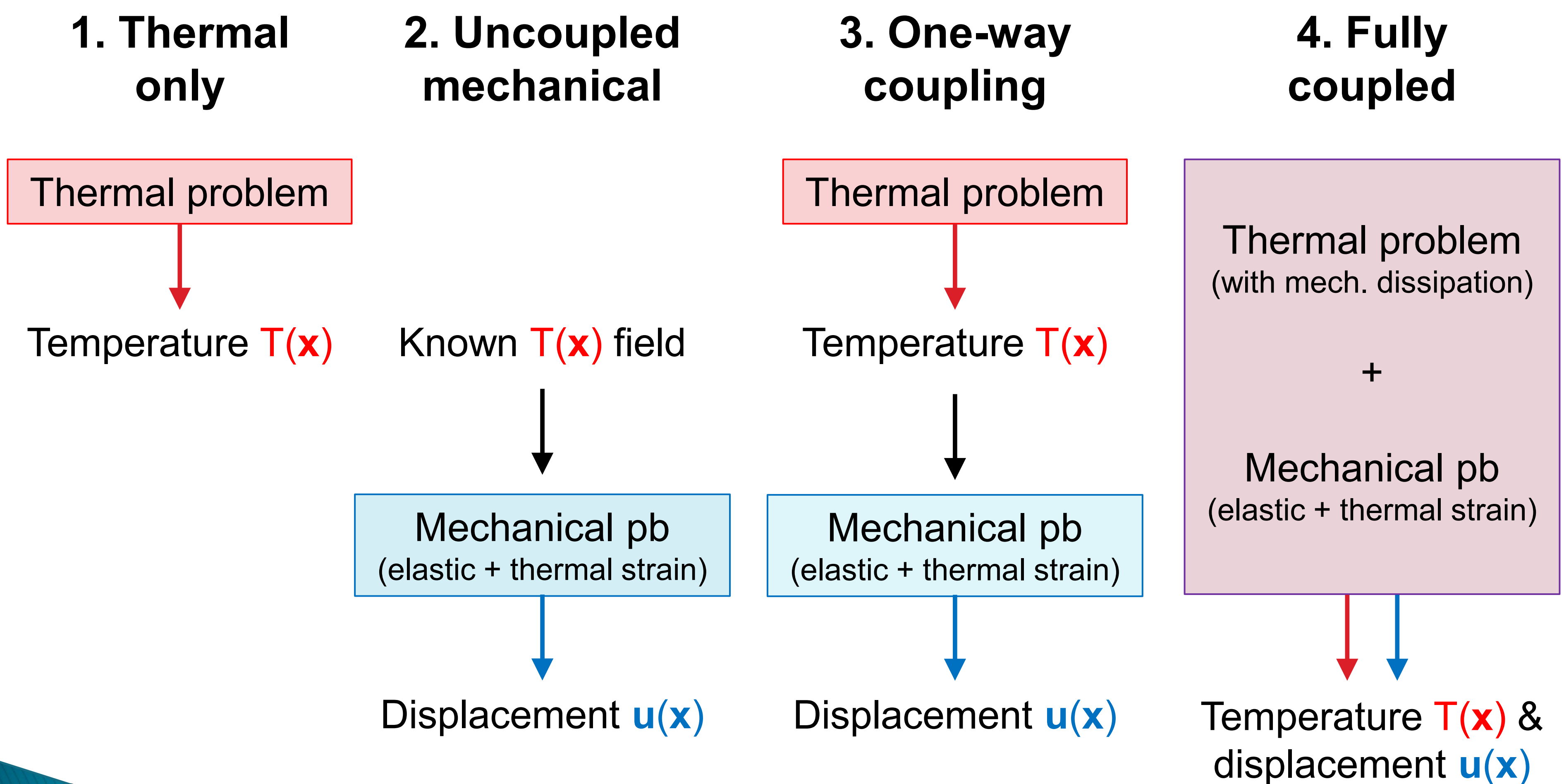
Thermo-mechanics

Modélisation et simulation par éléments finis

Edouard Boujo
Printemps 2024

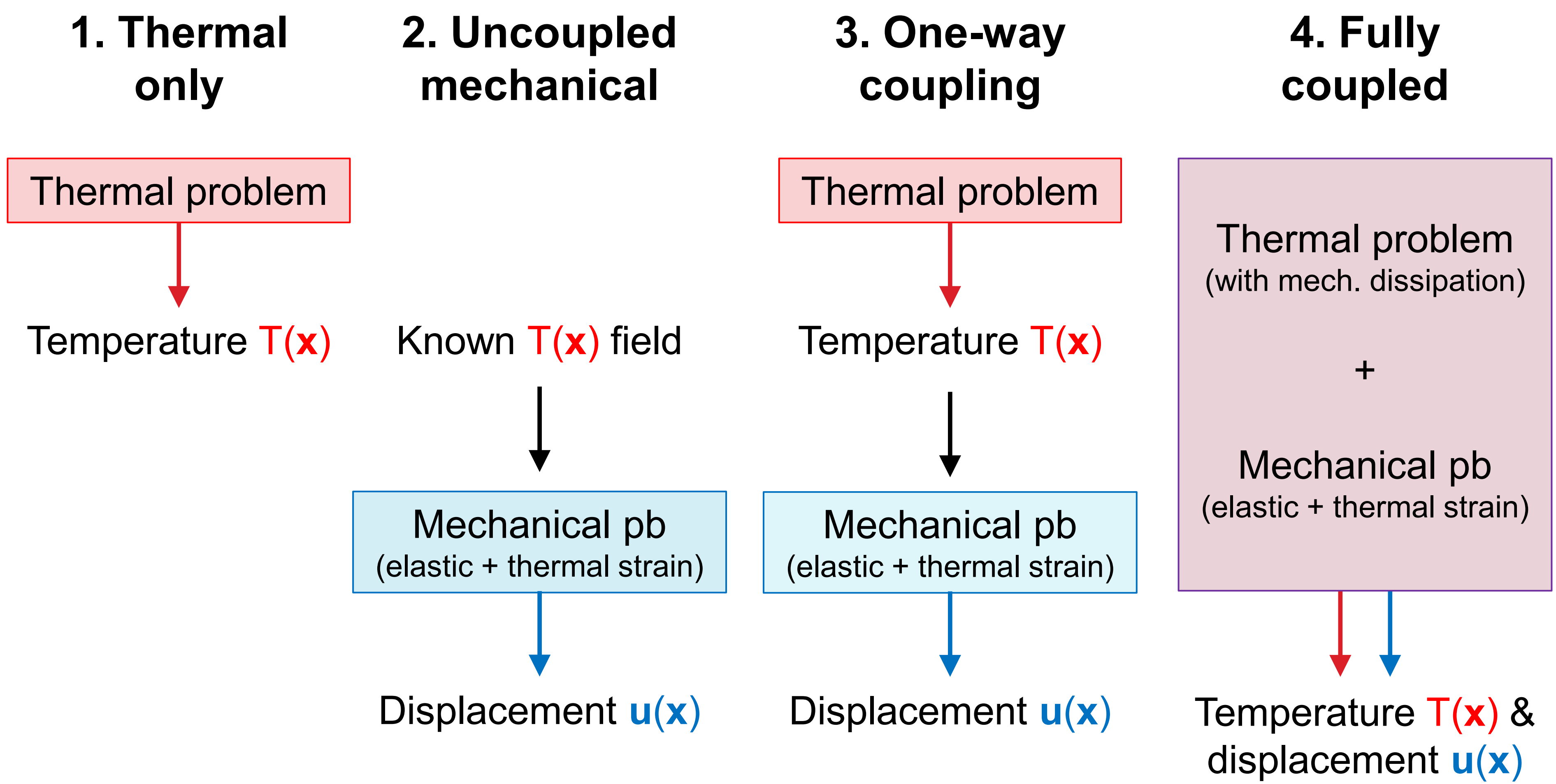
Four types of "thermo-mechanics"



Today's demo (piston)

Example (thermal switch)
+ Exo 9 (bearing support)

Four types of “thermo-mechanics”



