Lien entre «théorie» et étapes de simulation

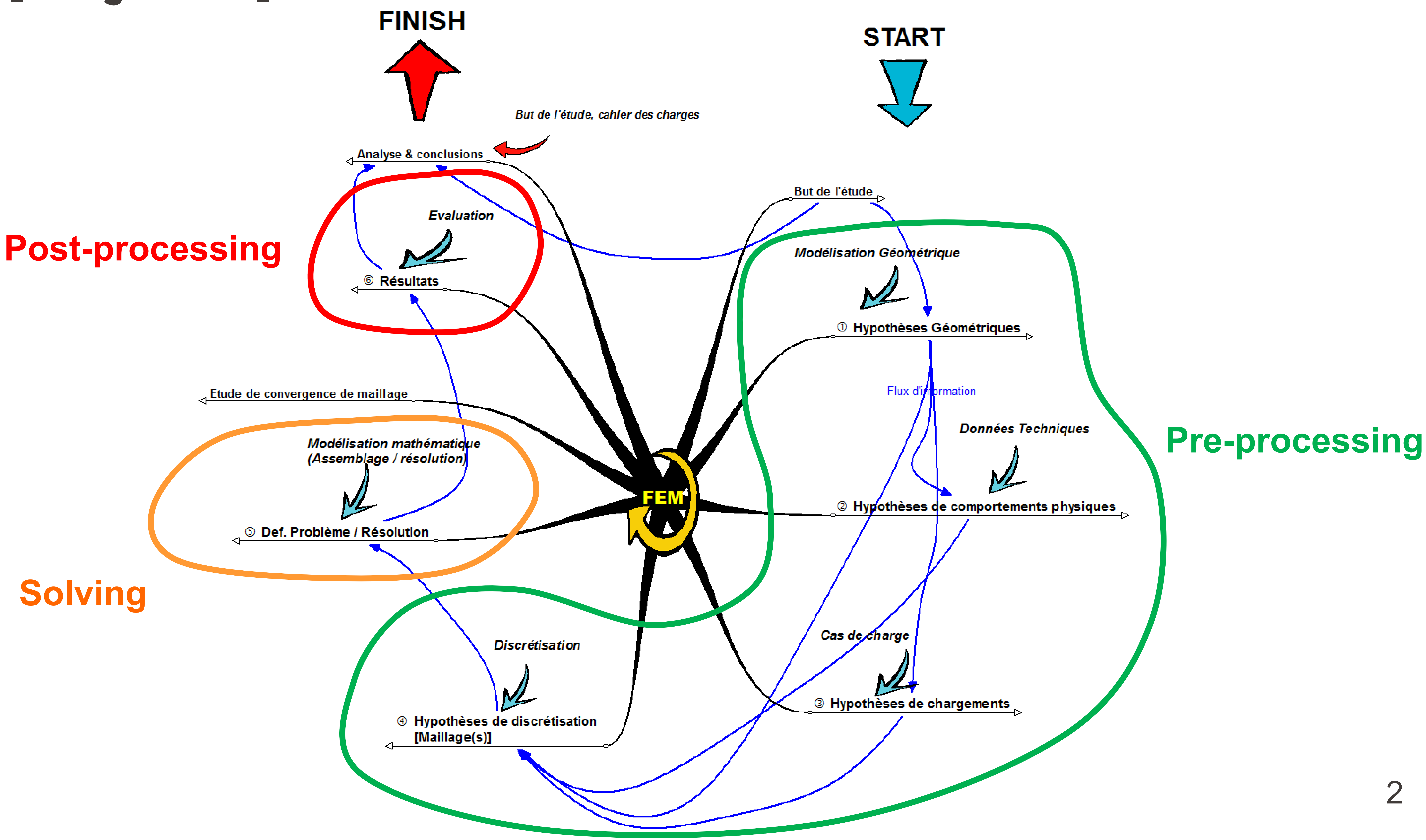
Modélisation et simulation par éléments finis

Edouard Boujo

Printemps 2025

École polytechnique fédérale de Lausanne

Aperçu du processus de simulation



Modélisation et simulation par éléments finis

Abaqus : 3 programmes, 1 interface graphique

Pre processor (Abaqus CAE)

Geometry	Physics of the problem	Physical properties	Boundary conditions	Meshing	Write input files

Solver (Abaqus Standard or Explicit)

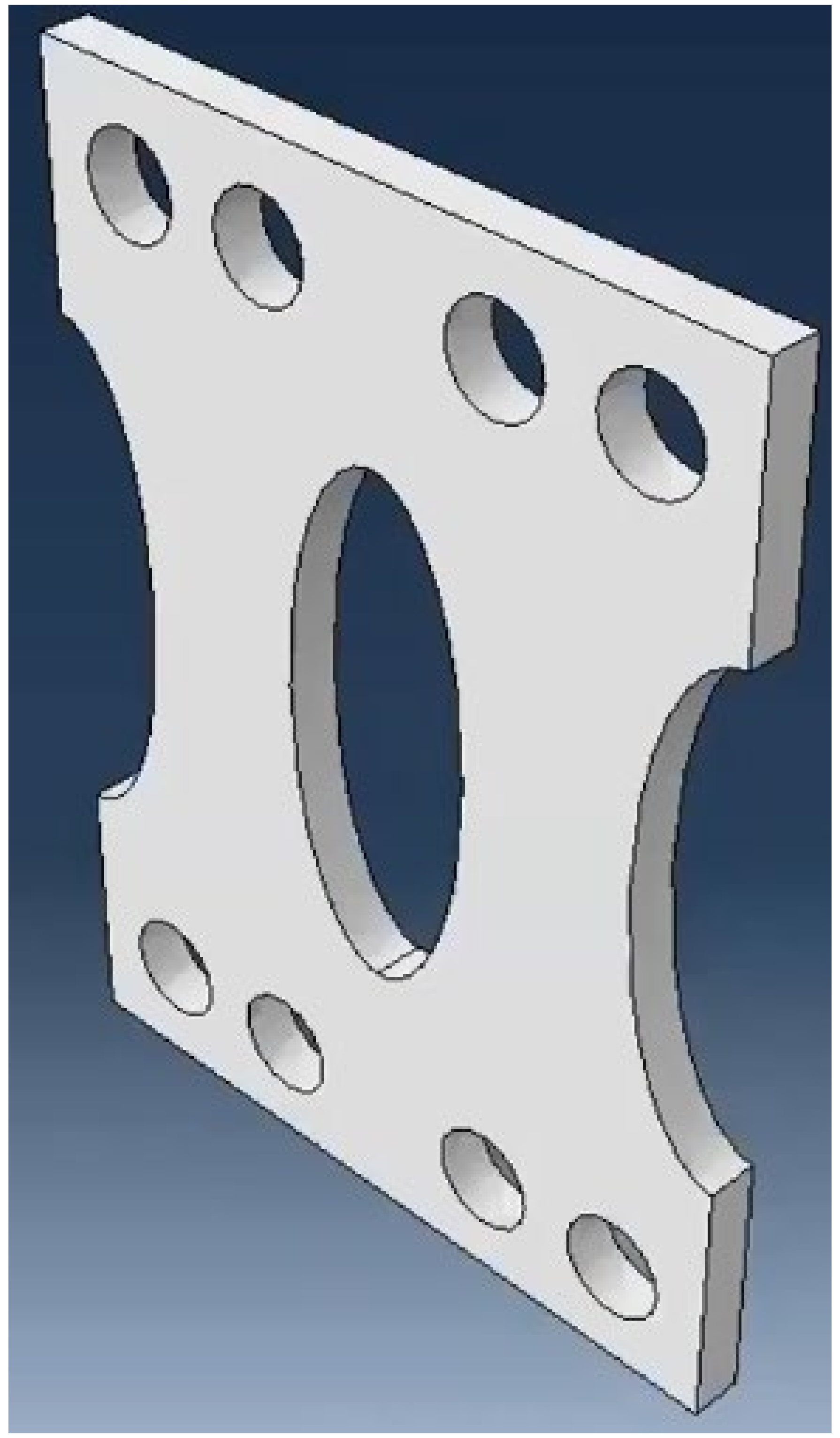
Read FE model description (nodes, elements, properties, BC)	Integrate finite element matrices	Assemble the global matrices	Solve the problem for the essential variables	Calculate derived quantities for post processing, write output

