introduction of **internal variables** to be obtained from:

$$\dot{\boldsymbol{z}} = \arg\inf\left\{\dot{W} + \phi^*\right\}, \qquad W = W(\boldsymbol{\varepsilon}, \boldsymbol{z})$$

time-incremental formulation:

$$\boldsymbol{z}^{\alpha+1} = \arg\inf\left\{W^{\alpha+1}(\boldsymbol{\varepsilon}^{\alpha+1}, \boldsymbol{z}^{\alpha+1}) + \Delta t\, \phi^*\left(\frac{\boldsymbol{z}^{\alpha+1} - \boldsymbol{z}^{\alpha}}{\Delta t}\right)\right\}$$

effective incremental potential:

$$\mathcal{F}_{\boldsymbol{z}^\alpha}(\boldsymbol{\varepsilon}^{\alpha+1}, \boldsymbol{z}^{\alpha+1}) = W^{\alpha+1}(\boldsymbol{\varepsilon}^{\alpha+1}, \boldsymbol{z}^{\alpha+1}) + \Delta t\, \phi^*\left(\frac{\boldsymbol{z}^{\alpha+1} - \boldsymbol{z}^{\alpha}}{\Delta t}\right)$$

condensed energy density (for *local* material models):

$$W^*_{\boldsymbol{z}^\alpha}(\boldsymbol{\varepsilon}^{\alpha+1}) = \mathcal{F}_{\boldsymbol{z}^\alpha}(\boldsymbol{\varepsilon}^{\alpha+1}, \boldsymbol{z}_*^{\alpha+1}), \qquad \boldsymbol{z}_*^{\alpha+1} = \arg\inf \mathcal{F}_{\boldsymbol{z}^\alpha}(\boldsymbol{\varepsilon}^{\alpha+1}, \boldsymbol{z}^{\alpha+1})$$

stress and incremental stiffness tensors:

$$\boldsymbol{\sigma}^{\alpha+1} = \frac{\partial W^*}{\partial \boldsymbol{\varepsilon}^{\alpha+1}}(\boldsymbol{\varepsilon}^{\alpha+1}, \boldsymbol{z}_*^{\alpha+1}) \qquad \mathbb{C}^{\alpha+1}_{ijkl} = \frac{\partial \sigma_{ij}^{\alpha+1}}{\partial \varepsilon_{kl}^{\alpha+1}} + \frac{\partial \sigma_{ij}^{\alpha+1}}{\partial \boldsymbol{z}^{\alpha+1}} \cdot \frac{\partial \boldsymbol{z}_*^{\alpha+1}}{\partial \varepsilon_{kl}^{\alpha+1}}$$

New Material Model Structure

Ae214a: Computational Solid Mechanics

condensed energy density (for *local* material models):

$$W^*_{\boldsymbol{z}^\alpha}(\boldsymbol{\varepsilon}^{\alpha+1}) = \mathcal{F}_{\boldsymbol{z}^\alpha}(\boldsymbol{\varepsilon}^{\alpha+1}, \boldsymbol{z}_*^{\alpha+1}), \qquad \boldsymbol{z}_*^{\alpha+1} = \arg\inf \mathcal{F}_{\boldsymbol{z}^\alpha}(\boldsymbol{\varepsilon}^{\alpha+1}, \boldsymbol{z}^{\alpha+1})$$

stress and incremental stiffness tensors:

$$\boldsymbol{\sigma}^{\alpha+1} = \frac{\partial W^*}{\partial \boldsymbol{\varepsilon}^{\alpha+1}}(\boldsymbol{\varepsilon}^{\alpha+1}, \boldsymbol{z}_*^{\alpha+1}) \qquad \mathbb{C}_{ijkl}^{\alpha+1} = \frac{\partial \sigma_{ij}^{\alpha+1}}{\partial \varepsilon_{kl}^{\alpha+1}} + \frac{\partial \sigma_{ij}^{\alpha+1}}{\partial \boldsymbol{z}^{\alpha+1}} \cdot \frac{\partial \boldsymbol{z}_*^{\alpha+1}}{\partial \varepsilon_{kl}^{\alpha+1}}$$

similar use as before:

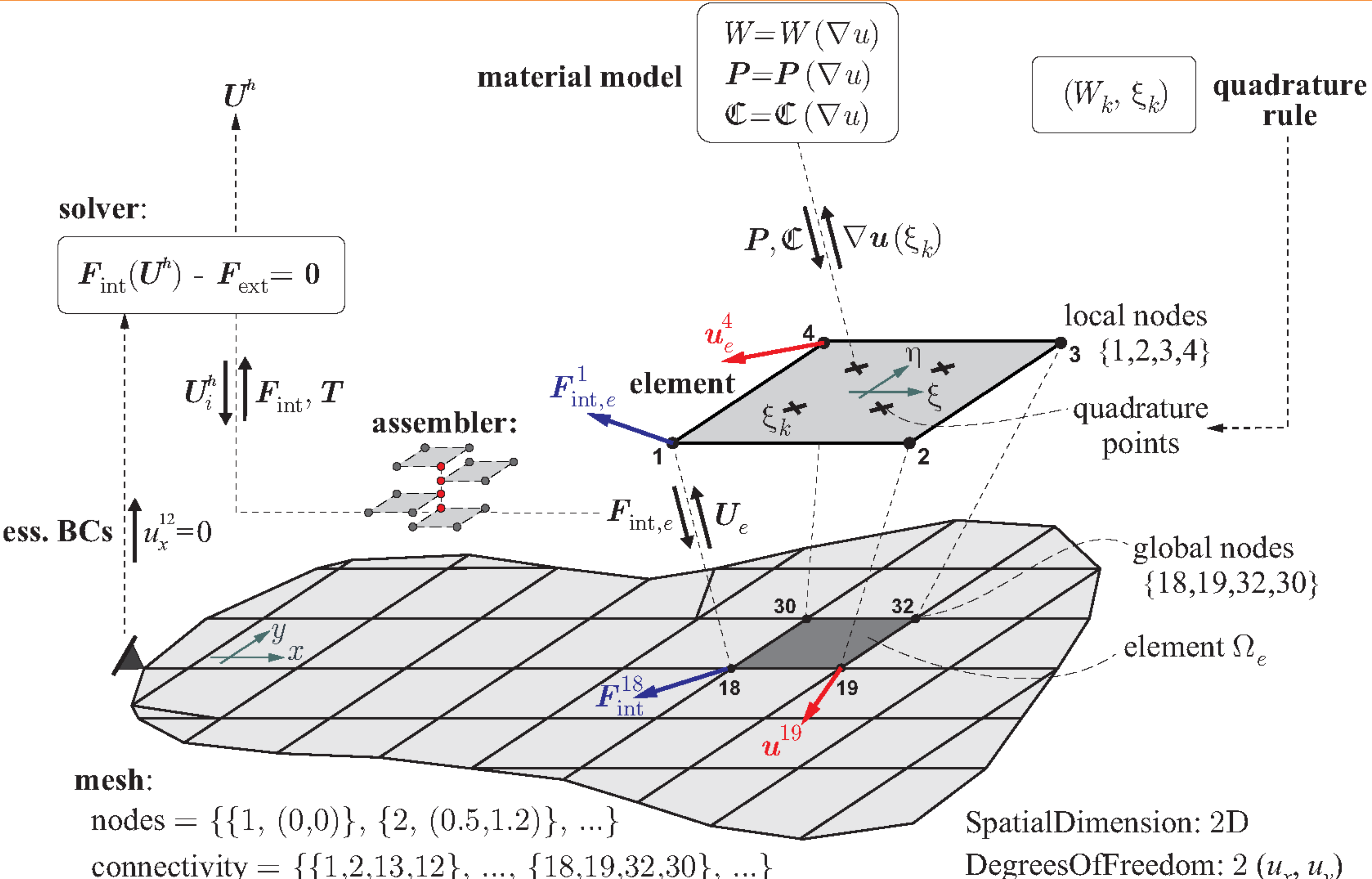
$$I_{\boldsymbol{z}^\alpha}[\boldsymbol{u}^{\alpha+1}, \boldsymbol{z}^{\alpha+1}] = \int_\Omega W^*_{\boldsymbol{z}^\alpha}(\boldsymbol{\varepsilon}^{\alpha+1})\,\mathrm{d}V - \int_\Omega \rho\,\boldsymbol{b}\cdot\boldsymbol{u}^{\alpha+1}\,\mathrm{d}V - \int_{\partial\Omega_N} \hat{\boldsymbol{t}}\cdot\boldsymbol{u}^{\alpha+1}\,\mathrm{d}S$$

new:

- need to introduce and pass internal variables (IVs)
- need to compute and update internal variables
- need updated IVs for computing energy stresses, …

Finite Element Method: Code Overview

Ae214a: Computational Solid Mechanics



an element of any type must compute (at least) the following quantities:

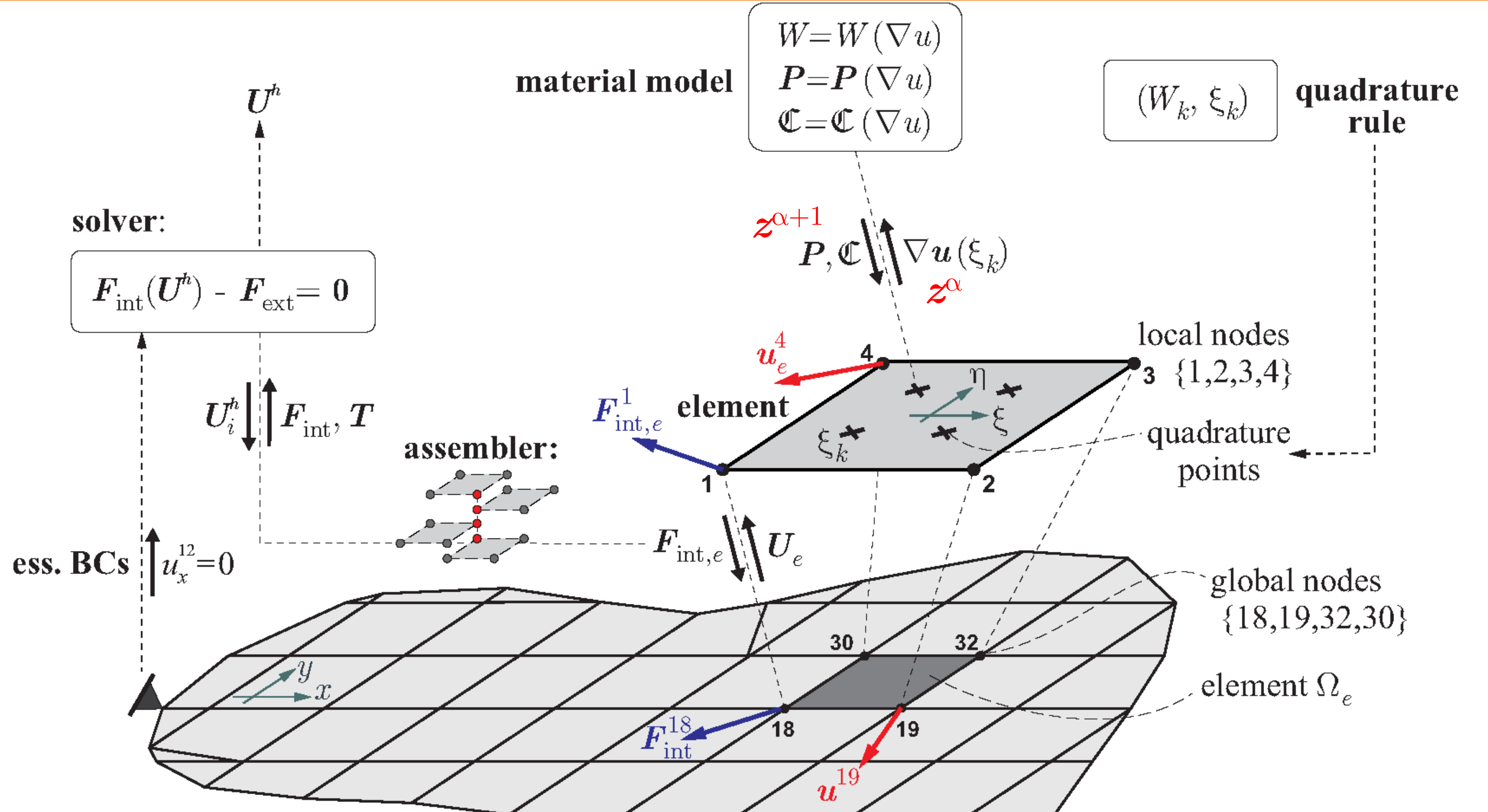
- **element energy**: $$\mathcal{I}^e = \sum_{k=1}^{n_{QP}} W_k\, J(\boldsymbol{\xi}_k) \boxed{W\left(\nabla \boldsymbol{u}(\boldsymbol{\xi}_k), \boldsymbol{z}^\alpha(\boldsymbol{\xi}_k)\right)}\, t$$
- **nodal forces**: $$F^a_{\text{int},i} = \sum_{k=1}^{n_{QP}} W_k\, J(\boldsymbol{\xi}_k) \boxed{P_{iJ}\left(\nabla \boldsymbol{u}(\boldsymbol{\xi}_k)), \boldsymbol{z}^\alpha(\boldsymbol{\xi}_k)\right)}\, N^a_{,J}(\boldsymbol{\xi}_k)\, t$$
- **stiffness matrix**: $$T^{ab}_{ij} = \sum_{k=1}^{n_{QP}} W_k\, J(\boldsymbol{\xi}_k) \boxed{\mathbb{C}_{iJkL}\left(\nabla \boldsymbol{u}(\boldsymbol{\xi}_k)), \boldsymbol{z}^\alpha(\boldsymbol{\xi}_k)\right)}\, N^a_{,J}(\boldsymbol{\xi}_k)\, N^b_{,L}(\boldsymbol{\xi}_k)\, t$$