In Abaqus:

1. Thermal only	2. Uncoupled mechanical	3. One-way coupling	4. Fully coupled
1 step: *“heat transfer”*	*1 step:* *“static”*	*2 steps:* *“heat transfer” → “static”*	*1 step:* *“coupled temp.-displ.”*

Thermal and mechanical problems

Thermal problem

Heat equation:

$$\rho c_p \frac{\partial T}{\partial t} + \nabla \cdot (\lambda \nabla T) = s$$

“conductivité thermique”
(coefficient de diffusion)

Mechanical pb
(elastic + thermal strain)

Total strain:

$$\varepsilon = \nabla u = \varepsilon^{el} + \varepsilon^{th} = \varepsilon^{el} + \alpha \Delta T$$

coefficient d’expansion th.

Force equilibrium
(+ Hooke’s law):

$$\begin{aligned} 0 &= \nabla \cdot \sigma + f \\ &= \nabla \cdot (C \varepsilon^{el}) + f \\ &= \nabla \cdot (C \left[\nabla u - \alpha \Delta T \right]) + f \end{aligned}$$

Thermal and mechanical problems

Thermal problem

Heat equation: $$\rho c_p \frac{\partial T}{\partial t} + \nabla \cdot (\lambda \nabla T) = \boxed{s}$$

Influence of deformations on the thermal problem

($s(\varepsilon)$ = heat source due to deformations)

Mechanical pb
(elastic + thermal strain)

Total strain: $$\varepsilon = \nabla u = \varepsilon^{el} + \varepsilon^{th} = \varepsilon^{el} + \alpha \Delta T$$

Force equilibrium
(+ Hooke's law):

$$\begin{aligned} 0 &= \nabla \cdot \sigma + f \\ &= \nabla \cdot (C \varepsilon^{el}) + f \\ &= \nabla \cdot (C [\nabla u - \boxed{\alpha \Delta T}]) + f \end{aligned}$$

Influence of temperature on the mechanical problem

Thermal loading in stress analysis

How to model thermal loading in static analysis:

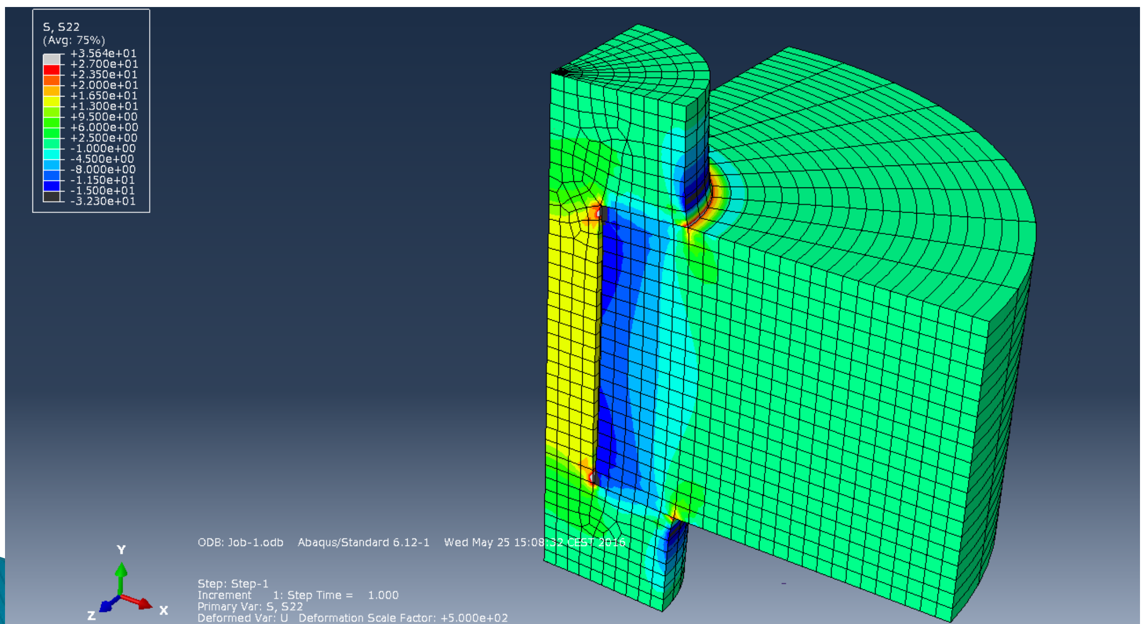
- Step= static, as usual
- Define material properties:
  - Coefficient of thermal expansion ("expansion"), α
  - Reference temperature corresponding to zero stress state, = 0 by default
- Loads:
  - Temperature field T(**x**) can be defined in static stress analysis using Load=>Predefined fields=>Other=> Temperature