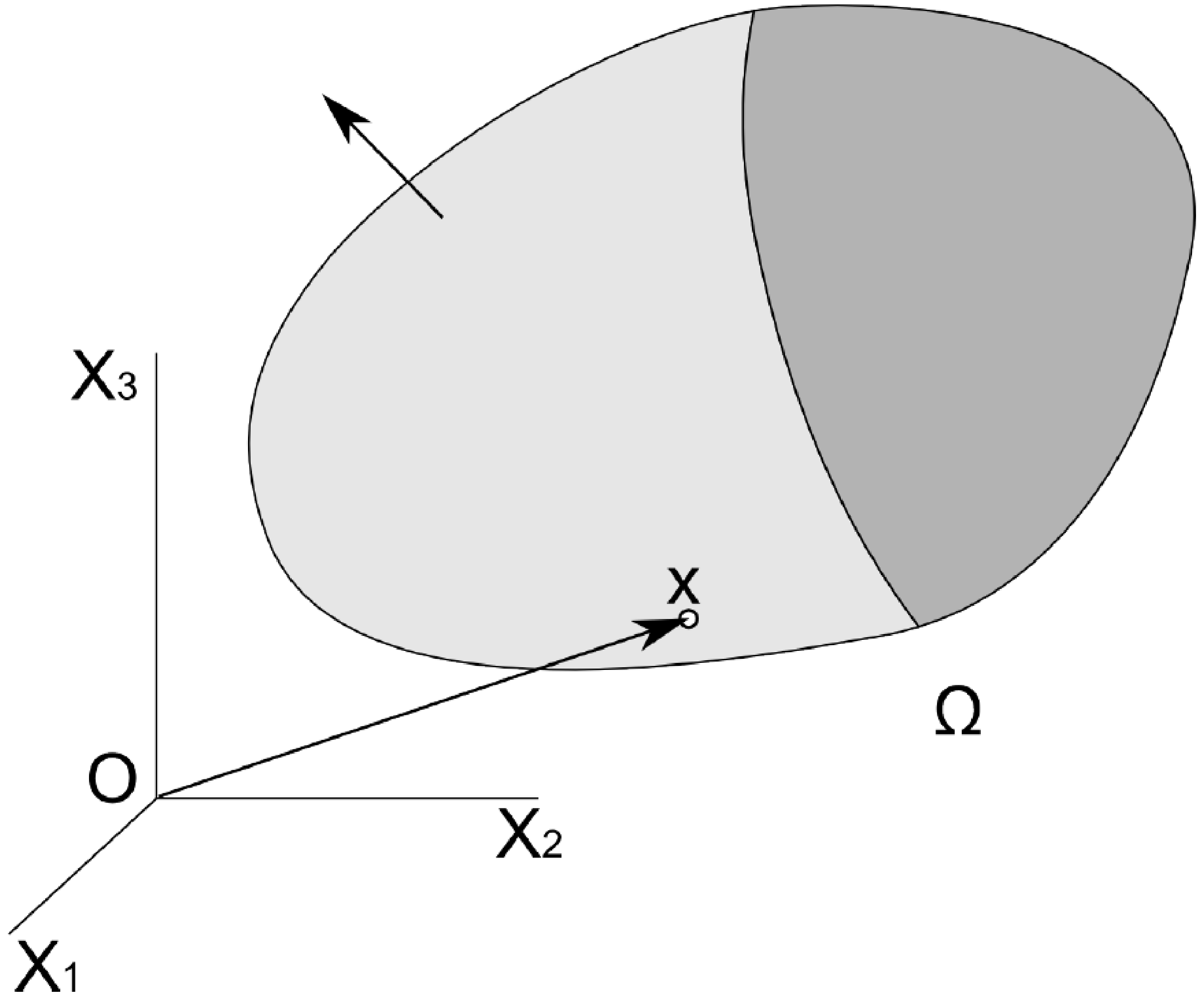
Post processor (Abaqus CAE/Viewer)

Load result database & mesh	Calculate derived quantities (invariants, criteria, averages)	Display the results in several ways	Extract data (1D or 2D plot) and export to Text files

Abaqus : module “Part”

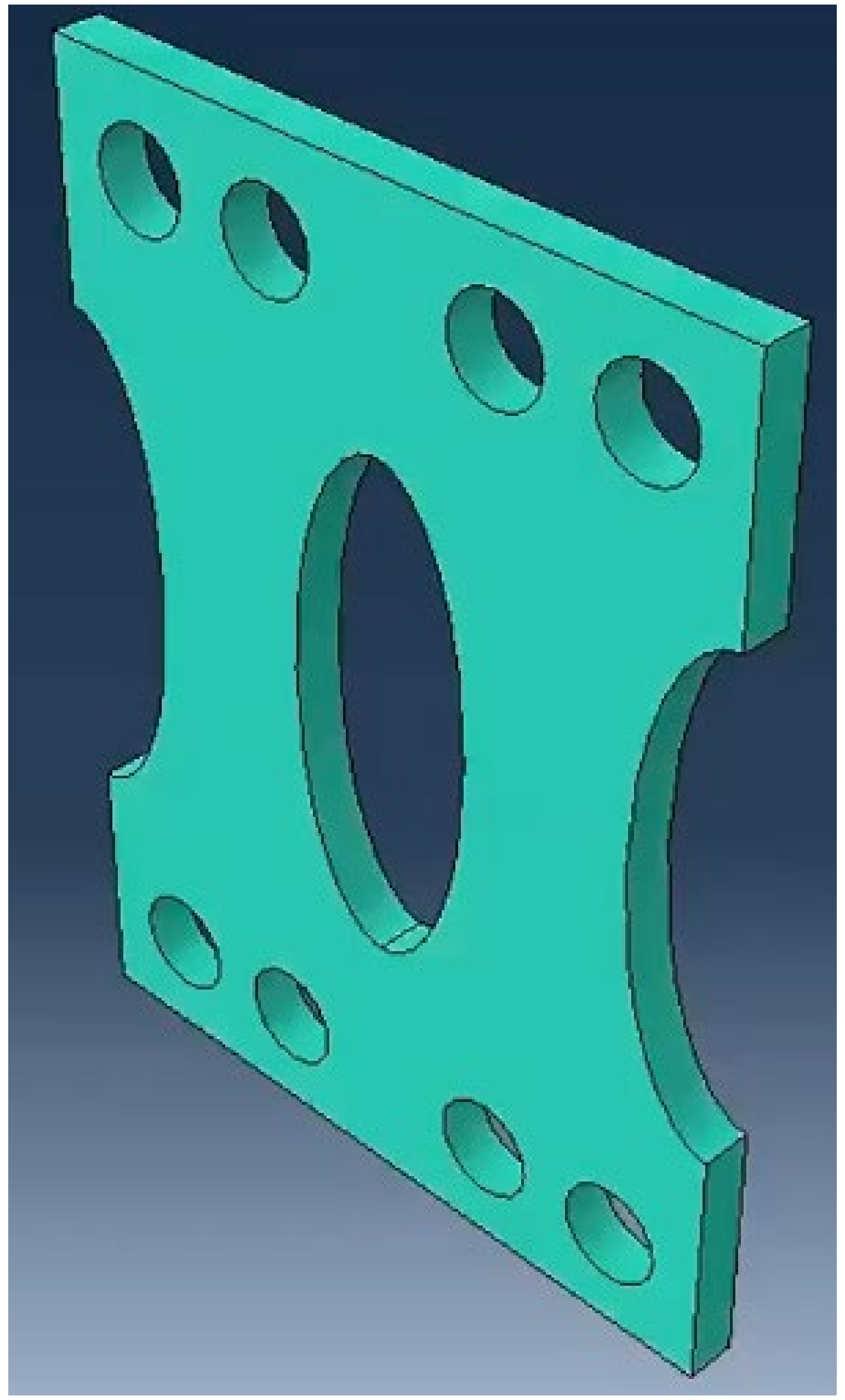
- Définition du domaine Ω (géométrie, coordonnées)