The element should store the old (previous) IVs for each quadrature point and pass those to the material model when asking for energy, stresses, …

- How does the element know what $\boldsymbol{z}$ looks like?
  → define the type of $\boldsymbol{z}$ **within the MaterialModel** (as a new object)

Finite Element Method: Code Overview

Ae214a: Computational Solid Mechanics



mesh:

nodes = {{1, (0,0)}, {2, (0.5,1.2)}, ...}

connectivity = {{1,2,13,12}, ..., {18,19,32,30}, ...}

SpatialDimension: 2D

DegreesOfFreedom: 2 (u_x, u_y)

New Material Model Structure

Ae214a: Computational Solid Mechanics

`computeEnergy(strains` $\varepsilon^{\alpha+1}$`, oldIVs` z^{α}`, time` $t^{\alpha+1}$`):`

$$W^*_{\boldsymbol{z}^\alpha}(\boldsymbol{\varepsilon}^{\alpha+1}) = \mathcal{F}_{\boldsymbol{z}^\alpha}(\boldsymbol{\varepsilon}^{\alpha+1}, \boxed{\boldsymbol{z}^{\alpha+1}_*})$$

`computeStresses(strains` $\varepsilon^{\alpha+1}$`, oldIVs` z^{α}`, time` $t^{\alpha+1}$`):`

$$\boldsymbol{\sigma}^{\alpha+1} = \frac{\partial W^*}{\partial \boldsymbol{\varepsilon}^{\alpha+1}}(\boldsymbol{\varepsilon}^{\alpha+1}, \boxed{\boldsymbol{z}^{\alpha+1}_*})$$

`computeTangentMatrix(strains` $\varepsilon^{\alpha+1}$`, oldIVs` z^{α}`, time` $t^{\alpha+1}$`):`

$$\mathbb{C}^{\alpha+1}_{ijkl} = \frac{\partial \sigma^{\alpha+1}_{ij}}{\partial \varepsilon^{\alpha+1}_{ij}}(\boxed{\boldsymbol{z}^{\alpha+1}_*}) + \frac{\partial \sigma^{\alpha+1}_{ij}}{\partial \boldsymbol{z}^{\alpha+1}}(\boxed{\boldsymbol{z}^{\alpha+1}_*}) \cdot \frac{\partial \boxed{\boldsymbol{z}^{\alpha+1}_*}}{\partial \varepsilon^{\alpha+1}_{ij}}$$

