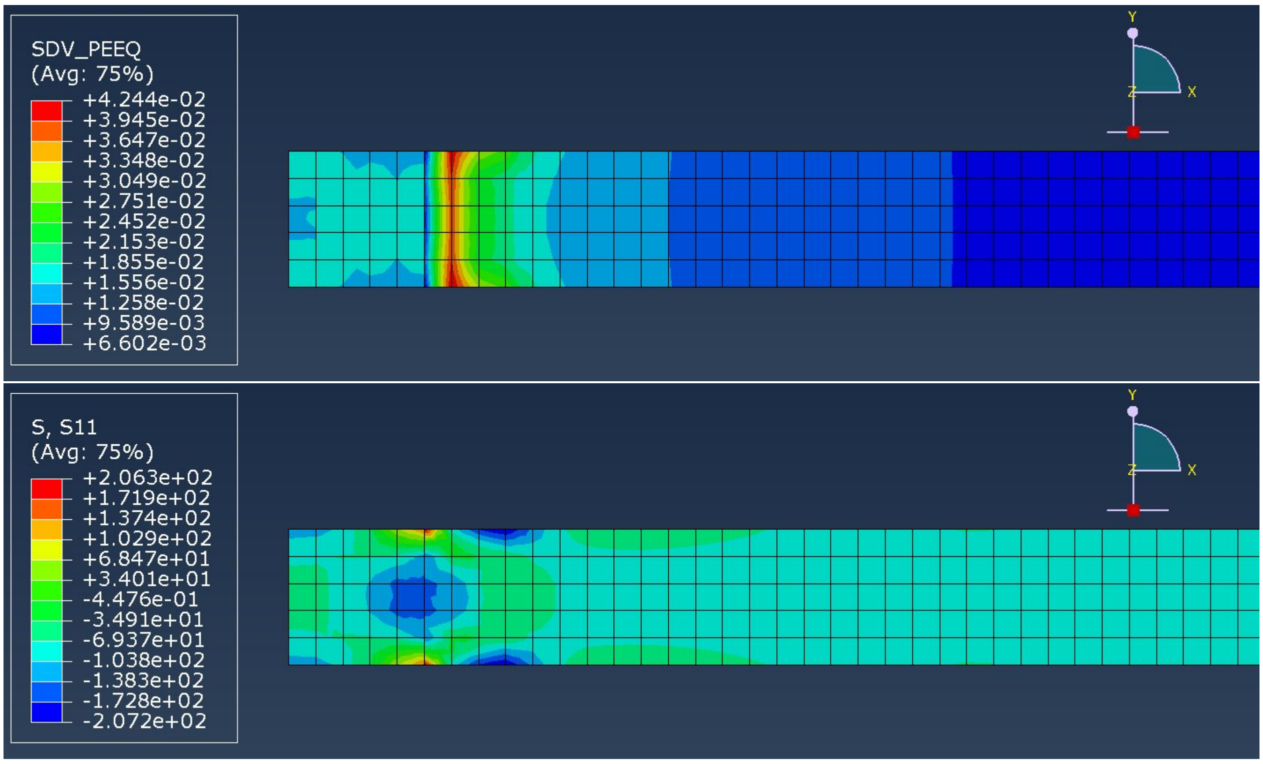


Figure 2: Results of the mechanical model.