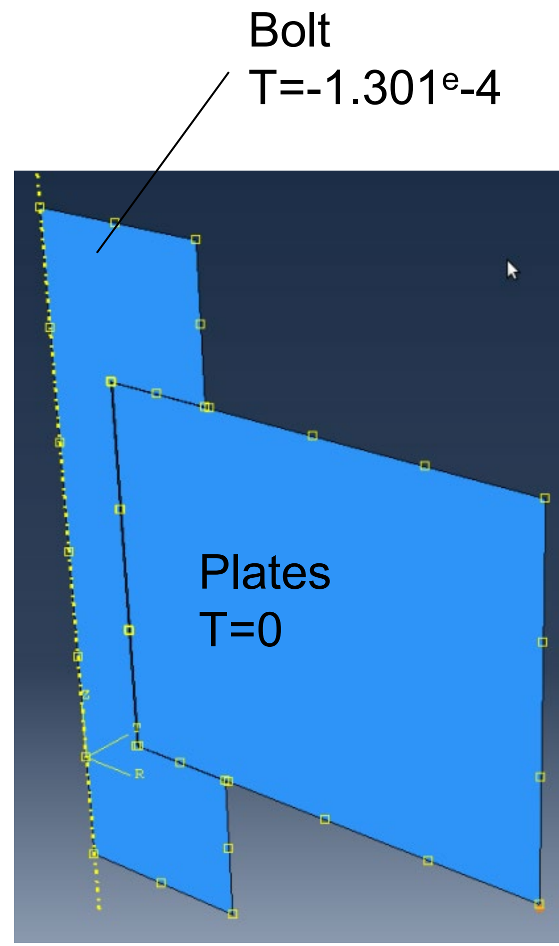
- Modeling preloads:
  - Very useful to define preloads in components:
    - Compute the strain ε_{th} corresponding to the preload
    - Specify a CTE $\alpha=1$ and a temperature change of the preloaded part $\Delta T = \varepsilon_{th}$ such that then $\varepsilon_{th} = \alpha \, \Delta T$

Thermal loading in stress analysis

Example: preloaded M8x20 bolt

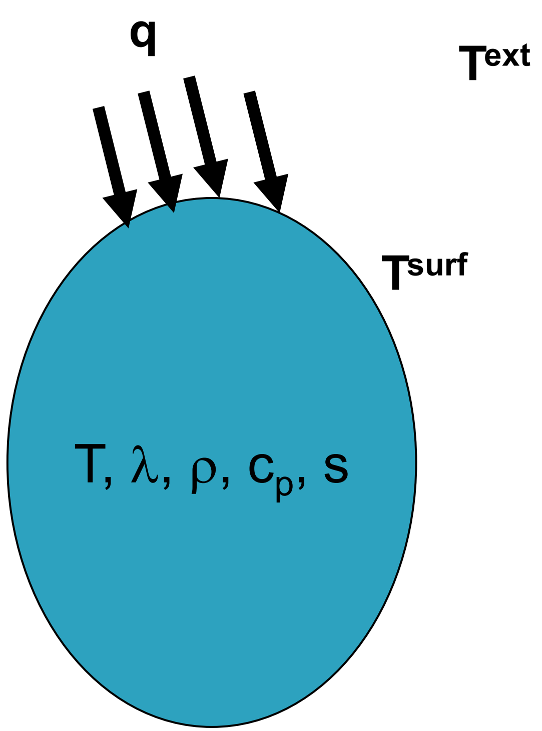
- Preload = 1000N, E = 210 GPa
- Equivalent section of M8 bolt= 36.6mm^2
- Prestress= 1000/36.6 = 27.3 MPa
- Prestrain= 27.3/210e3 = 1.301e-4
- Alpha=1, dT=-1.301e-4 (contraction)





Thermal problem

- Variables
  - Essential variable: Temperature field T
  - Natural variable: heat flux **q**
- Material
  - Conductivity λ
  - Density ρ & specific heat c_p if transient
  - Expansion coefficient for stress analysis
- Boundary conditions:
  - Temperature: $T^{surf} = f(t,x)$
  - Surface heat fluxes:
    - Imposed heat flux $\mathbf{q}^{surf}=f(t,x)$
    - Convection: $\mathbf{q}^{surf} = h\ (T^{surf}-T^{ext}(t))$
  - Volume heat source: $\mathbf{s} = f(t)$



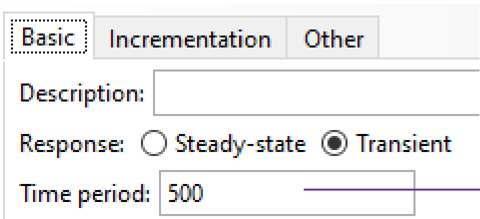
Thermal problem in Abaqus

- Select Step = **Heat transfer**
  - Choose **steady state or transient**
  - If transient: set time period, set small initial increment, set max ΔT per increment (<1/10 of max ΔT)
- In Mesh:
  - select **element type**: **Heat transfer,** linear
- Loading:
  - If steady, need to **impose at least one temperature** (otherwise "thermal rigid body").
  - **Adiabatic interface: leave free = no flux!**
  - Loading: Heat Flux = Load; Temperature = BC
  - Convection: in *Interaction* module, *create Surface Film condition*, enter h and T^{ext} => $\mathbf{q}^{surf} = h\,(T - T^{ext}(t))$
  - **If transient: define an amplitude curve** (tool => amplitude), need to start at T=0 for t=0. **Tabular** or **smooth step** are good choices.

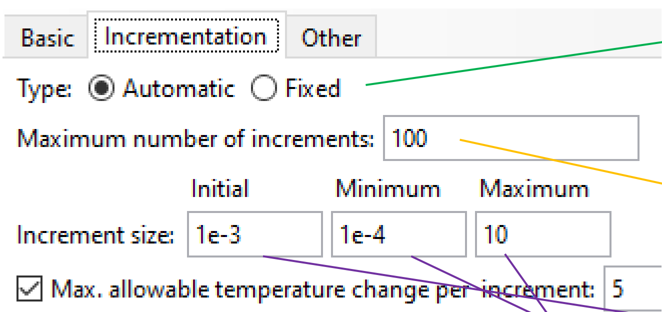
Coupled thermo-mechanics in Abaqus

- Select Step = Coupled Temp-Displacement
  - Choose **steady state or transient**
  - If transient: set time period, set small initial increment, set max ΔT per increment (<1/10 of max ΔT)
- In Mesh:
  - select **element type**: Coupled Temp.-Displ., quadratic
- Loading:
  - If steady, need to impose at least one temperature. Must also block 6 rigid body motions.
  - **Adiabatic interface: leave free = no flux!**
  - Loading: Heat Flux = Load; Temperature = BC
  - Convection: in *Interaction* module, *create Surface Film condition*, enter h and T^{ext} => $\mathbf{q}^{surf} = h\,(T - T^{ext}(t))$
  - **If transient: define an amplitude curve** (tool => amplitude), need to start at T=0 for t=0. **Tabular** or **smooth step** are good choices.

Transient simulation: time settings



Total physical time to be simulated.



Automatic incrementation: Abaqus adjusts the size of the time step (time increment dt from one iteration at t to the next iteration at t+dt), based on how quickly the solution converges.

Maximum number of iterations. Simulation will stop if reach this number of iterations.

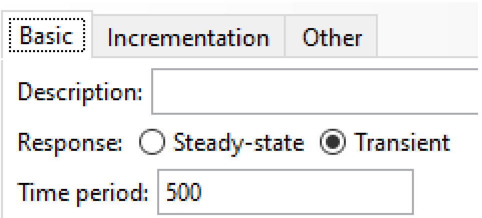
Initial increment size dt at t=0.

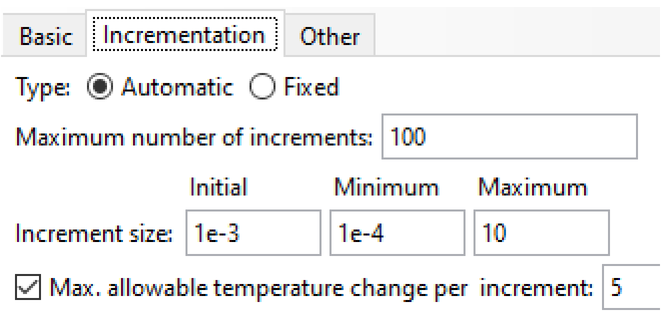
Minimum increment size dt. If at some point Abaqus needs a smaller time increment than this value to reach a convergent solution, it stops the analysis.

Maximum increment size dt. Abaqus will not increase the increment size beyond this value.

Abaqus will limit the increment size dt such that the temperature T does not vary by more than this value between t and t+dt anywhere in the model (optional).

Transient simulation: time settings





In this example, the transient simulation:

- is from t=0 s to t=500 s,
- will start with dt=0.001 s,
- will then automatically adjust dt between 0.0001 s and 10 s (→ number of iterations falls necessarily between 500/10=50 and 500/0.0001=5e6),
- will also limit dt such that T varies by no more than 5 degrees between t and t+dt,
- will stop if it reaches 100 iterations.

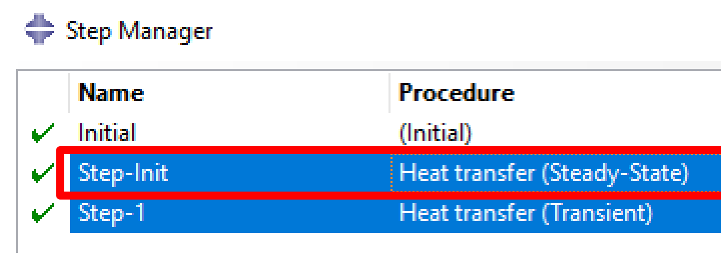
Thermal problem in Abaqus

- Note #1: **units of temperature**
  - By default, Abaqus uses K, but one can choose to work with °C. Unimportant if no radiation and no source term: solution of heat equation defined up to a uniform constant.
  - However, everything must be defined consistently (boundary and initial conditions, etc).

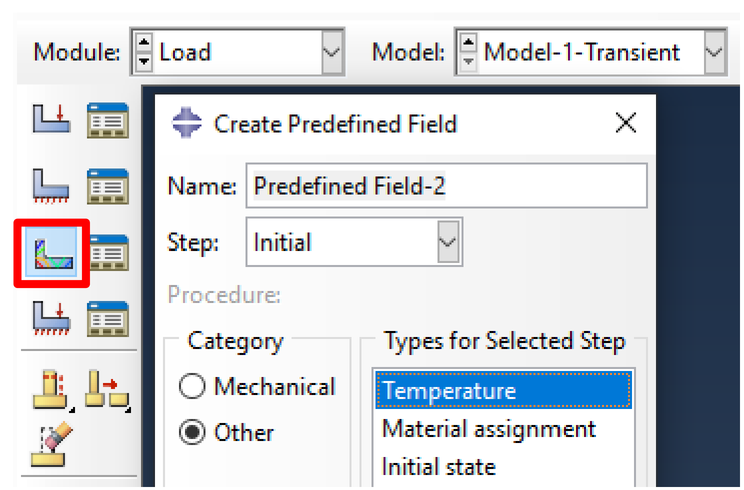
- Note #2: **absolute zero**
  - Possible to define the value of the absolute zero temperature (Model → Edit attributes → Physical constants).
  - E.g., setting −273.15 implicitly defines the units as °C, while setting 0 (default) implicitly defines the units as K.

Thermal problem in Abaqus

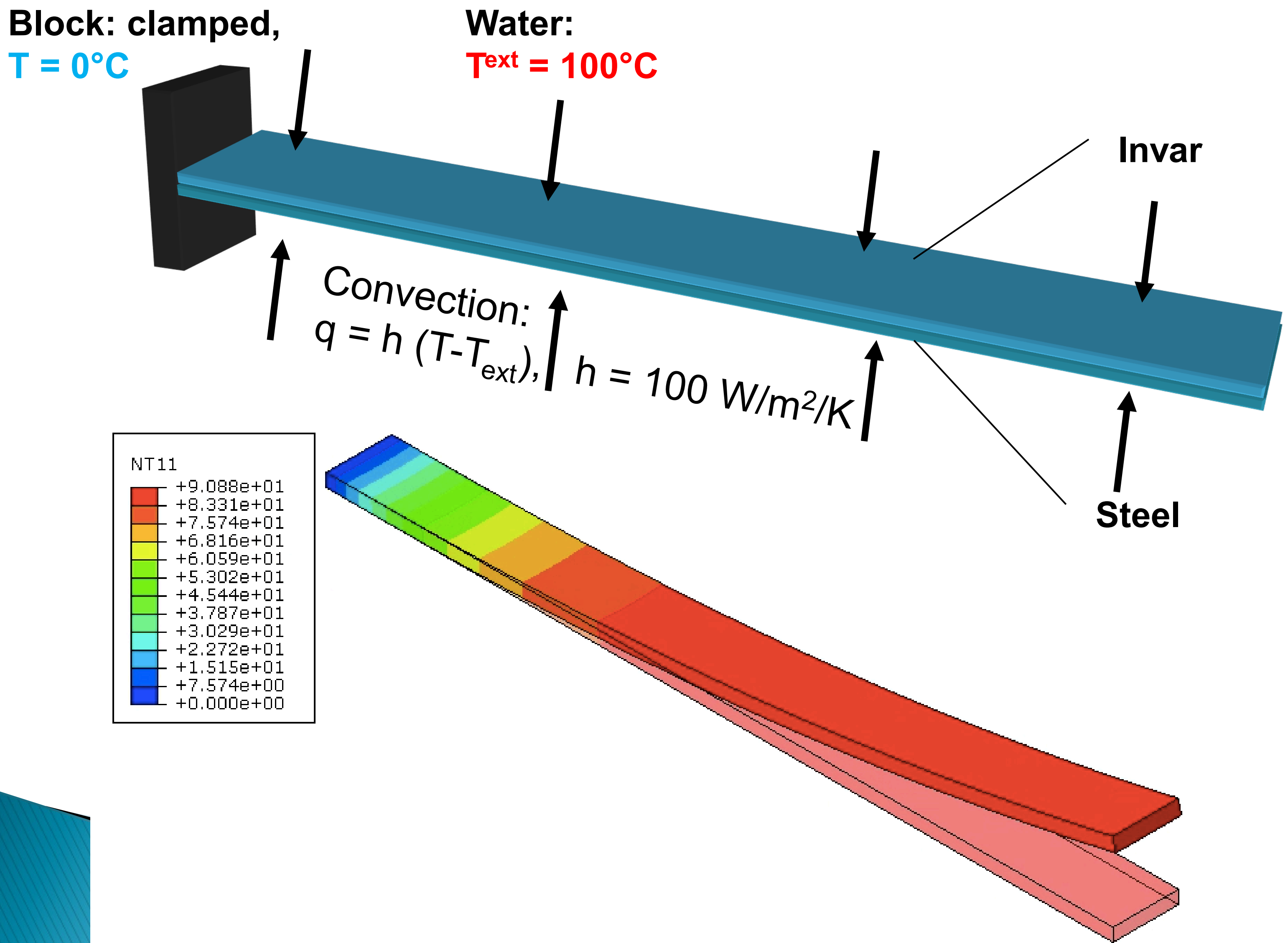
- Note #3: **initial conditions**
  - By default, initial temperature in unsteady simulations: T(t=0)=0.
  - Possible to define a non-zero initial condition T(x):
    - In a **purely thermal** analysis, need a steady *heat transfer* step with the desired T as boundary condition before the unsteady *heat transfer* step.