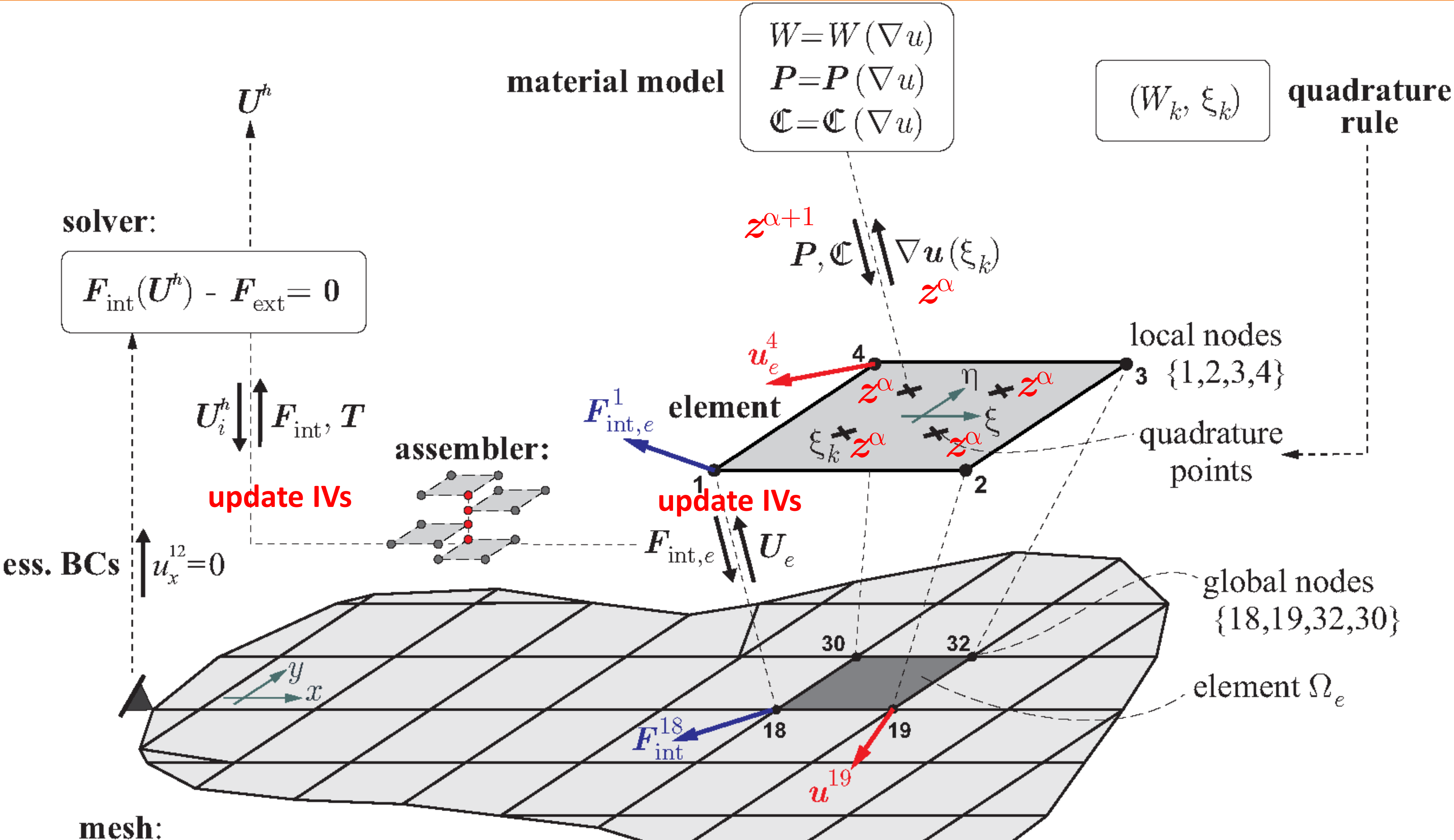
`computeNewIVs(strains` $\varepsilon^{\alpha+1}$`, oldIVs` z^{α}`, time` $t^{\alpha+1}$`):`

$$\boxed{\boldsymbol{z}^{\alpha+1}_*} = \arg\inf \mathcal{F}_{\boldsymbol{z}^\alpha}(\boldsymbol{\varepsilon}^{\alpha+1}, \boldsymbol{z}^{\alpha+1})$$