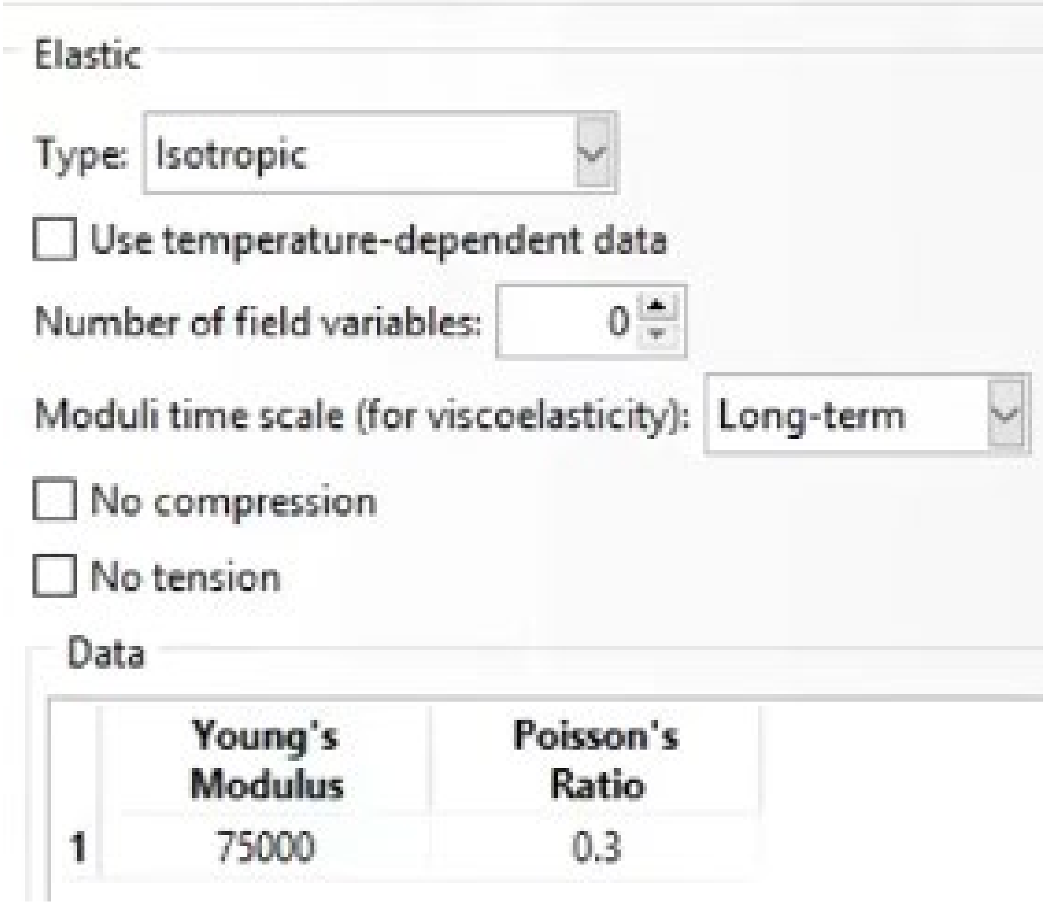




Abaqus : module "Property"

- Définition des matériaux : comportement du matériau (e.g. élastique isotrope) et valeurs numériques des constantes (e.g. module de Young *E* et coefficient de Poisson *v*)





Elastic

Type: Isotropic

☐ Use temperature-dependent data

Number of field variables: 0

Moduli time scale (for viscoelasticity): Long-term

☐ No compression

☐ No tension

Data

	Young's Modulus	Poisson's Ratio
1	75000	0.3

Abaqus : module “Property”

- Définition des matériaux : comportement du matériau (e.g. élastique isotrope) et valeurs numériques des constantes (e.g. module de Young *E* et coefficient de Poisson *v*)

- Définition des sections : matériaux + section si poutre / coque (dimensions, moment d’inertie etc)

- Assignation des sections aux régions → matrice des constantes élastiques **C**(**x**)

$$\mathbf{C} = \Delta \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix}$$

with: $\Delta = \frac{E}{(1+\nu)(1-2\nu)}$

Abaqus : module "Step"

- Définition du type d'analyse (statique / dynamique ? élasticité linéaire / thermique ?) → type d'inconnue (**q** = déplacement / temperature…)

Exemples :

1. Step "Static, General" → on va résoudre l'équilibre des contraintes.

$$div(\sigma(\mathbf{x})) + \mathbf{f}(\mathbf{x}) = 0$$

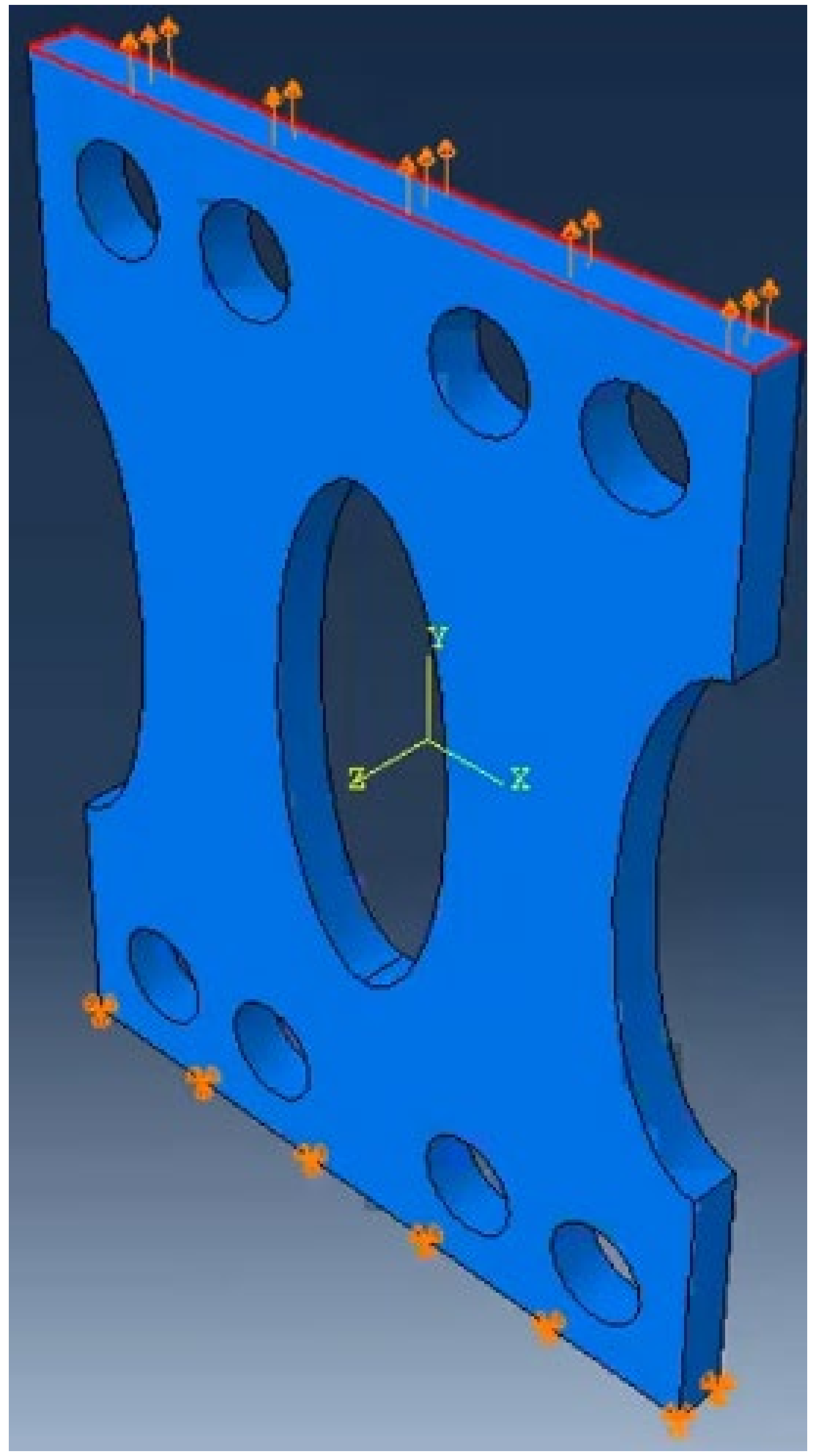
   En forme faible et après discrétisation, on obtient le système algébrique **K q** = **r**, dont l'inconnue **q** contient les déplacements **u**=(u_1,u_2,u_3) aux noeuds.

2. Step "Heat transfer" → on va résoudre l'équation de la chaleur.

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

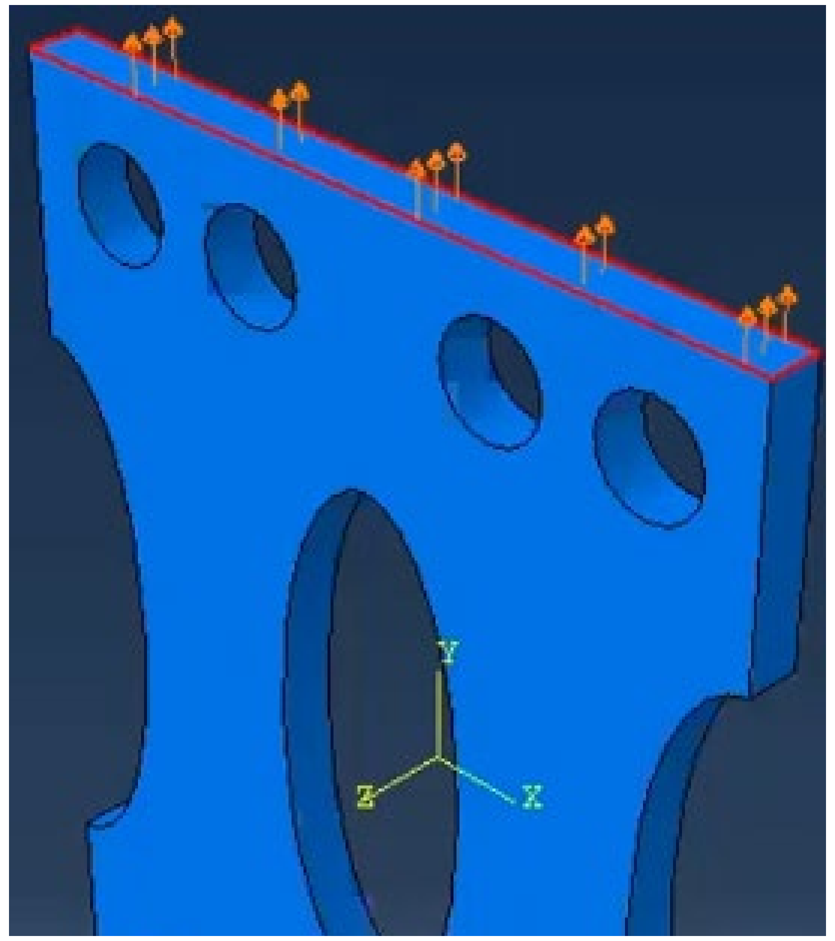
   En forme faible et après discrétisation, on obtient le système algébrique **K q** = **r**, dont l'inconnue **q** contient la température T aux noeuds.

Abaqus : module “Load”

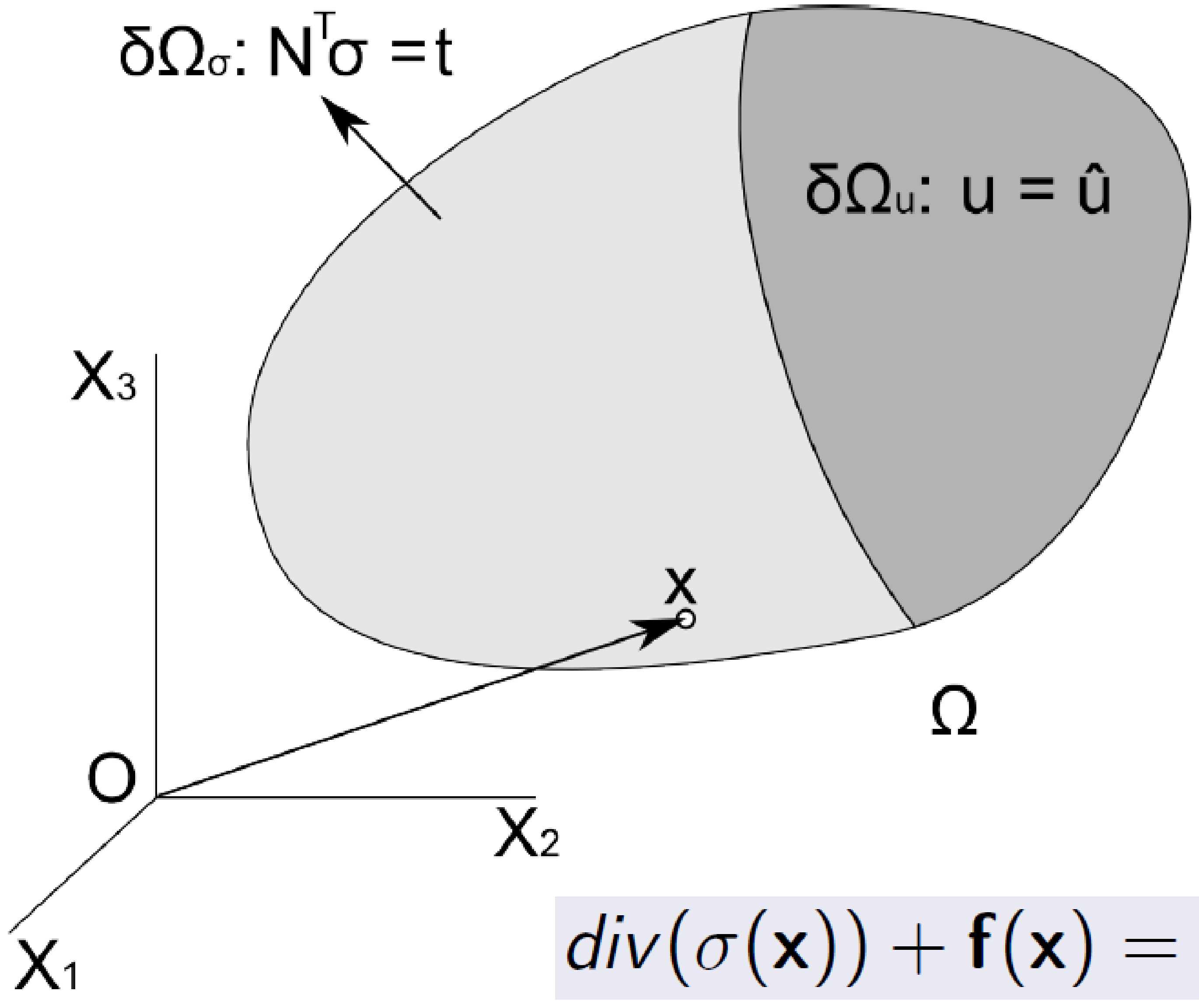


- Définition :
  - des conditions aux limites naturelles (Neumann = “Load” ; par ex. chargements surfaciques),
  - des conditions aux limites essentielles (Dirichlet = “Boundary condition” ; par ex. déplacements),
  - des charges volumiques (“Load” ; par ex. gravité, force centrifuge etc.).

Abaqus : module “Load”



- Définition :
  - des conditions aux limites naturelles (Neumann = “Load” ; par ex. chargements surfaciques),
  - des conditions aux limites essentielles (Dirichlet = “Boundary condition” ; par ex. déplacements),
  - des charges volumiques (“Load” ; par ex. gravité, force centrifuge etc.).



$$div(\sigma(\mathbf{x})) + \mathbf{f}(\mathbf{x}) = 0$$

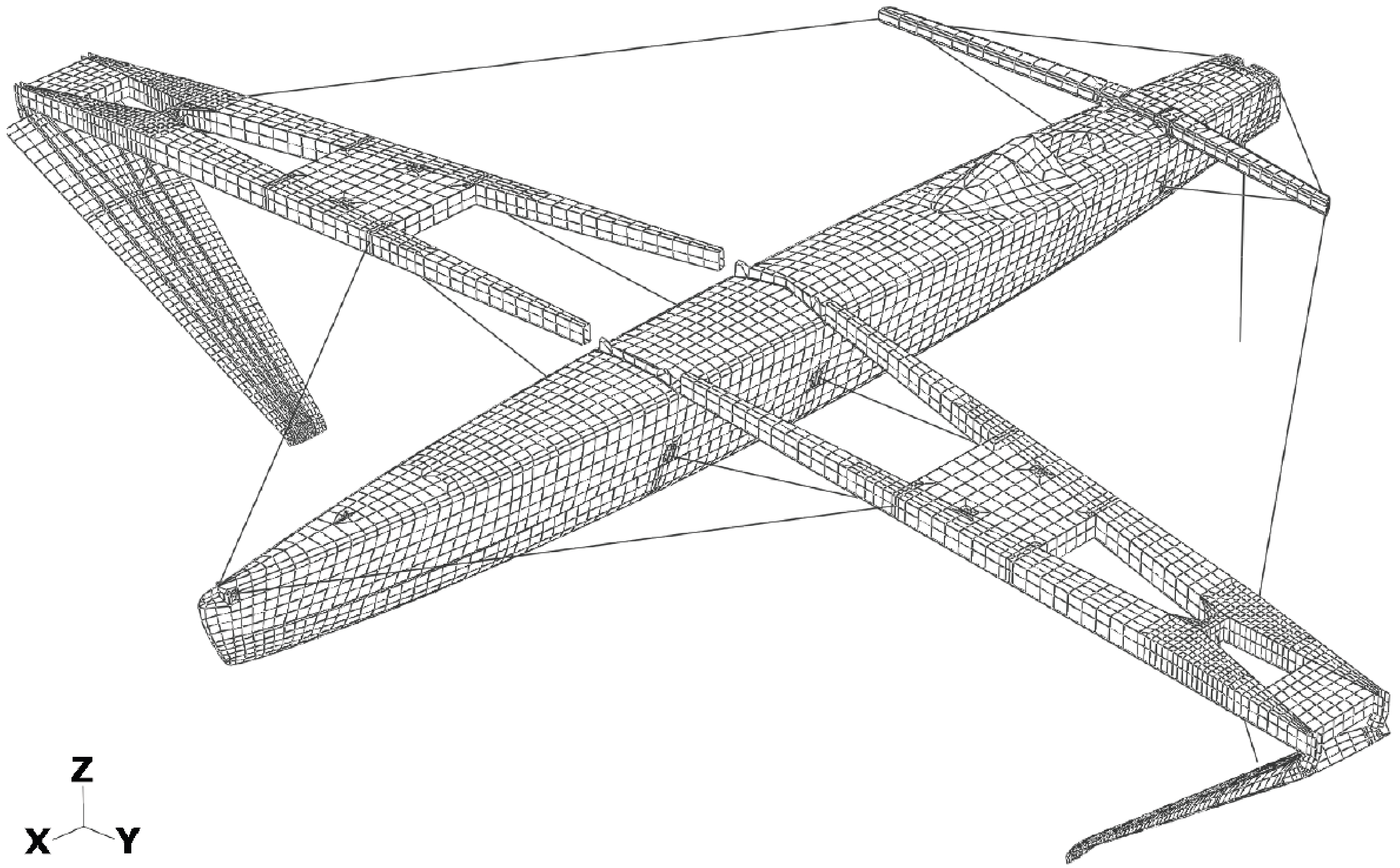
→ Régions $\delta\Omega_\sigma$ et $\delta\Omega_u$ + valeurs et orientations des forces surfaciques **t**, des déplacements **û** (pour une face, implique que tous les noeuds de cette face ont un déplacement nodal **q** imposé) et des forces volumiques **f**.

Abaqus : module “Mesh”

- Définition :
  - du type d’élément (→ géométrie de l’élément de reference Ω^a),
  - du type de fonction de base (e.g. linéaires / quadratiques → définit le nombre de noeuds par élément) et du schema d’intégration,
  - de la taille des éléments.

Abaqus : module “Mesh”

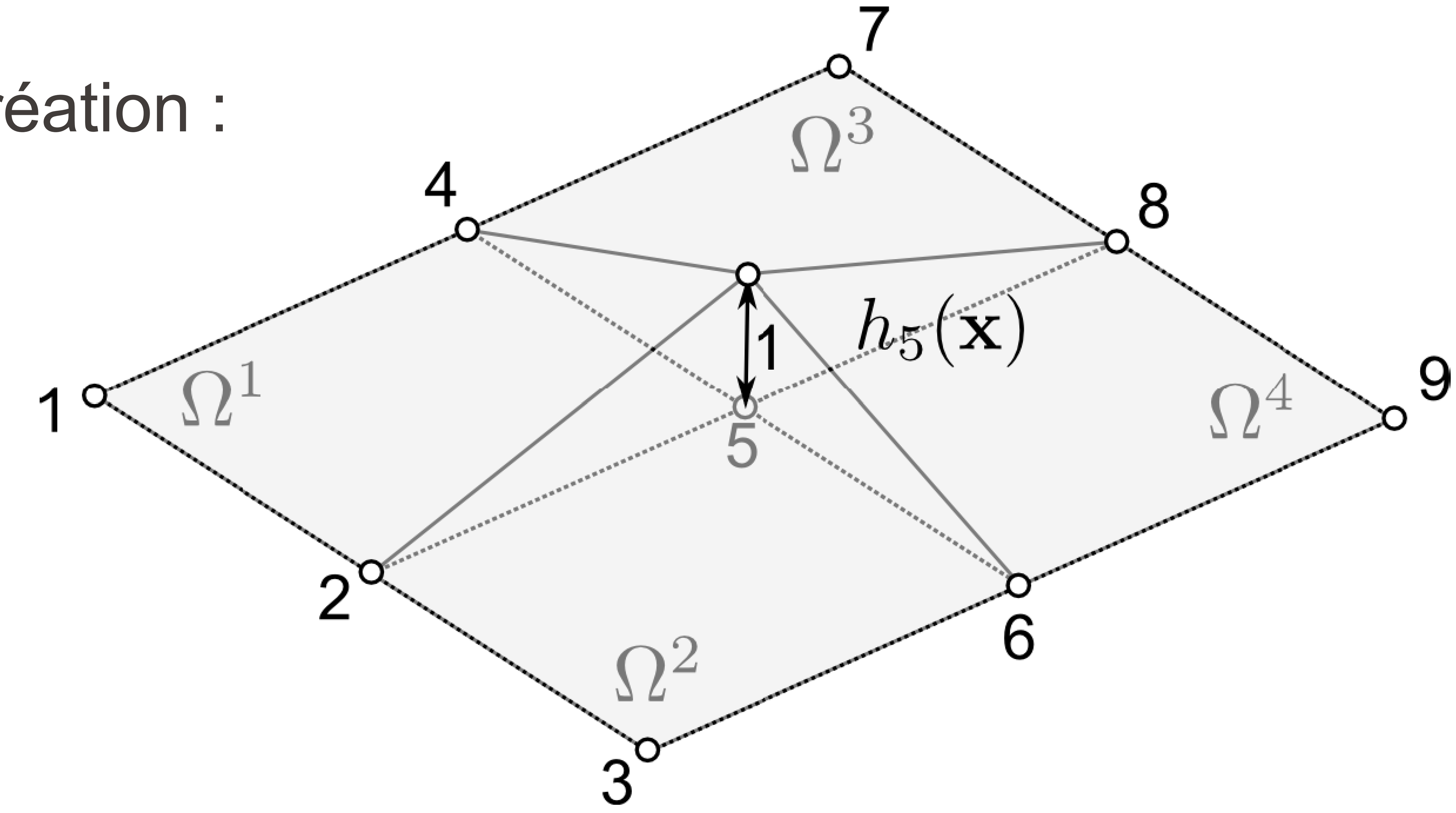
- Définition :
  - du type d’élément (→ géométrie de l’élément de reference Ω^a),
  - du type de fonction de base (e.g. linéaires / quadratiques → définit le nombre de noeuds par élément) et du schema d’intégration,
  - de la taille des éléments.
- Maillage proprement dit → création :
  - de tous les éléments Ω^e,



Abaqus : module “Mesh”

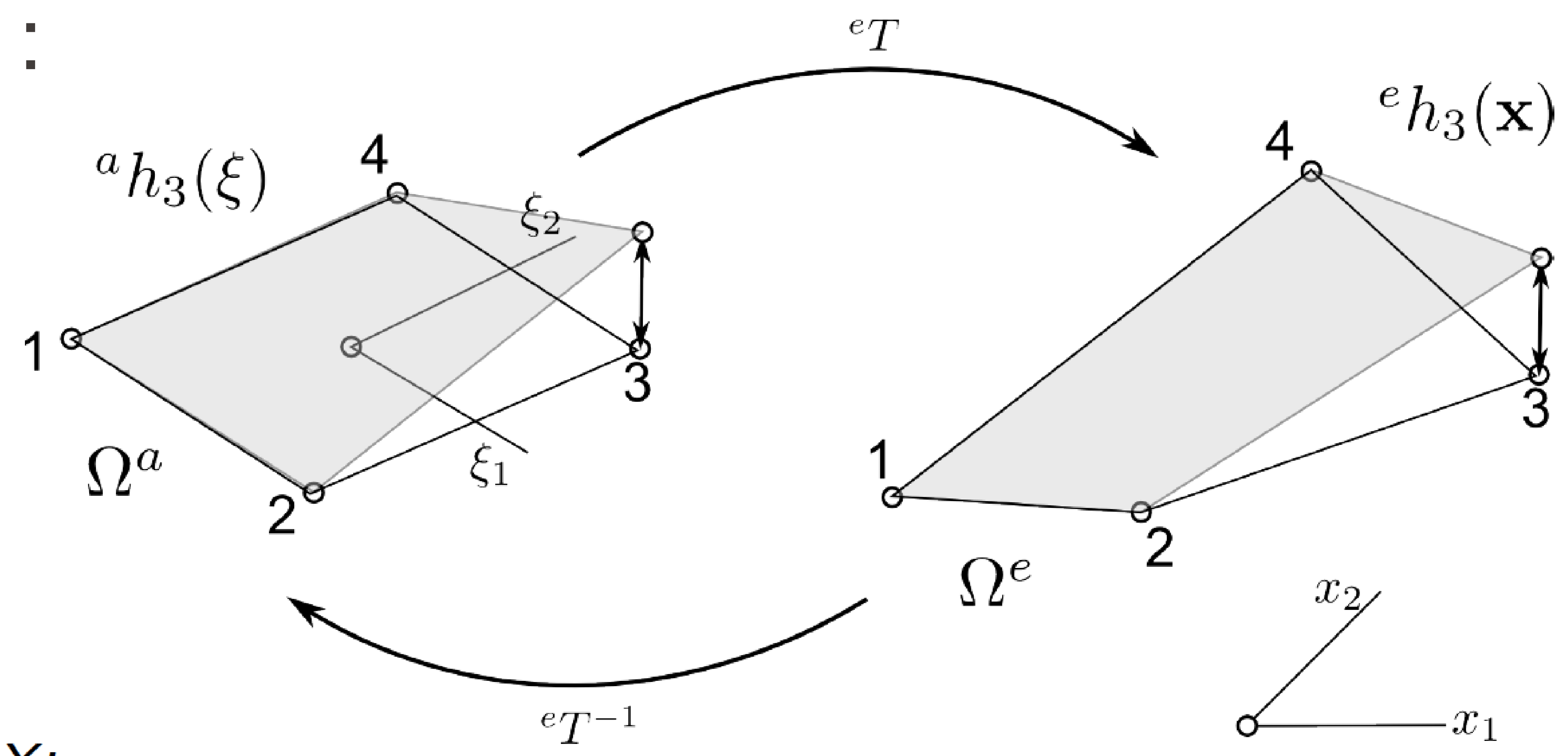
- Définition :
  - du type d’élément (→ géométrie de l’élément de reference Ω^a),
  - du type de fonction de base (e.g. linéaires / quadratiques → définit le nombre de noeuds par élément) et du schema d’intégration,
  - de la taille des éléments.
- Maillage proprement dit → création :
  - de tous les éléments Ω^e,
  - de la table de connectivité,

${}^e\Omega$	Ω^1	Ω^2	Ω^3	Ω^4
1	1	2	4	5
2	2	3	5	6
3	5	6	8	9
4	4	5	7	8



Abaqus : module “Mesh”

- Définition :
  - du type d’élément (→ géométrie de l’élément de reference Ω^a),
  - du type de fonction de base (e.g. linéaires / quadratiques → définit le nombre de noeuds par élément) et du schema d’intégration,
  - de la taille des éléments.
- Maillage proprement dit → création :
  - de tous les éléments Ω^e,
  - de la table de connectivité,
  - des coordonnées des noeuds, des fonctions de forme ${}^a h(\xi)$, de la transformation ${}^e T$ et de la Jacobienne ${}^e\mathbf{J}$

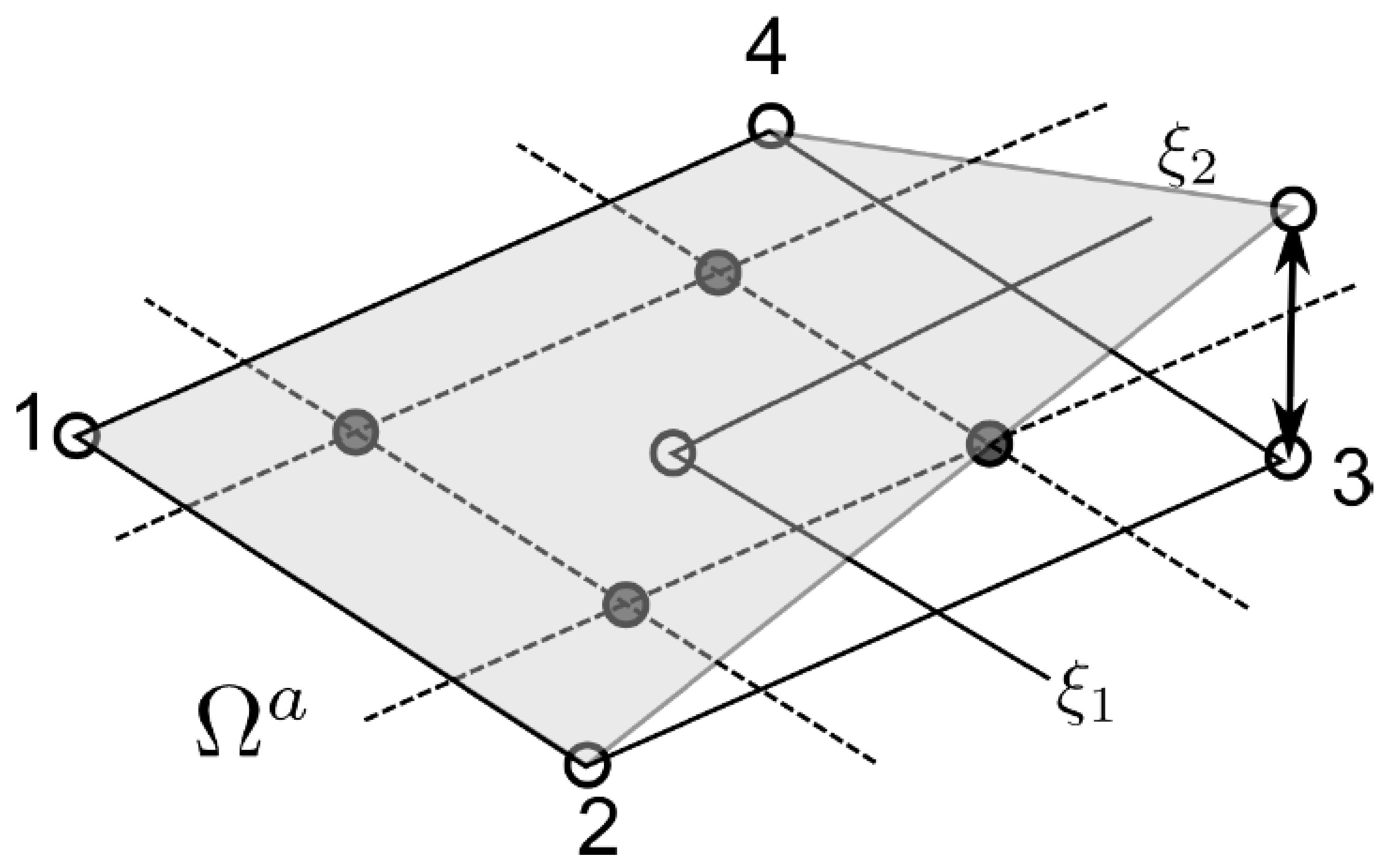


$$ {}^e\mathbf{J}_{ij} = \frac{\partial x_i}{\partial \xi_j} $$

Modélisation et simulation par éléments finis

Abaqus : module “Mesh”

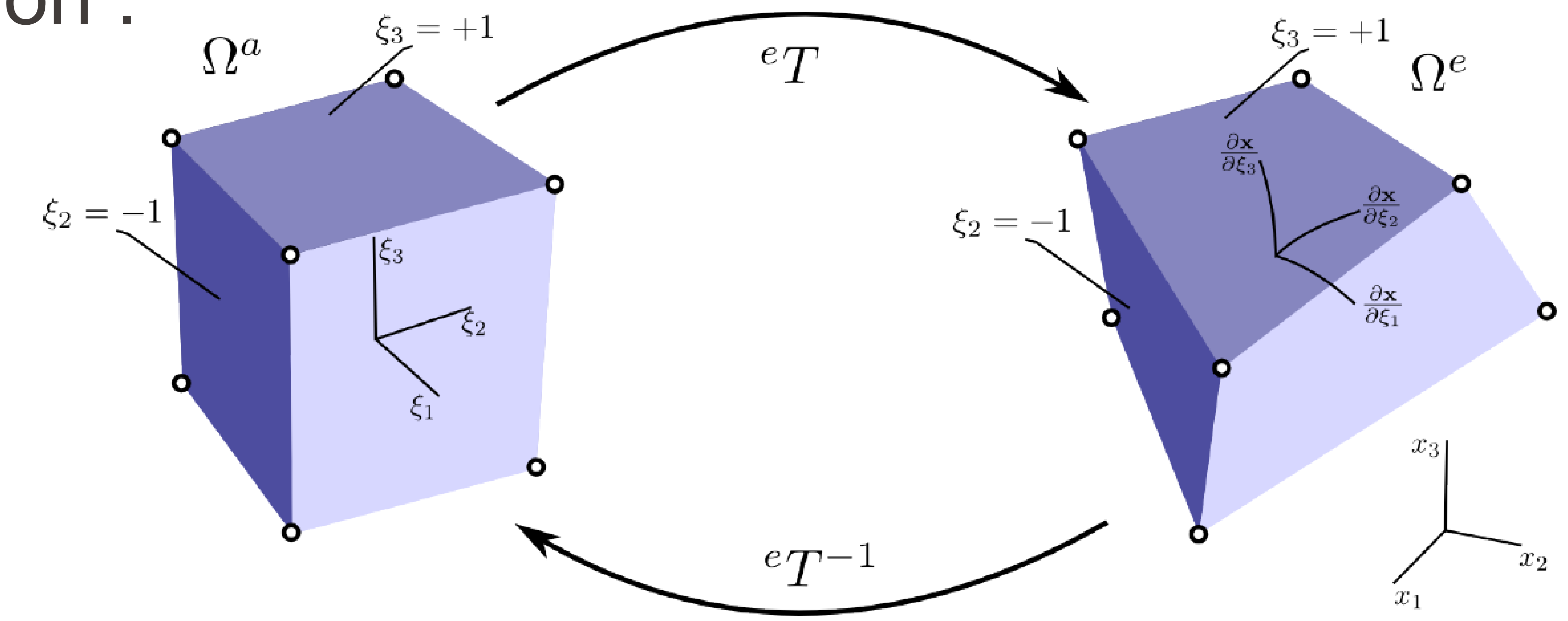
- Définition :
  - du type d’élément (→ géométrie de l’élément de reference Ω^a),
  - du type de fonction de base (e.g. linéaires / quadratiques → définit le nombre de noeuds par élément) et du schema d’intégration,
  - de la taille des éléments.
- Maillage proprement dit → création :
  - de tous les éléments Ω^e,
  - de la table de connectivité,
  - des coordonnées des noeuds,
    des fonctions de forme ${}^a h(\xi)$,
    de la transformation ${}^e T$
    et de la Jacobienne ${}^e\mathbf{J}$,
  - de l’intégration numérique



$$\int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1} (\cdot)\, d\xi_3 d\xi_2 d\xi_1 \approx \sum_{i=1}^{r}\sum_{j=1}^{s}\sum_{k=1}^{t} \omega_1^i \omega_2^j \omega_3^k (\cdot)\, \big|_{\xi_1=\xi_1^i, \xi_2=\xi_2^j, \xi_3=\xi_3^k}$$

Abaqus : module “Mesh”

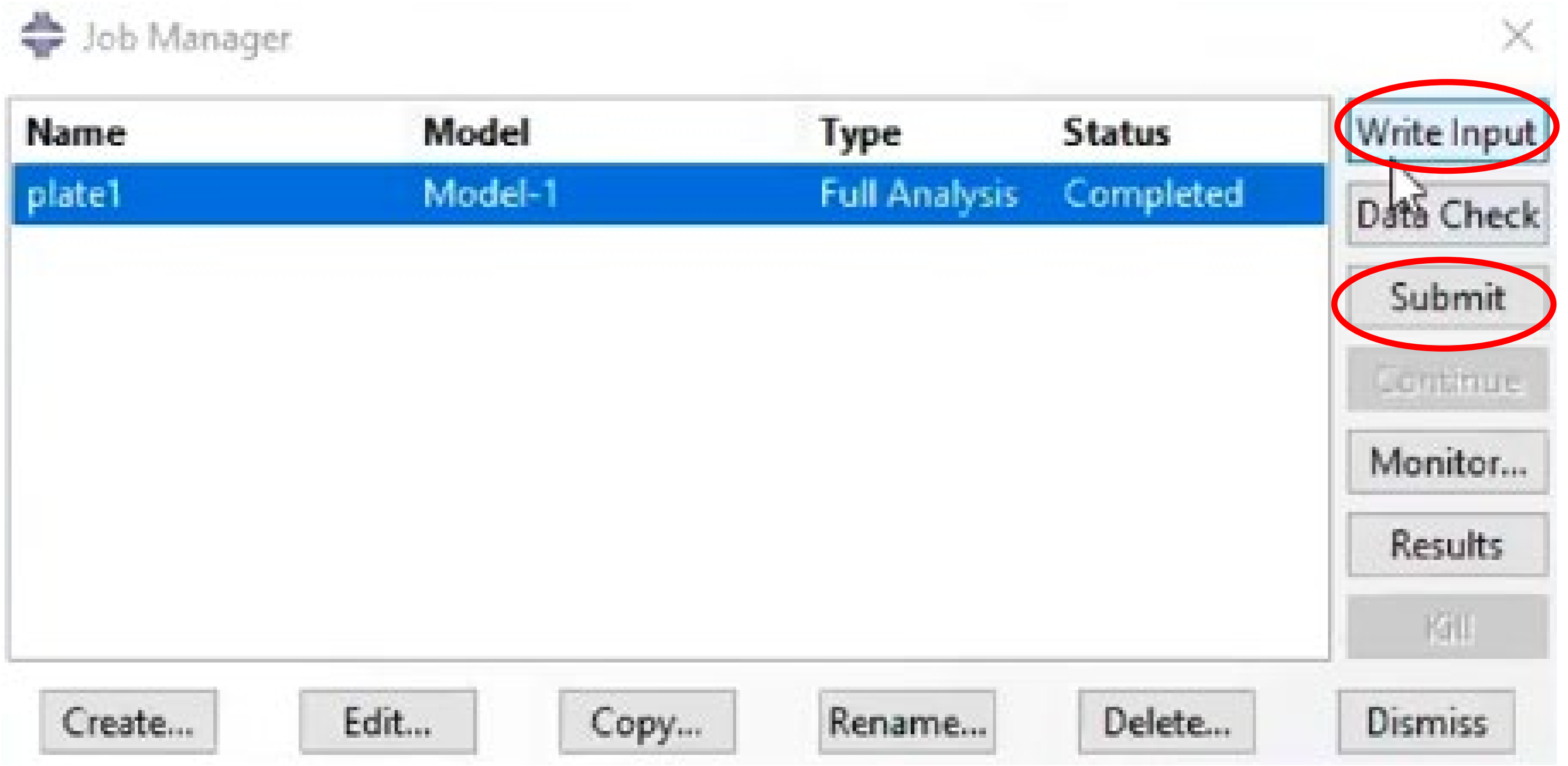
- Définition :
  - du type d’élément (→ géométrie de l’élément de reference Ω^a),
  - du type de fonction de base (e.g. linéaires / quadratiques → définit le nombre de noeuds par élément) et du schema d’intégration,
  - de la taille des éléments.
- Maillage proprement dit → création :
  - de tous les éléments Ω^e,
  - de la table de connectivité,
  - des coordonnées des noeuds, des fonctions de forme ${}^a h(\xi)$, de la transformation ${}^e T$ et de la Jacobienne ${}^e\mathbf{J}$,
  - de l’intégration numérique



Exemple : hexahèdre (→ cube), fcts de forme quadratiques → 20 noeuds, schema d’intégration réduit → “C3D20R”

Abaqus : module “Job”

- Résolution du problème : fin de la modélisation, début de la résolution numérique.
- Dans Abaqus, après “Write Input” ou “Submit”, le pré-processeur crée un fichier *.inp avec toutes les informations nécessaires au solveur pour assembler le problème (matrice **K** et vecteur **r**) et le résoudre :



Abaqus : module “Job”

- Résolution du problème : fin de la modélisation, début de la résolution numérique.
- Dans Abaqus, après “Write Input” ou “Submit”, le pré-processeur crée un fichier *.inp avec toutes les informations nécessaires au solveur pour assembler le problème (matrice **K** et vecteur **r**) et le résoudre :
  - Maillage : noeuds (coordonnées), éléments (type + table de connectivité)
  - Matériaux : type, valeurs des constantes élastiques
  - Section
  - Step
  - Conditions limites

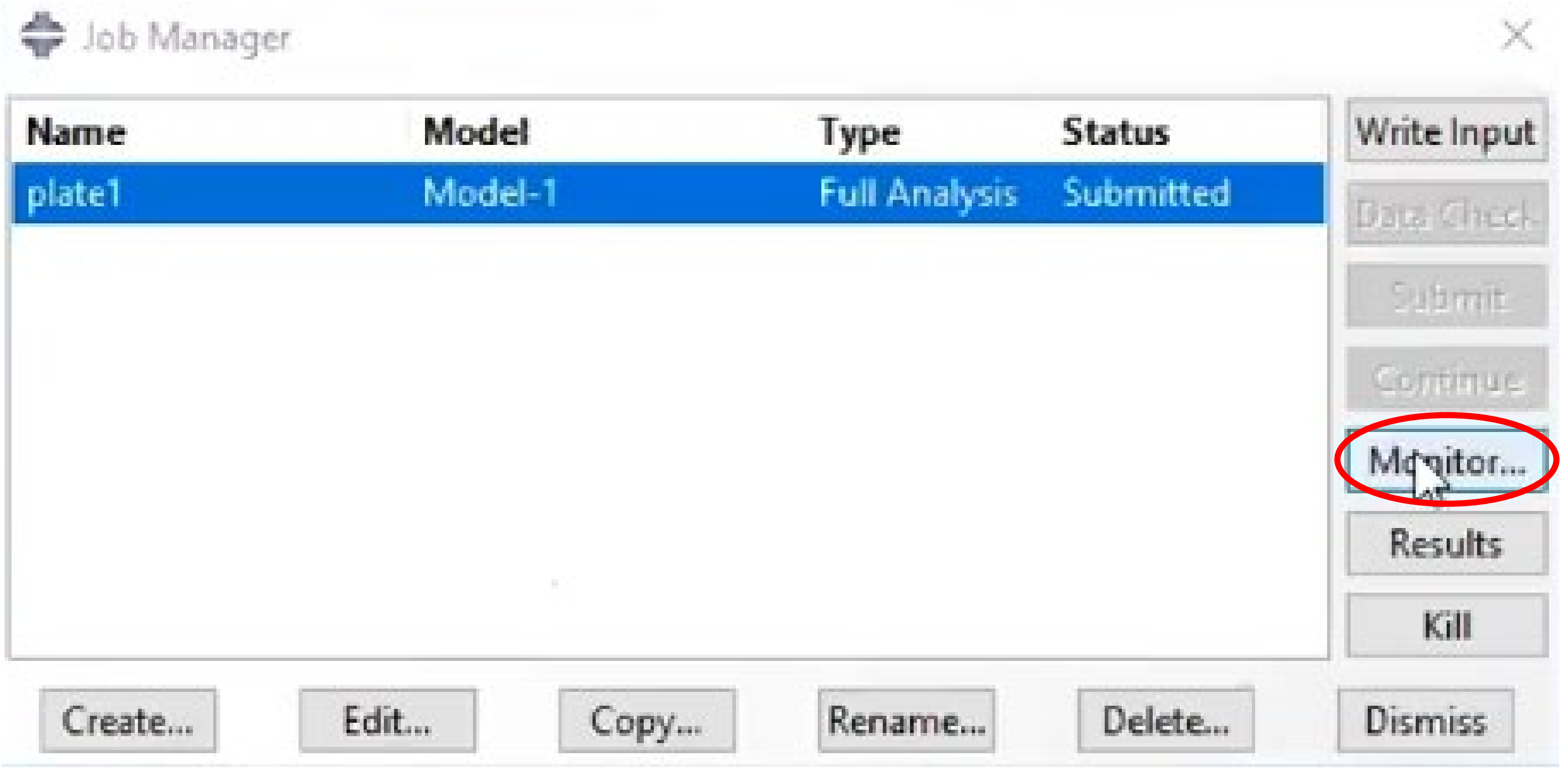
plate1.inp - Notepad

File Edit Format View Help

```
*Elset, elset=__PickedSurf5_S3, internal, instance=p
  62,  87, 228
*Surface, type=ELEMENT, name=_PickedSurf5, internal
__PickedSurf5_S4, S4
__PickedSurf5_S6, S6
__PickedSurf5_S5, S5
__PickedSurf5_S3, S3
*End Assembly
**
** MATERIALS
**
*Material, name=ALU
*Elastic
75000., 0.3
** ----------------------------------------------------------------
**
** STEP: static
**
*Step, name=static
*Static
1., 1., 1e-05, 1.
**
** BOUNDARY CONDITIONS
**
** Name: BC-1 Type: Displacement/Rotation
*Boundary
_PickedSet4, 1, 1
_PickedSet4, 2, 2
_PickedSet4, 3, 3
**
** LOADS
**
** Name: Load-1   Type: Pressure
*Dsload
_PickedSurf5, P, -1.
**
```

Abaqus : module “Job”

- “Monitor” : informations sur le modèle (analyse du fichier *.inp pour détecter d’éventuels problèmes) et sur la résolution (itérations, nombre de degrés de liberté, mémoire, temps de calcul, etc.)



- Le solveur crée un fichier *.odb avec les résultats et informations nécessaires au post-traitement.

Modélisation et simulation par éléments finis

Abaqus : module “Visualization”

- Post-traitement :
  - Reconstruction de la solution à partir des fonctions de forme et des valeurs nodales :

$$\mathbf{u}^h(\mathbf{x}) = \mathbf{H}(\mathbf{x})\,\mathbf{q}$$

  - Calcul des quantités dérivées : par ex. déformations, contraintes, invariants etc.
  - Affichage / extraction.
  - Plus de détails dans les cours “Post-traitement” et “Critères de rupture”.