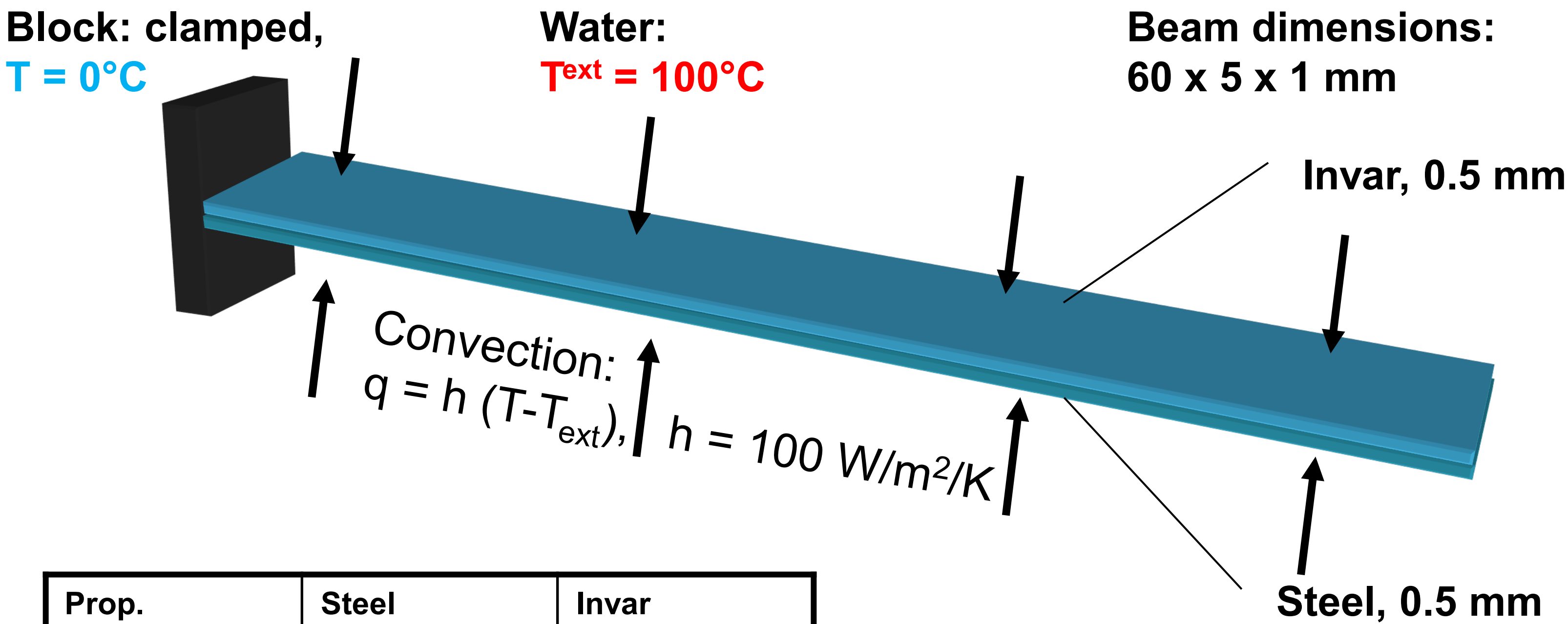
    - In a **coupled thermo-mechanical** analysis (*coupled temperature-displacement* step), can create a "**Predefined field**" for the Initial step in the module "Load" (see for example exercise 10).



Example: bi-material beam (thermal switch)



Example: bi-material beam (thermal switch)



Prop.	Steel	Invar
Young's modulus	210 GPa	141 GPa
Poisson ratio	0.3	0.3
Th. Expansion	1e-5	1e-6
Density	7800 kg/m3	8000 kg/m3
Conductivity	30 W/m/K	10 W/m/K
Specific heat	1000 J/kg/K	500 J/kg/K