`class InternalVariables` defined within the MaterialModel

Finite Element Method: Code Overview

Ae214a: Computational Solid Mechanics



nodes = {{1, (0,0)}, {2, (0.5,1.2)}, ...}

connectivity = {{1,2,13,12}, ..., {18,19,32,30}, ...}

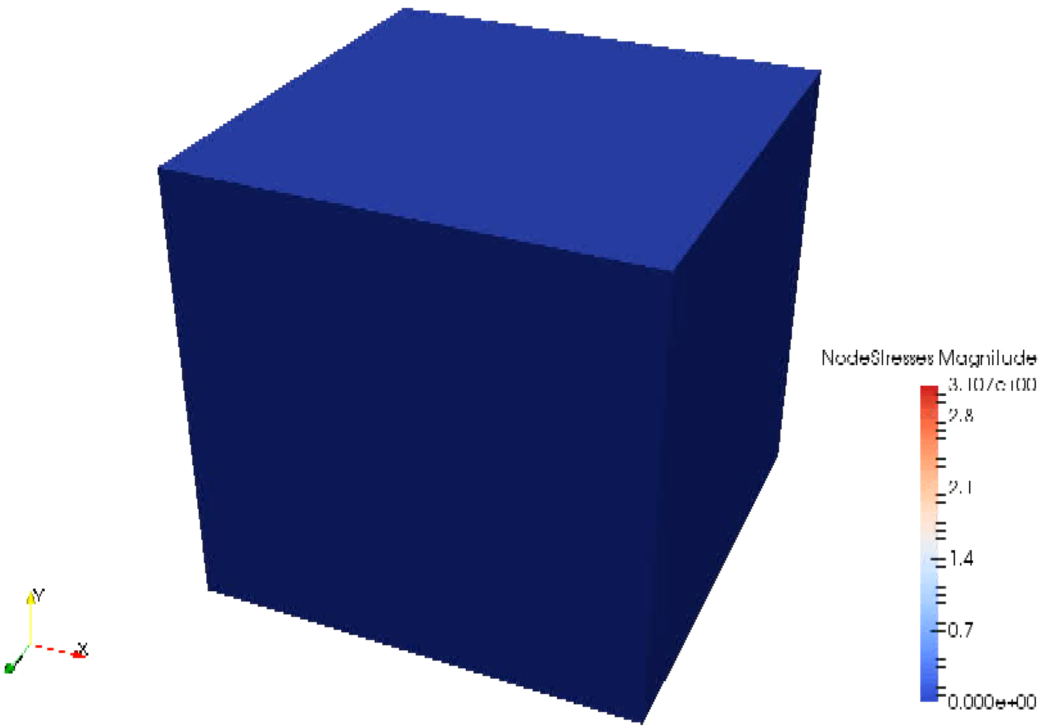
SpatialDimension: 2D

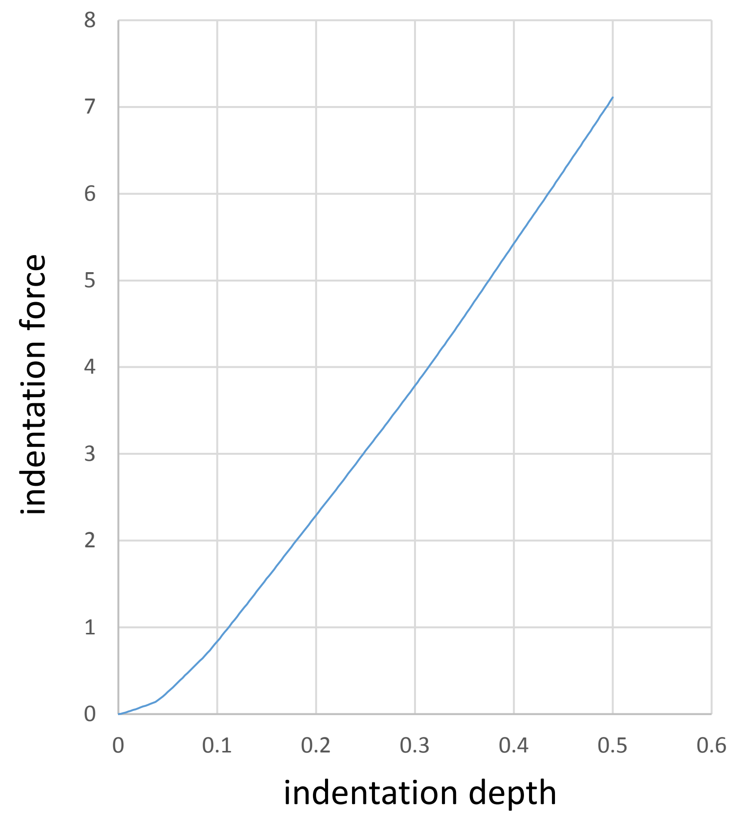
DegreesOfFreedom: 2 (u_x, u_y)

Problem Set #6

Ae214a: Computational Solid Mechanics

1) Indentation with a **linear viscoelastic** (rate-dependent) material model:





Problem Set #6

Ae214a: Computational Solid Mechanics

2) Dynamic wall impact with a **nonlinear elastic** material model: