Nonlinear Finite Element Methods Material nonlinearities

Vinh Phu Nguyen

Sources of nonlinearities

Geometrical nonlinearities

large displacement/rotation (structural instability)

finite deformation: large strain (metal forming)

Material nonlinearities

plasticity

cohesive zone models

damage

visco-plasticity

$$\boldsymbol{\sigma} = f(\boldsymbol{\epsilon}, \boldsymbol{\alpha})$$

Boundary nonlinearities

contact problems

Sources of nonlinearities

Geometrical nonlinearities

large displacement/rotation (structural instability)

finite deformation: large strain (metal forming)

Material nonlinearities

plasticity

cohesive zone models

damage

visco-plasticity

$$\boldsymbol{\sigma} = f(\boldsymbol{\epsilon}, \boldsymbol{\alpha})$$

Boundary nonlinearities

contact problems

References

Nonlinear finite elements for continua and structures, T. Belytschko, W.K. Liu and B. Moran, Wiley, 2000.

Nonlinear finite element methods, P. Wriggers, Springer, 2008.

Non-linear finite element analysis of solids and structures, R. de Borst, M.A. Crisfield, J.J.C. Remmers, C.V. Verhoosel, Wiley, 2012.

Nonlinear continuum mechanics for finite element analysis, J. Bonet and R.D. Wood, Cambridge, 1997.

Material models

Thermodynamics+experiments

Rate independent

(Time independent)

Elastic

Elasto-plastic

Elasto-damage

Plastic-damage

strain rate

Rate dependent

(Time dependent)

Viscoelastic

Viscoplastic

Visco-elasto-damage

Equilibrium equation

$$\int_{\Omega} \delta \boldsymbol{\epsilon}^{\mathrm{T}} \boldsymbol{\sigma} \mathrm{d}\Omega = \int_{\Gamma_t} \delta \mathbf{u}^{\mathrm{T}} \bar{\mathbf{t}} \mathrm{d}\Gamma_t$$

Linear problems

$$\int_{\Omega} \delta \boldsymbol{\epsilon}^{\mathrm{T}} \mathbf{D}^e \boldsymbol{\epsilon} \mathrm{d}\Omega = \int_{\Gamma_t} \delta \mathbf{a}^{\mathrm{T}} \bar{\mathbf{t}} \mathrm{d}\Gamma_t$$

$$\int_{\Omega} \delta \mathbf{a}^{\mathrm{T}} \mathbf{B}^{\mathrm{T}} \mathbf{D}^e \mathbf{B} \mathbf{a} \mathrm{d}\Omega = \int_{\Gamma_t} \delta \mathbf{a}^{\mathrm{T}} \mathbf{N}^{\mathrm{T}} \bar{\mathbf{t}} \mathrm{d}\Gamma_t$$

$$\left(\int_{\Omega} \mathbf{B}^{\mathrm{T}} \mathbf{D}^e \mathbf{B} \mathrm{d}\Omega \right) \mathbf{a} = \int_{\Gamma_t} \mathbf{N}^{\mathrm{T}} \bar{\mathbf{t}} \mathrm{d}\Gamma_t$$

Equilibrium equation

$$\int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{T}}\boldsymbol{\sigma}\mathrm{d}\Omega = \int_{\Gamma_t} \delta\mathbf{u}^{\mathrm{T}}\bar{\mathbf{t}}\mathrm{d}\Gamma_t$$

Linear problems

Nonlinear problems

$$\int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{T}}\mathbf{D}^e\boldsymbol{\epsilon}\mathrm{d}\Omega = \int_{\Gamma_t} \delta\mathbf{a}^{\mathrm{T}}\bar{\mathbf{t}}\mathrm{d}\Gamma_t$$

$$\int_\Omega \delta\mathbf{a}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{D}^e\mathbf{B}\mathbf{a}\mathrm{d}\Omega = \int_{\Gamma_t} \delta\mathbf{a}^{\mathrm{T}}\mathbf{N}^{\mathrm{T}}\bar{\mathbf{t}}\mathrm{d}\Gamma_t$$

$$\left(\int_\Omega \mathbf{B}^{\mathrm{T}}\mathbf{D}^e\mathbf{B}\mathrm{d}\Omega\right)\mathbf{a} = \int_{\Gamma_t} \mathbf{N}^{\mathrm{T}}\bar{\mathbf{t}}\mathrm{d}\Gamma_t$$

Equilibrium equation

$$\int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{T}}\boldsymbol{\sigma}\,\mathrm{d}\Omega = \int_{\Gamma_t} \delta\mathbf{u}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma_t$$

Linear problems

Nonlinear problems

$$\int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{T}}\mathbf{D}^e\boldsymbol{\epsilon}\,\mathrm{d}\Omega = \int_{\Gamma_t} \delta\mathbf{a}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma_t$$

$$\int_\Omega \delta\mathbf{a}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{D}^e\mathbf{B}\mathbf{a}\,\mathrm{d}\Omega = \int_{\Gamma_t} \delta\mathbf{a}^{\mathrm{T}}\mathbf{N}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma_t$$

$$\left(\int_\Omega \mathbf{B}^{\mathrm{T}}\mathbf{D}^e\mathbf{B}\,\mathrm{d}\Omega\right)\mathbf{a} = \int_{\Gamma_t} \mathbf{N}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma_t$$

$$\int_\Omega \mathbf{B}^{\mathrm{T}}\boldsymbol{\sigma}(\mathbf{a})\,\mathrm{d}\Omega = \int_{\Gamma_t} \mathbf{N}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma_t$$

Equilibrium equation

Discrete equilibrium equation

$$\mathbf{f}_{\text{int}}(\mathbf{a}) = \mathbf{f}_{\text{ext}}$$

with

$$\mathbf{f}_{\text{int}}(\mathbf{a}) = \int_{\Omega} \mathbf{B}^{\mathrm{T}} \boldsymbol{\sigma} \mathrm{d}\Omega$$

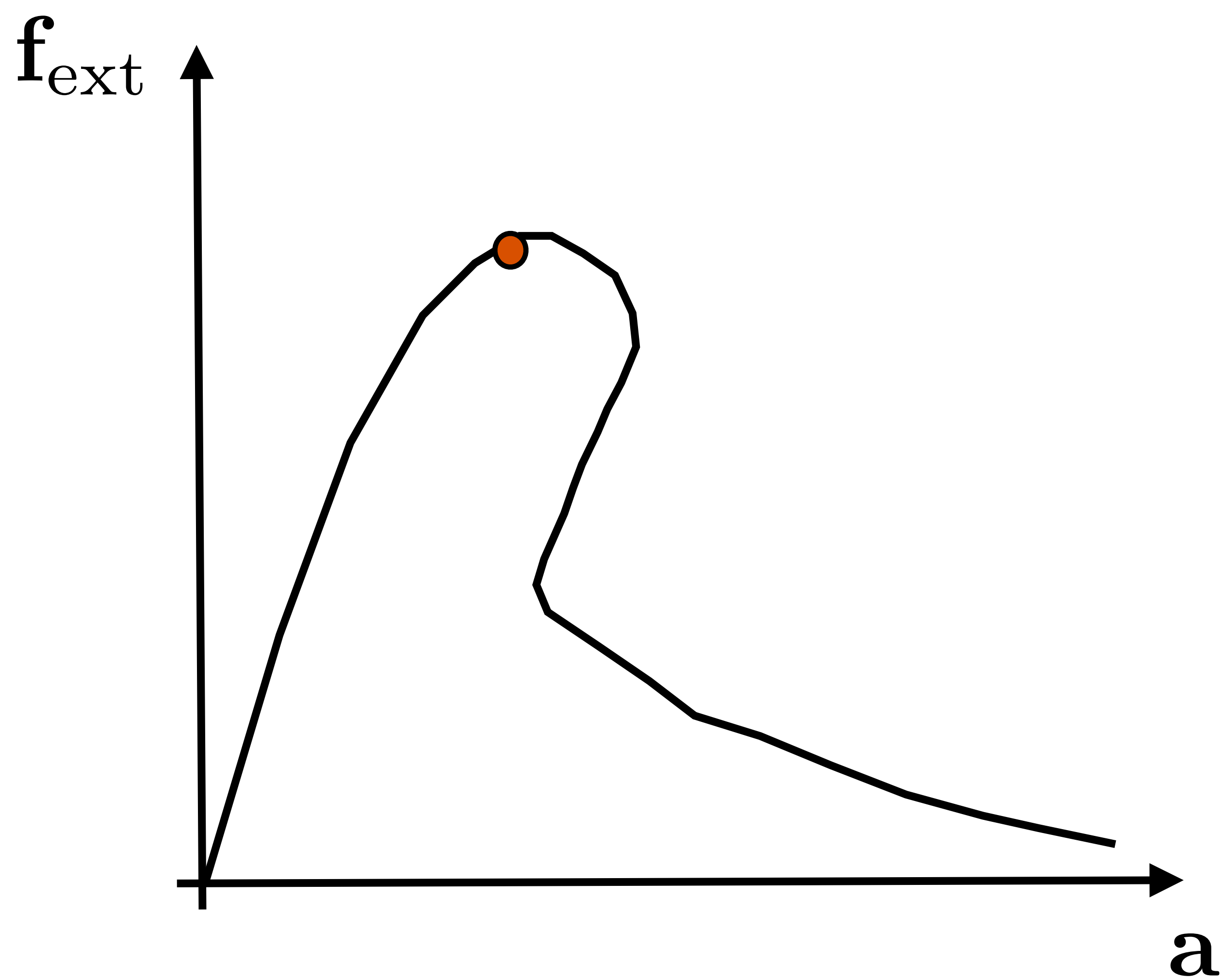
$$\mathbf{f}_{\text{ext}} = \int_{\Gamma_t} \mathbf{N}^{\mathrm{T}} \bar{\mathbf{t}} \mathrm{d}\Gamma_t$$

$$\boldsymbol{\epsilon} = \mathbf{B}\mathbf{a}$$

$$\boldsymbol{\sigma} = f(\boldsymbol{\epsilon}, \alpha)$$

history variables

equilibrium path/curve



Argyris (1964)

Equilibrium equation

Discrete equilibrium equation

$$\mathbf{f}_{\text{int}}(\mathbf{a}) = \mathbf{f}_{\text{ext}}$$

with

$$\mathbf{f}_{\text{int}}(\mathbf{a}) = \int_\Omega \mathbf{B}^{\mathrm{T}} \boldsymbol{\sigma} \mathrm{d}\Omega$$

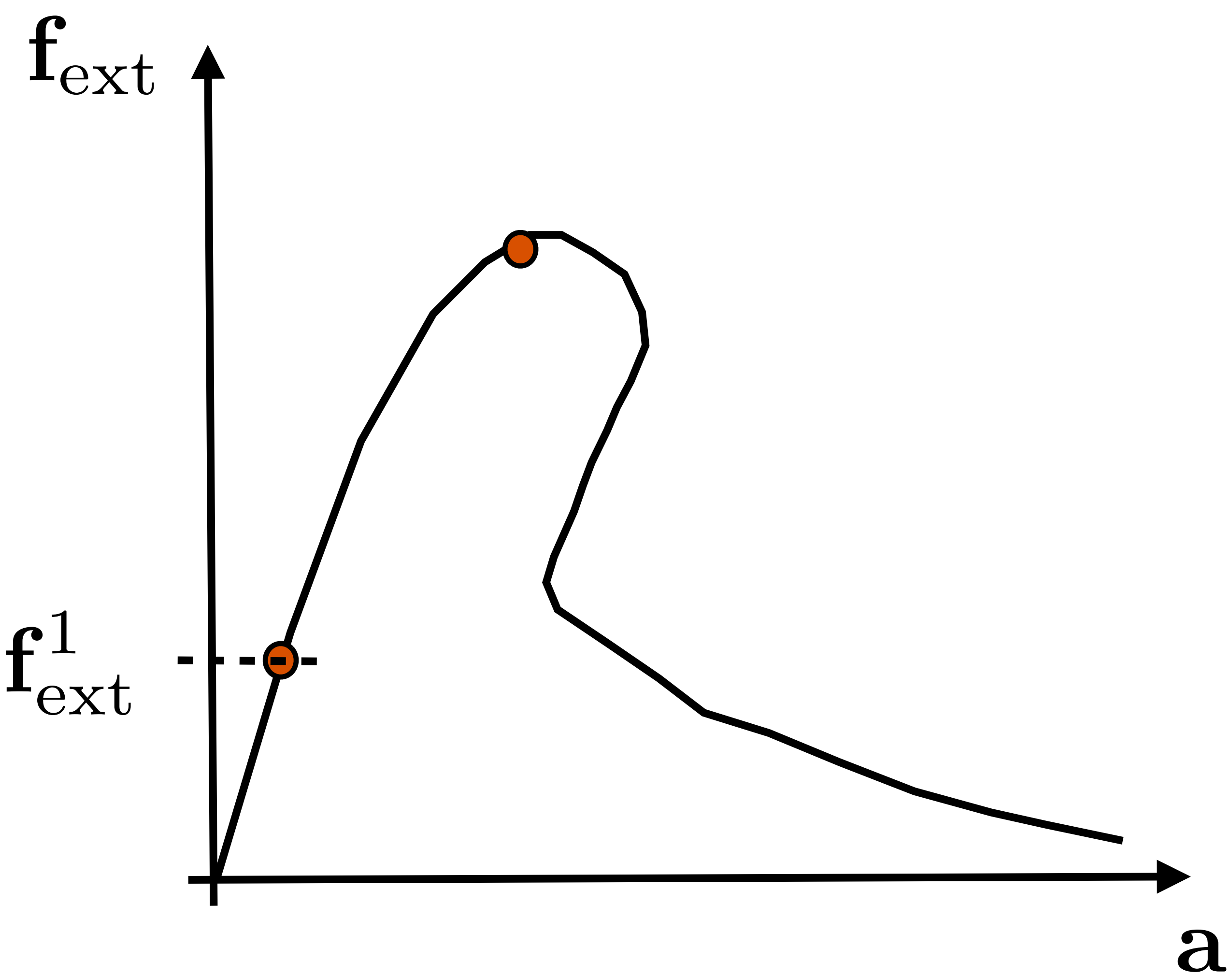
$$\mathbf{f}_{\text{ext}} = \int_{\Gamma_t} \mathbf{N}^{\mathrm{T}} \bar{\mathbf{t}} \mathrm{d}\Gamma_t$$

$$\boldsymbol{\epsilon} = \mathbf{B}\mathbf{a}$$

$$\boldsymbol{\sigma} = f(\boldsymbol{\epsilon}, \alpha)$$

history variables

equilibrium path/curve



Argyris (1964)

Equilibrium equation

Discrete equilibrium equation

$$\mathbf{f}_{\text{int}}(\mathbf{a}) = \mathbf{f}_{\text{ext}}$$

with

$$\mathbf{f}_{\text{int}}(\mathbf{a}) = \int_{\Omega} \mathbf{B}^{\mathrm{T}} \boldsymbol{\sigma} \mathrm{d}\Omega$$

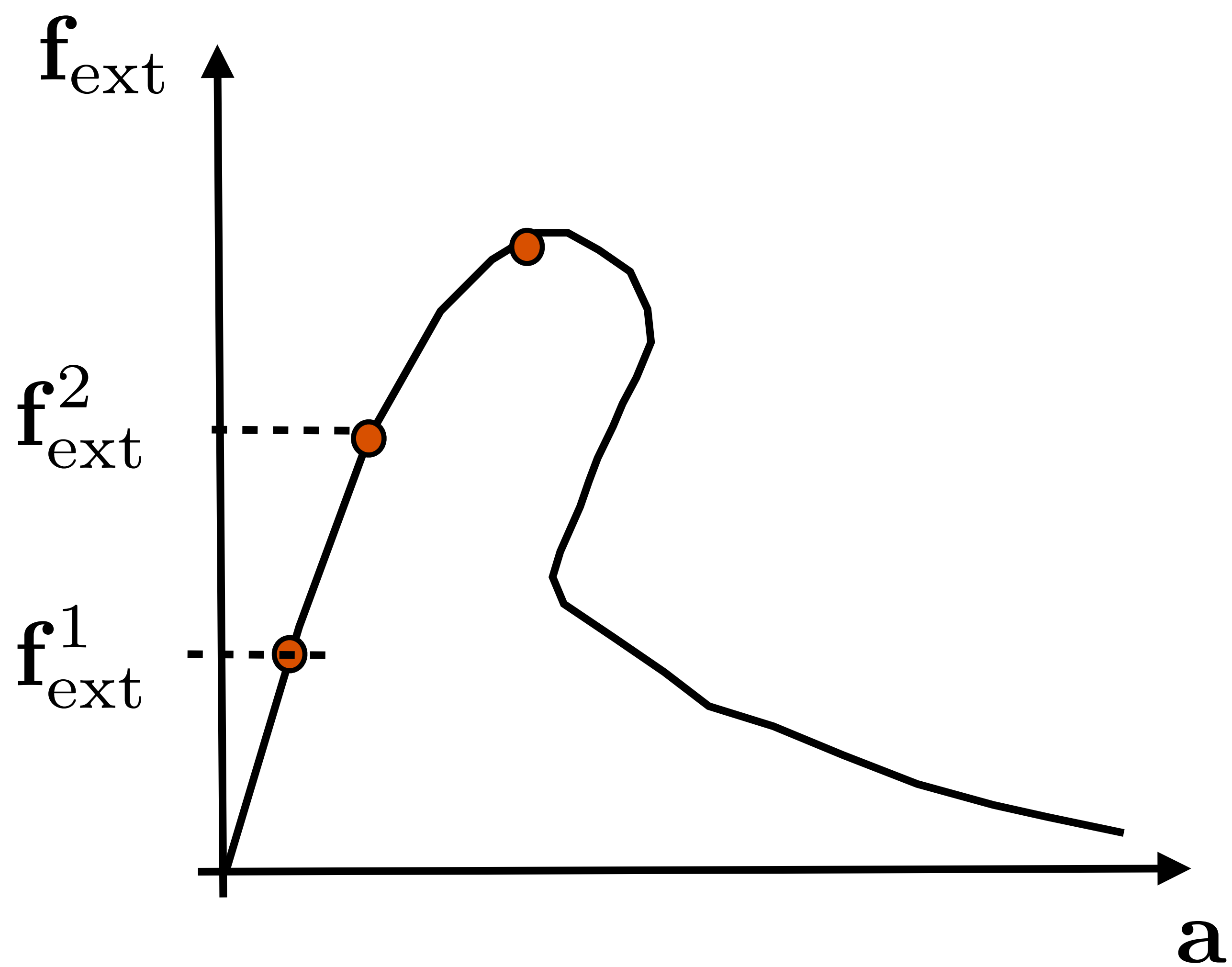
$$\mathbf{f}_{\text{ext}} = \int_{\Gamma_t} \mathbf{N}^{\mathrm{T}} \bar{\mathbf{t}} \mathrm{d}\Gamma_t$$

$$\boldsymbol{\epsilon} = \mathbf{B}\mathbf{a}$$

$$\boldsymbol{\sigma} = f(\boldsymbol{\epsilon}, \alpha)$$

history variables

equilibrium path/curve



Argyris (1964)

Equilibrium equation

Discrete equilibrium equation

$$\mathbf{f}_{\text{int}}(\mathbf{a}) = \mathbf{f}_{\text{ext}}$$

with

$$\mathbf{f}_{\text{int}}(\mathbf{a}) = \int_\Omega \mathbf{B}^{\text{T}} \boldsymbol{\sigma} \text{d}\Omega$$

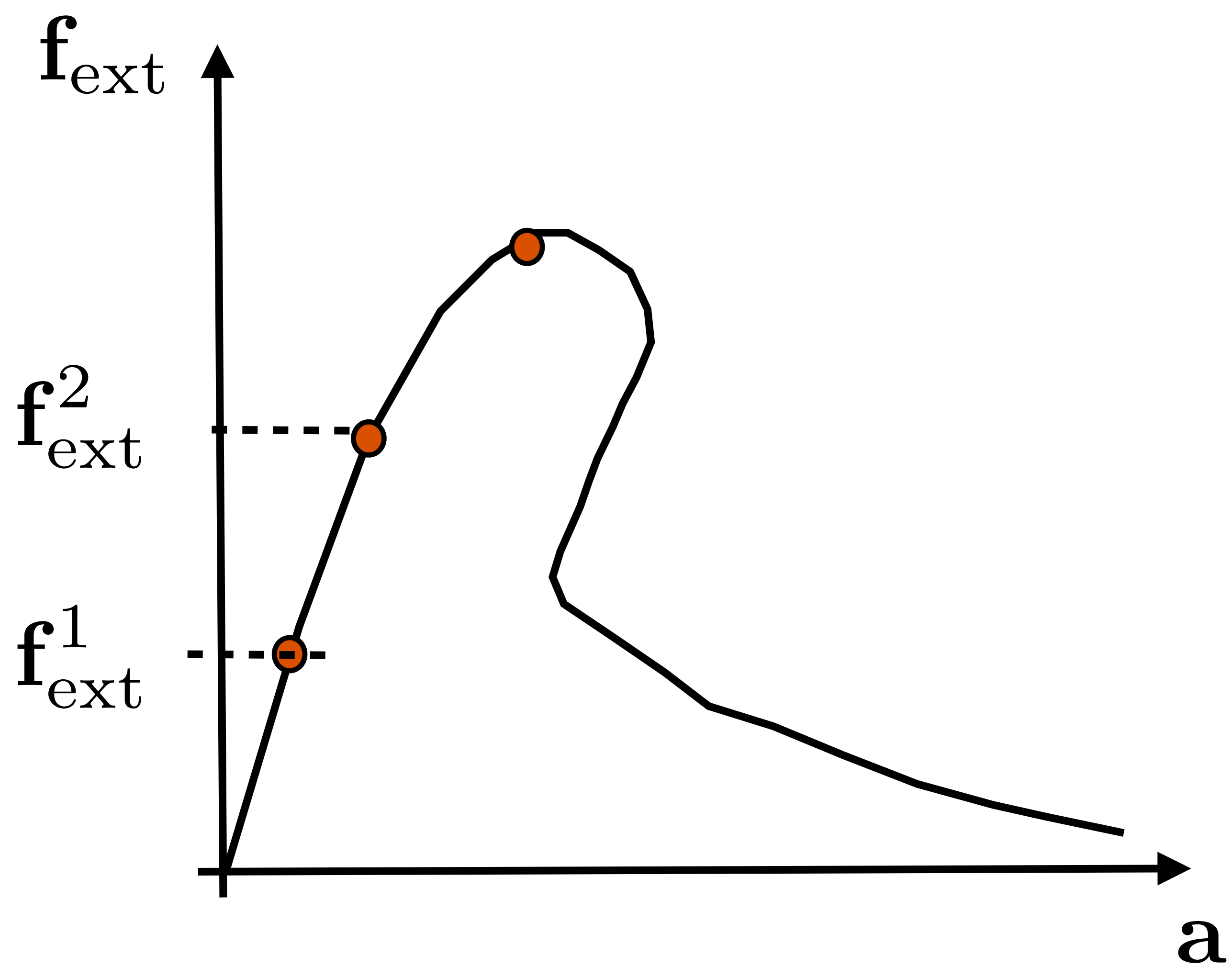
$$\mathbf{f}_{\text{ext}} = \int_{\Gamma_t} \mathbf{N}^{\text{T}} \bar{\mathbf{t}} \text{d}\Gamma_t$$

$$\boldsymbol{\epsilon} = \mathbf{B}\mathbf{a}$$

$$\boldsymbol{\sigma} = f(\boldsymbol{\epsilon}, \alpha)$$

history variables

equilibrium path/curve



Incremental

Argyris (1964)

Equilibrium equation

Discrete equilibrium equation

$$\mathbf{f}_{\text{int}}(\mathbf{a}) = \mathbf{f}_{\text{ext}}$$

with

$$\mathbf{f}_{\text{int}}(\mathbf{a}) = \int_{\Omega} \mathbf{B}^{\text{T}} \boldsymbol{\sigma} \mathrm{d}\Omega$$

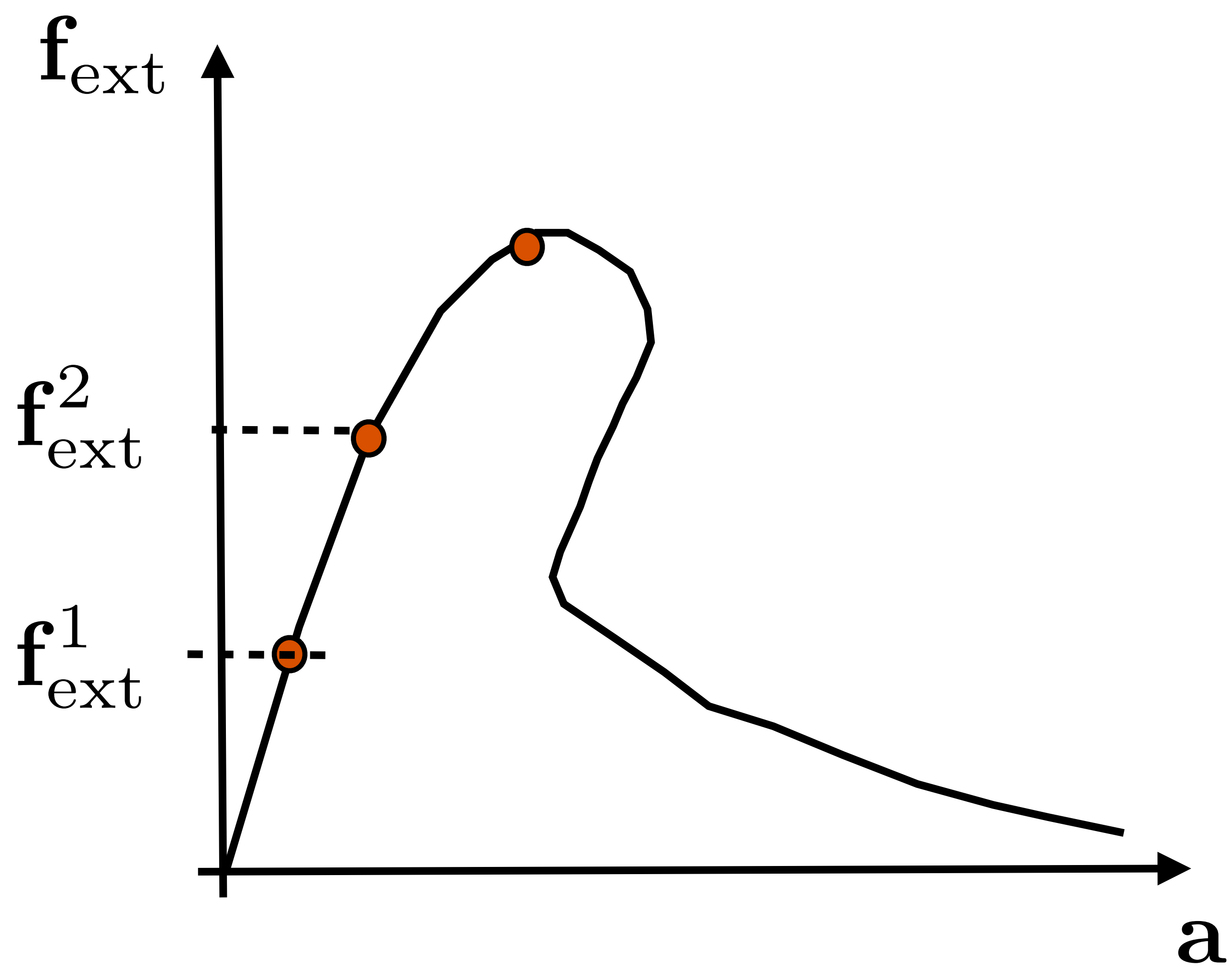
$$\mathbf{f}_{\text{ext}} = \int_{\Gamma_t} \mathbf{N}^{\text{T}} \bar{\mathbf{t}} \mathrm{d}\Gamma_t$$

$$\boldsymbol{\epsilon} = \mathbf{B}\mathbf{a}$$

$$\boldsymbol{\sigma} = f(\boldsymbol{\epsilon}, \alpha)$$

history variables

equilibrium path/curve

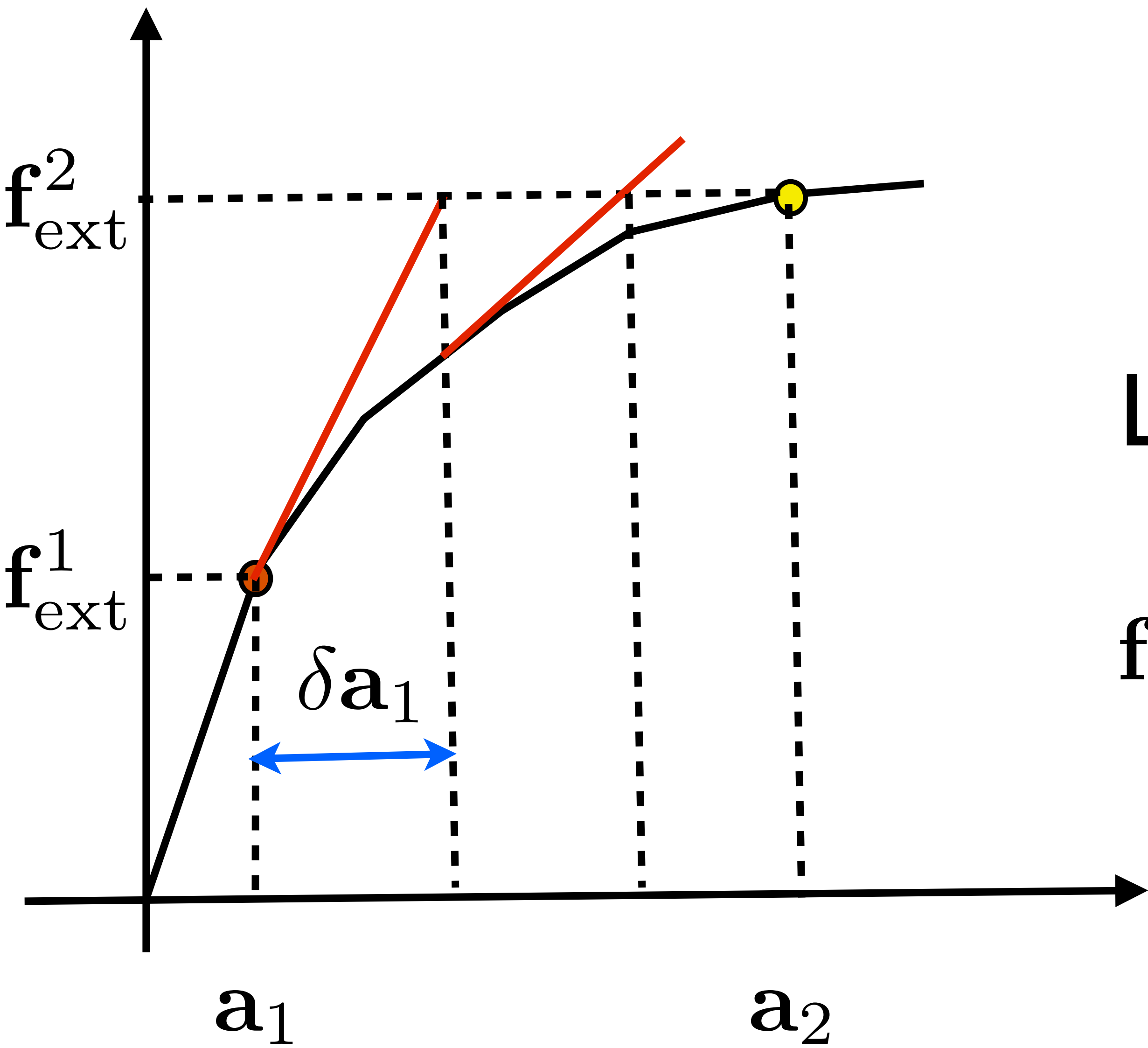


Incremental iterative

Argyris (1964)

Newton-Raphson methods

Full NR scheme



$$\mathbf{f}_{\text{int}}(\mathbf{a}_2) - \mathbf{f}_{\text{ext}}^2 = \mathbf{0}$$

Linearization around $\mathbf{a}_1$

$$\mathbf{f}_{\text{int}}^n(\mathbf{a}_1) - \mathbf{f}_{\text{ext}}^2 + \frac{\partial \mathbf{f}_{\text{int}}^n}{\partial \mathbf{a}}\Big|_{\mathbf{a}_1} \delta\mathbf{a} = \mathbf{0}$$

$$\mathbf{K}(\mathbf{a}_1)\delta\mathbf{a}_1 = \mathbf{f}_{\text{ext}}^n - \mathbf{f}_{\text{int}}^n(\mathbf{a}_1)$$

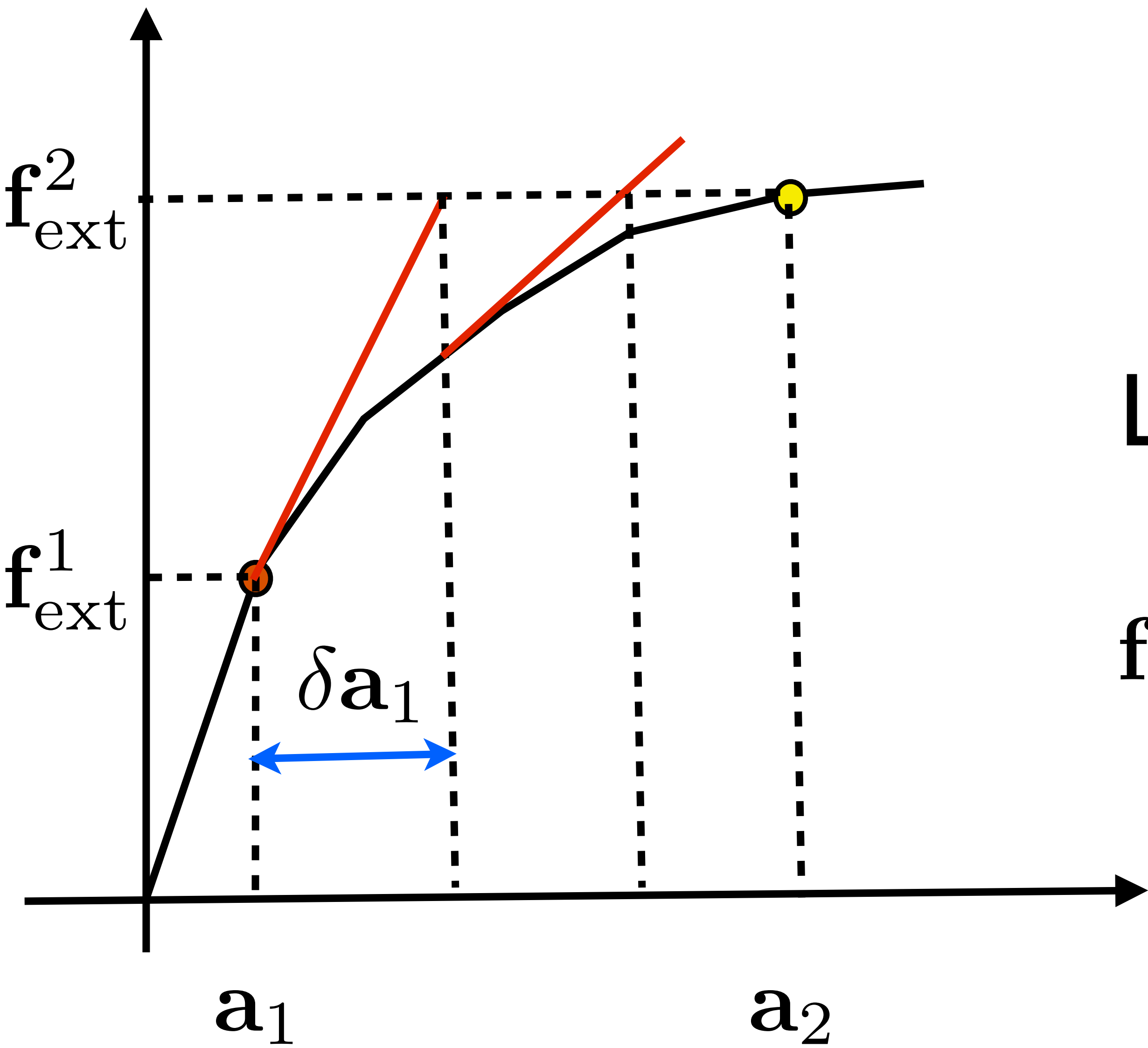
$$\mathbf{a}_1 \leftarrow \delta\mathbf{a}_1$$

Tangent stiffness matrix

$$\mathbf{K} = \frac{\partial \mathbf{f}_{\text{int}}}{\partial \mathbf{a}}$$

Newton-Raphson methods

Full NR scheme



$$\mathbf{f}_{\text{int}}(\mathbf{a}_2) - \mathbf{f}_{\text{ext}}^2 = \mathbf{0}$$

Linearization around $\mathbf{a}_1$

$$\mathbf{f}_{\text{int}}^n(\mathbf{a}_1) - \mathbf{f}_{\text{ext}}^2 + \frac{\partial \mathbf{f}_{\text{int}}^n}{\partial \mathbf{a}}|_{\mathbf{a}_1} \delta\mathbf{a} = \mathbf{0}$$

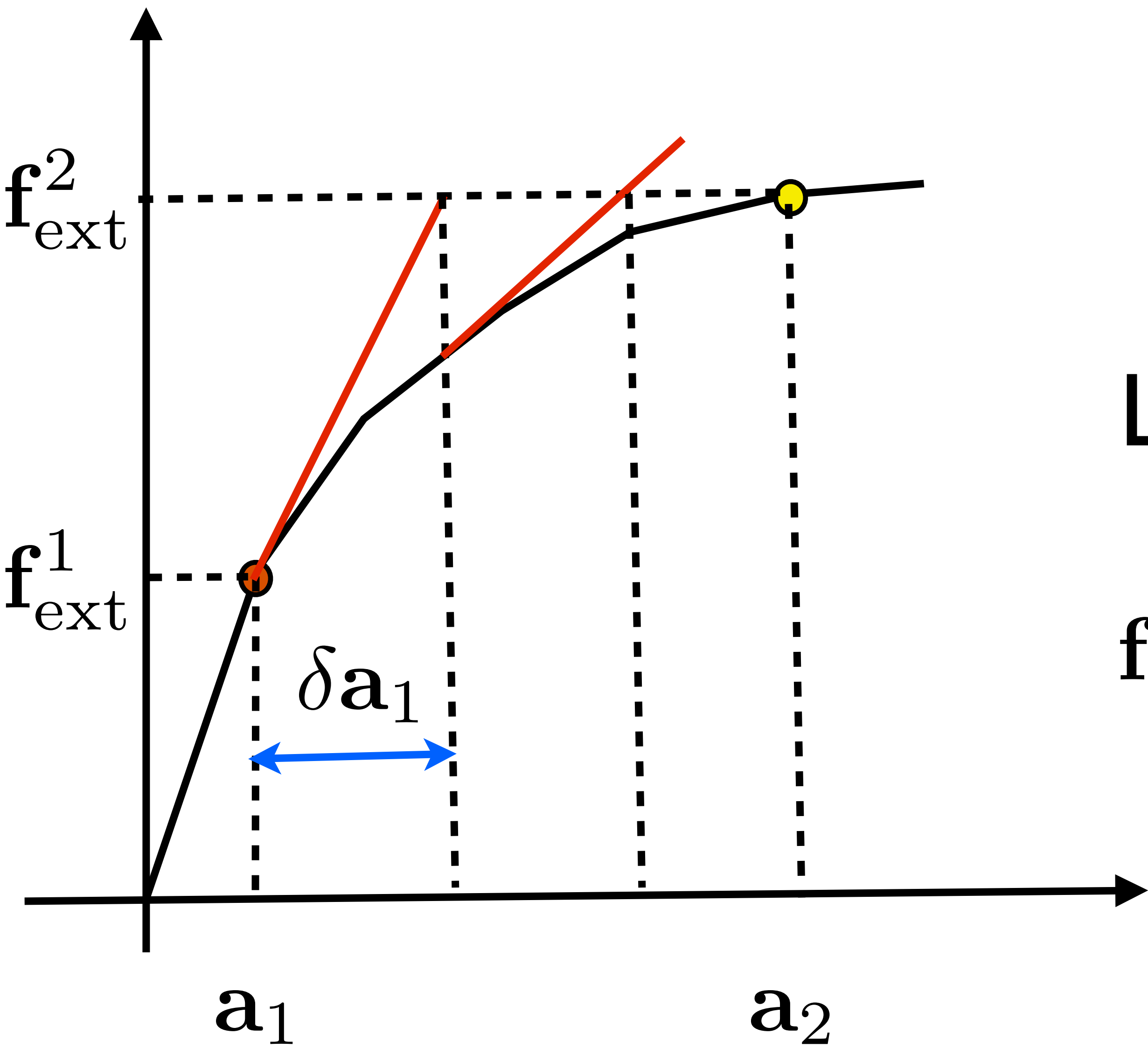
$$\mathbf{K}(\mathbf{a}_1)\delta\mathbf{a}_1 = \mathbf{f}_{\text{ext}}^n - \mathbf{f}_{\text{int}}^n(\mathbf{a}_1)$$

$$\mathbf{a}_1 \leftarrow \delta\mathbf{a}_1$$

Tangent stiffness matrix
$$\mathbf{K} = \frac{\partial \mathbf{f}_{\text{int}}}{\partial \mathbf{a}}$$

Newton-Raphson methods

Full NR scheme



$$\mathbf{f}_{\text{int}}(\mathbf{a}_2) - \mathbf{f}_{\text{ext}}^2 = \mathbf{0}$$

Linearization around $\mathbf{a}_1$

$$\mathbf{f}_{\text{int}}^n(\mathbf{a}_1) - \mathbf{f}_{\text{ext}}^2 + \frac{\partial \mathbf{f}_{\text{int}}^n}{\partial \mathbf{a}}\Big|_{\mathbf{a}_1} \delta\mathbf{a} = \mathbf{0}$$

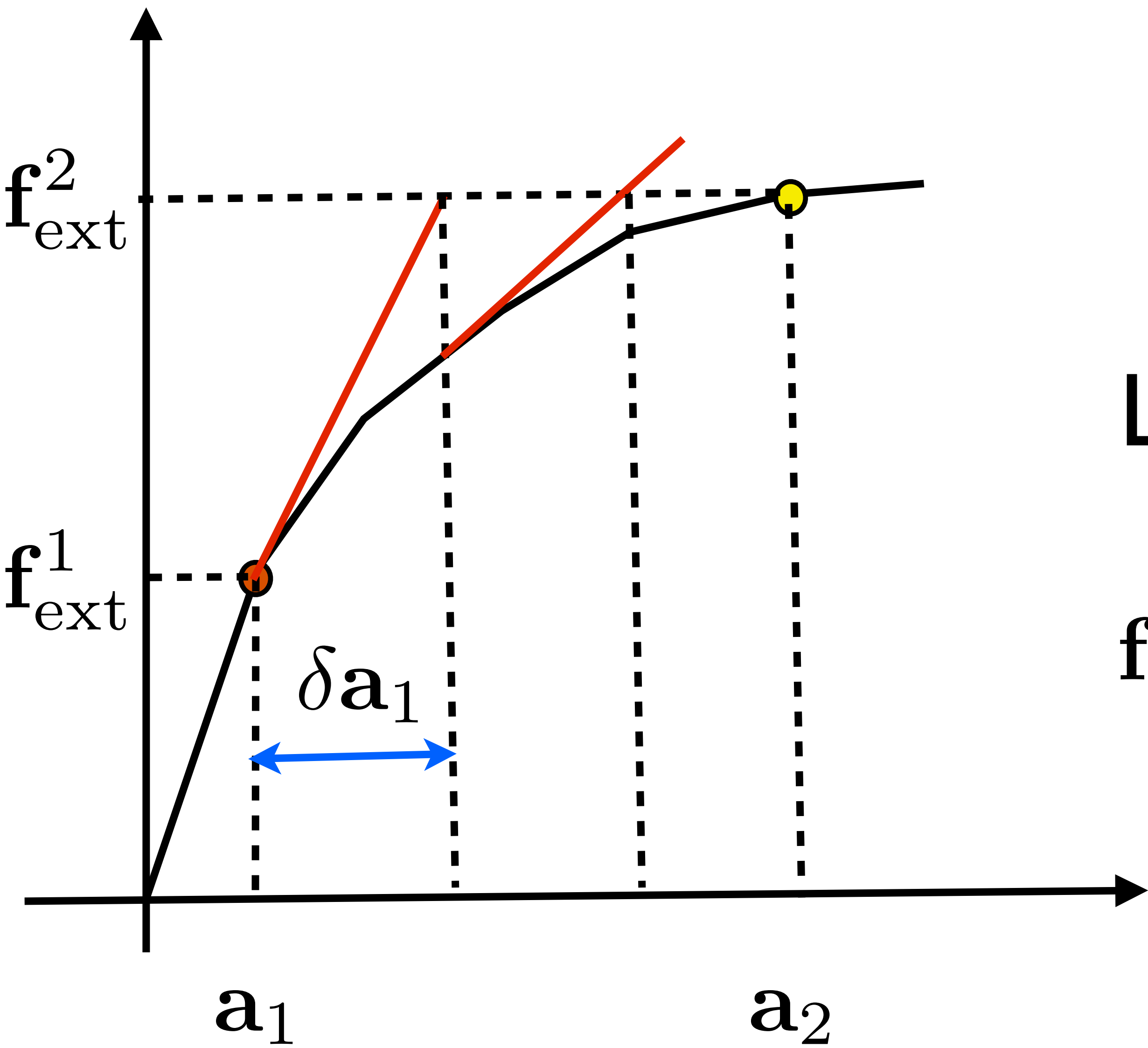
$$\mathbf{K}(\mathbf{a}_1)\delta\mathbf{a}_1 = \mathbf{f}_{\text{ext}}^n - \mathbf{f}_{\text{int}}^n(\mathbf{a}_1)$$

$$\mathbf{a}_1 \leftarrow \delta\mathbf{a}_1 \qquad \mathbf{r}$$

Tangent stiffness matrix $\mathbf{K} = \dfrac{\partial \mathbf{f}_{\text{int}}}{\partial \mathbf{a}}$

Newton-Raphson methods

Full NR scheme



$$\mathbf{f}_{\text{int}}(\mathbf{a}_2) - \mathbf{f}_{\text{ext}}^2 = \mathbf{0}$$

Linearization around $\mathbf{a}_1$

$$\mathbf{f}_{\text{int}}^n(\mathbf{a}_1) - \mathbf{f}_{\text{ext}}^2 + \frac{\partial \mathbf{f}_{\text{int}}^n}{\partial \mathbf{a}}\Big|_{\mathbf{a}_1} \delta\mathbf{a} = \mathbf{0}$$

$$\mathbf{K}(\mathbf{a}_1)\delta\mathbf{a}_1 = \mathbf{f}_{\text{ext}}^n - \mathbf{f}_{\text{int}}^n(\mathbf{a}_1)$$

$$\mathbf{a}_1 \leftarrow \delta\mathbf{a}_1 \qquad \mathbf{r}$$

Tangent stiffness matrix $\mathbf{K} = \dfrac{\partial \mathbf{f}_{\text{int}}}{\partial \mathbf{a}}$ residual/out of balance vector

Tangent stiffness matrix

$$\mathbf{f}_{\text{int}}(\mathbf{a}) = \int_\Omega \mathbf{B}^{\text{T}} \boldsymbol{\sigma} \text{d}\Omega$$

$$\mathbf{K} = \frac{\partial \mathbf{f}_{\text{int}}^n}{\partial \mathbf{a}}$$

$$\mathbf{f}_{\text{int}}(\mathbf{a}) = \int_\Omega \mathbf{B}^{\text{T}} \boldsymbol{\sigma}[\boldsymbol{\epsilon}(\mathbf{a})] \text{d}\Omega$$

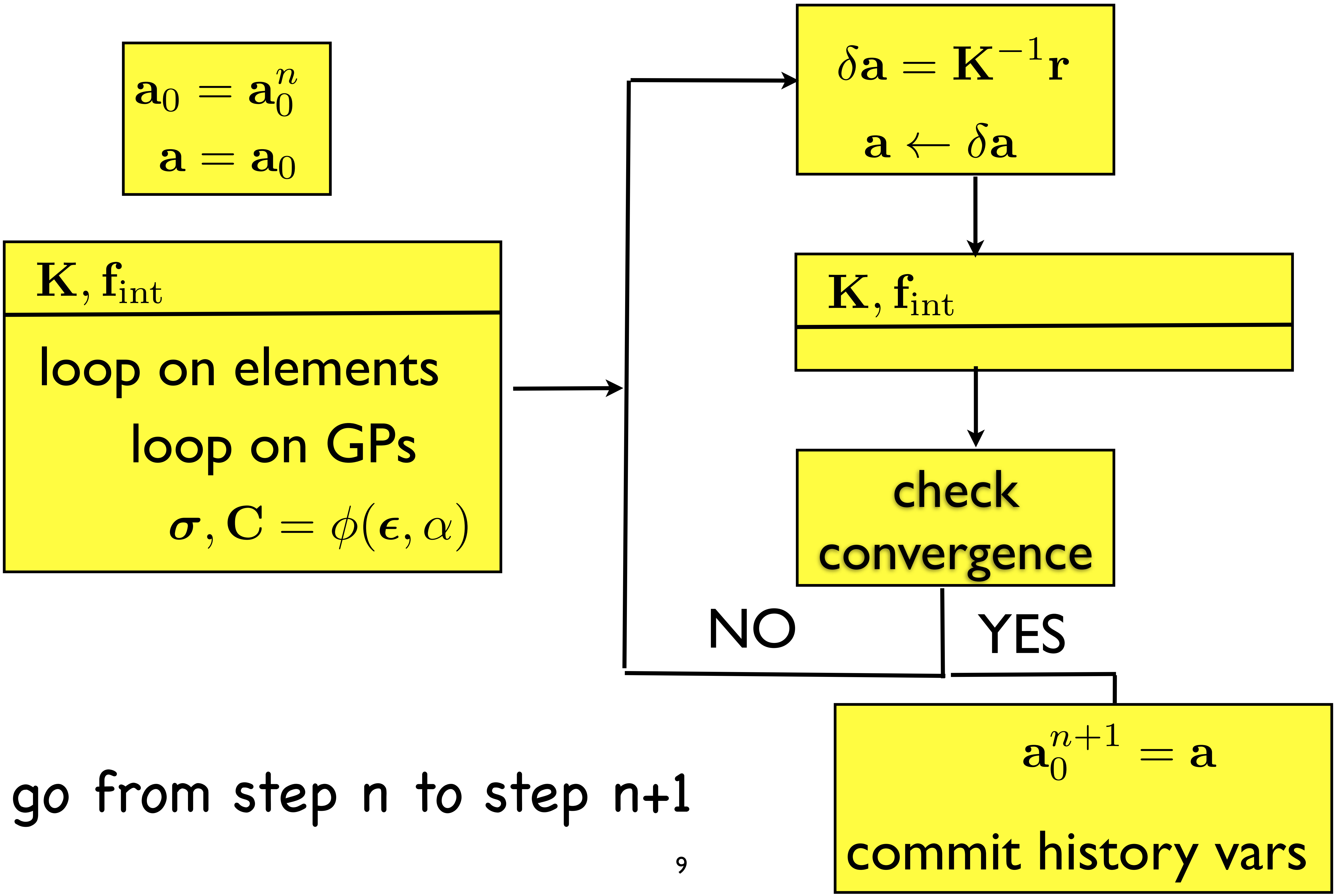
$$\mathbf{K} = \int_\Omega \mathbf{B}^{\text{T}} \frac{\partial \boldsymbol{\sigma}[\boldsymbol{\epsilon}(\mathbf{a})]}{\partial \mathbf{a}} \text{d}\Omega$$

$$\boxed{\text{d}\boldsymbol{\sigma} = \mathbf{C}\text{d}\boldsymbol{\epsilon}}$$

$$\boxed{\mathbf{K} = \int_\Omega \mathbf{B}^{\text{T}} \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\epsilon}} \mathbf{B} \text{d}\Omega}$$

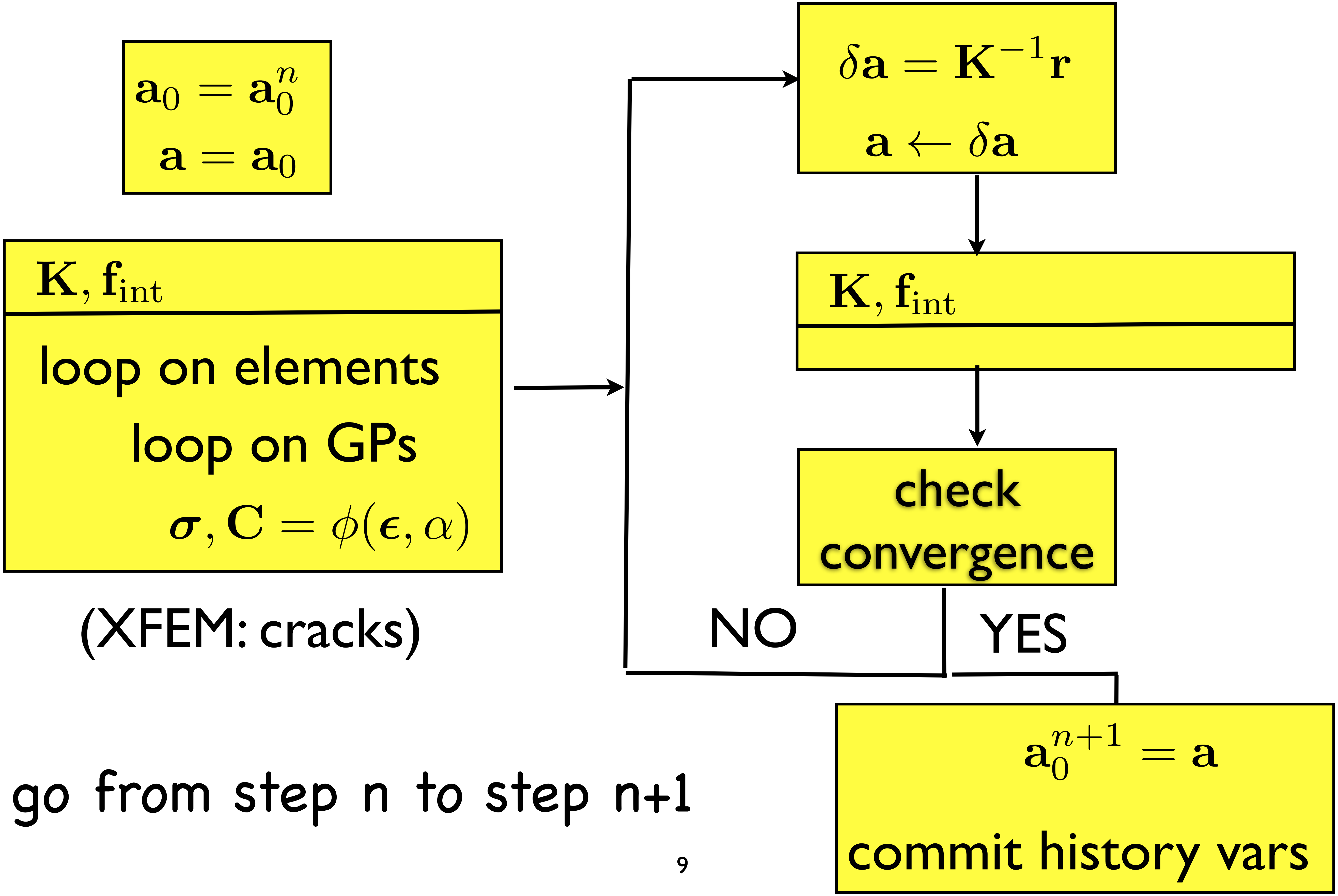
$$\boxed{\mathbf{K} = \int_\Omega \mathbf{B}^{\text{T}} \mathbf{C} \mathbf{B} \text{d}\Omega}$$

Flowchart



go from step n to step n+1

Flowchart



go from step n to step n+1

At integration level

For one Gauss point, do

1. compute displacement **a**
2. compute strains ε, get history α
3. compute stresses/material tangent
4. compute internal force vector
5. compute tangent stiffness

$$\mathbf{a} = \mathbf{N}\mathbf{a}_e$$

$$\boldsymbol{\epsilon} = \mathbf{B}\mathbf{a}$$

$$\boldsymbol{\sigma} = f(\boldsymbol{\epsilon}, \alpha) \qquad \mathbf{C}$$

$$\mathbf{f}_{\text{int}} = \int_\Omega \mathbf{B}^{\text{T}} \boldsymbol{\sigma} \mathrm{d}\Omega$$

$$\mathbf{K} \leftarrow \int_\Omega \mathbf{B}^{\text{T}} \mathbf{C} \mathbf{B} \mathrm{d}\Omega$$

At integration level

For one Gauss point, do

1. compute displacement **a**
2. compute strains ε, get history α
3. compute stresses/material tangent
4. compute internal force vector
5. compute tangent stiffness

$$\mathbf{a} = \mathbf{N}\mathbf{a}_e$$

$$\boldsymbol{\epsilon} = \mathbf{B}\mathbf{a}$$

$$\boldsymbol{\sigma} = f(\boldsymbol{\epsilon}, \alpha) \qquad \mathbf{C}$$

$$\mathbf{f}_{\text{int}} = \int_\Omega \mathbf{B}^{\text{T}}\boldsymbol{\sigma}\text{d}\Omega$$

$$\mathbf{K} \leftarrow \int_\Omega \mathbf{B}^{\text{T}}\mathbf{C}\mathbf{B}\text{d}\Omega$$

At integration level

For one Gauss point, do

1. compute displacement **a**
2. compute strains ε, get history α
3. compute stresses/material tangent
4. compute internal force vector
5. compute tangent stiffness

$$\mathbf{a} = \mathbf{N}\mathbf{a}_e$$

$$\boldsymbol{\epsilon} = \mathbf{B}\mathbf{a}$$

$$\boldsymbol{\sigma} = f(\boldsymbol{\epsilon}, \alpha) \qquad \mathbf{C}$$

$$\mathbf{f}_{\text{int}} = \int_{\Omega} \mathbf{B}^{\text{T}} \boldsymbol{\sigma} \text{d}\Omega$$

$$\mathbf{K} \leftarrow \int_{\Omega} \mathbf{B}^{\text{T}} \mathbf{C} \mathbf{B} \text{d}\Omega$$

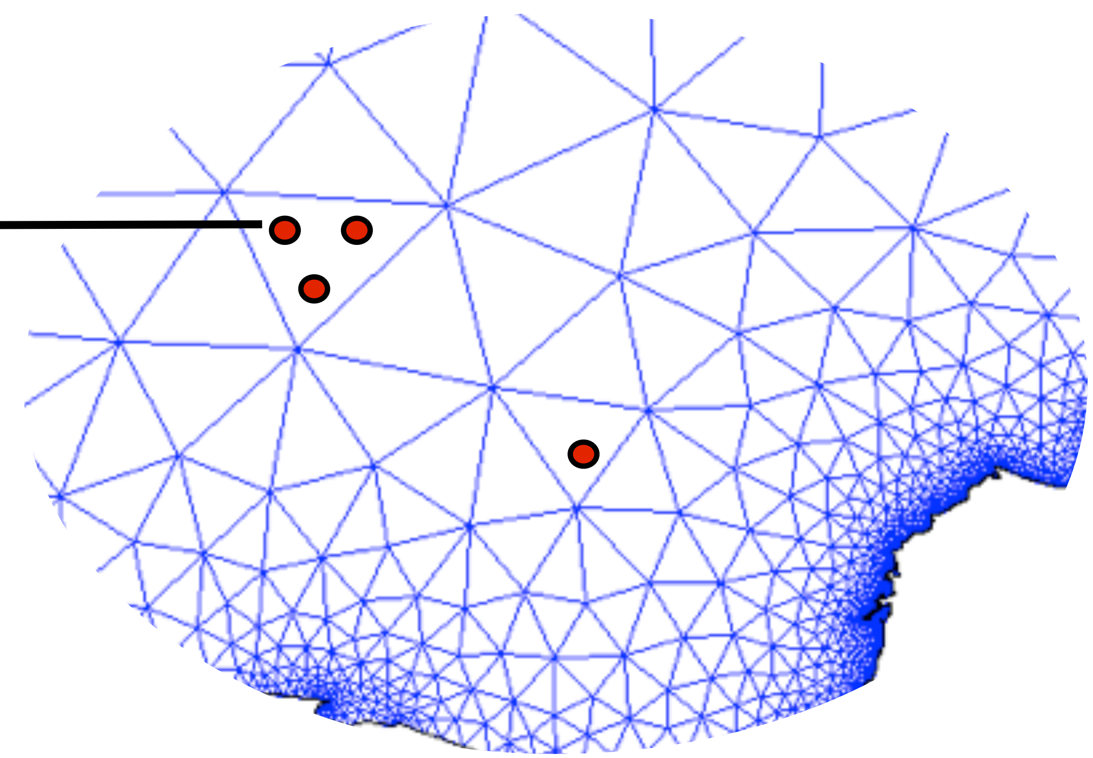
stress update

Storage of history vars

History variables

FE mesh

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_{n-1} \\ \alpha_n \end{bmatrix}$$



$\boldsymbol{\alpha}_0$: history vars at previous converged load step

n: number of integration points of the mesh

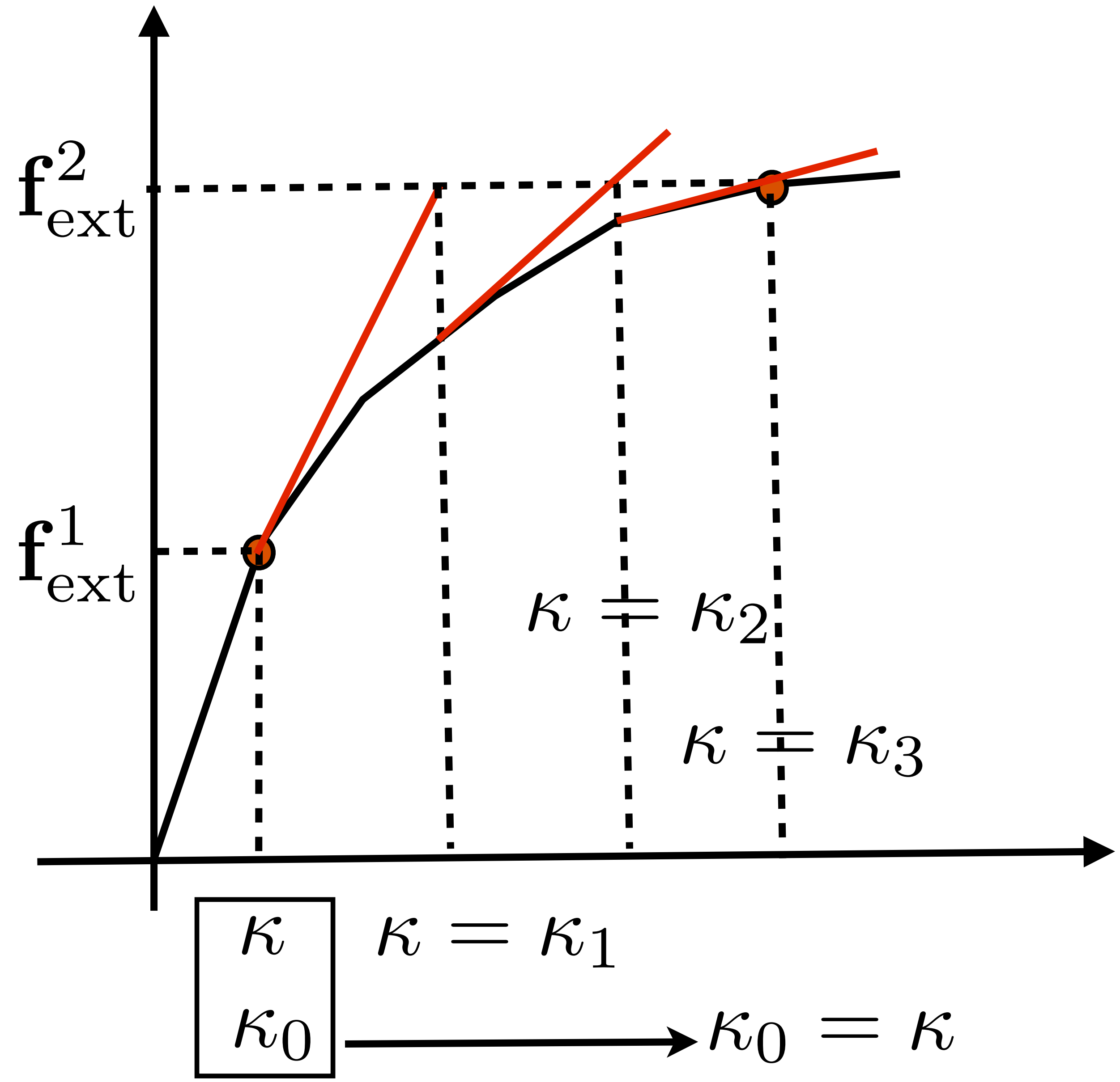
Isotropic damage model

$$\boldsymbol{\sigma} = (1-\omega)\mathbf{D}\boldsymbol{\epsilon}$$

$$\kappa = \max \epsilon_{\text{eq}}$$

$$\omega = f(\kappa)$$

$$\epsilon_{\text{eq}} = g(\boldsymbol{\epsilon})$$



commit history variables

$$\dot{\boldsymbol{\sigma}} = (1-\omega)\mathbf{D}\dot{\boldsymbol{\epsilon}} - \mathbf{D}\boldsymbol{\epsilon}\dot{\omega}$$

$$\dot{\boldsymbol{\sigma}} = (1-\omega)\mathbf{D}\dot{\boldsymbol{\epsilon}} - \mathbf{D}\boldsymbol{\epsilon}\frac{\partial f}{\partial \epsilon_{\text{eq}}}\frac{\partial \epsilon_{\text{eq}}}{\partial \boldsymbol{\epsilon}}\dot{\boldsymbol{\epsilon}}$$

$$\dot{\boldsymbol{\sigma}} = \left[(1-\omega)\mathbf{D} - \mathbf{D}\boldsymbol{\epsilon}\frac{\partial f}{\partial \epsilon_{\text{eq}}}\frac{\partial \epsilon_{\text{eq}}}{\partial \boldsymbol{\epsilon}}\right]\dot{\boldsymbol{\epsilon}} \qquad \mathbf{C}$$

consistent material tangent

$$\omega = 1 - \frac{\kappa}{\kappa_I}[1 - \alpha + \alpha \exp^{-\beta(\kappa - \kappa_I)}], \quad \kappa \geq \kappa_I$$

$$\epsilon_{\text{eq}} = \sqrt{\langle \epsilon_1 \rangle^2 + \langle \epsilon_2 \rangle^2 + \langle \epsilon_3 \rangle^2} \qquad \langle x \rangle = 0.5(x + |x|)$$

$$\frac{\partial \epsilon_{\text{eq}}}{\partial \epsilon_{ij}} = \frac{1}{\epsilon_{\text{eq}}} \left(\langle \epsilon_1 \rangle \frac{\partial \epsilon_1}{\partial \epsilon_{ij}} + \langle \epsilon_2 \rangle \frac{\partial \epsilon_2}{\partial \epsilon_{ij}} + \langle \epsilon_3 \rangle \frac{\partial \epsilon_3}{\partial \epsilon_{ij}} \right)$$

$$\langle \epsilon_k \rangle \frac{\partial \langle \epsilon_k \rangle}{\partial \epsilon_{ij}} = \begin{cases} \epsilon_k \frac{\partial \epsilon_k}{\partial \epsilon_{ij}} & \text{if } \epsilon_k > 0 \\ 0 & \text{if } \epsilon_k \leq 0 \end{cases}$$

$$\epsilon^3 - I_1 \epsilon^2 + I_2 \epsilon - I_3 = 0$$

$$I_1 = \epsilon_{xx} + \epsilon_{yy} + \epsilon_{zz}$$
$$I_2 = \epsilon_{xx}\epsilon_{yy} + \epsilon_{yy}\epsilon_{zz} + \epsilon_{zz}\epsilon_{xx} - \epsilon_{xy}^2 - \epsilon_{yz}^2 - \epsilon_{zx}^2$$
$$I_3 = \epsilon_{xx}\epsilon_{yy}\epsilon_{zz} + 2\epsilon_{xy}\epsilon_{yz}\epsilon_{zx} - \epsilon_{xx}\epsilon_{yz}^2 - \epsilon_{yy}\epsilon_{zx}^2 - \epsilon_{zz}\epsilon_{xy}^2$$

$$\frac{\partial \epsilon_k}{\partial \epsilon_{ij}} = \frac{\partial \epsilon_k}{\partial I_1} \frac{\partial I_1}{\partial \epsilon_{ij}} + \frac{\partial \epsilon_k}{\partial I_2} \frac{\partial I_1}{\partial \epsilon_{ij}} + \frac{\partial \epsilon_k}{\partial I_3} \frac{\partial I_1}{\partial \epsilon_{ij}}$$

Isotropic damage: stress update

PyFEM

```
def getStress( self, kinematics ):

  kappa = self.getHistoryParameter('kappa')

  eps , detadstrain = self.getEquivStrain( kinematics.strain )

  if eps > kappa:
    progDam = True
    kappa   = eps
  else:
    progDam = False

  self.setHistoryParameter( 'kappa', kappa )

  omega , domegadkappa = self.getDamage( kappa )

  effStress = dot( self.De , kinematics.strain )

  stress    = ( 1. - omega ) * effStress
  tang      = ( 1. - omega ) * self.De

  if progDam:
    tang += -domegadkappa * outer( effStress , detadstrain )

  return stress , tang
```

One-dimensional elasto-plasticity

```
def getStress( self, deformation ):

  # retrieve history variables

  epsp0  = self.getHistoryParameter('plasticStr')
  alpha0 = self.getHistoryParameter('hardening')

  sigmaTrial = self.E * ( deformation.strain - epsp0 )
  yieldFunc  = abs ( sigmaTrial ) - ( self.sigmaY + self.K * alpha0 )

  if yieldFunc <= 0.:
    sigma = sigmaTrial
    tang  = self.E

    epsp  = epsp0
    alpha = alpha0
  else:
    deltaGamma = yieldFunc / ( self.E + self. K )
    sigma      = ( 1. - deltaGamma * self.E / abs (sigmaTrial)  ) * sigmaTrial
    epsp       = epsp0  + deltaGamma * sign ( sigmaTrial )
    alpha      = alpha0 + deltaGamma
    tang       = self.E * self.K / ( self.E + self.K )

  self.setHistoryParameter( 'plasticStr', epsp   )
  self.setHistoryParameter( 'hardening',  alpha  )

  return sigma, tang
```

two history variables/GP

isotropic hardening

Convergence rate

It has been mathematically proved, full NR converges quadratically closed to the solution.

Convergence rate

It has been mathematically proved, full NR converges quadratically closed to the solution.

In reality …

Convergence rate

It has been mathematically proved, full NR converges quadratically closed to the solution.

In reality …

```
AttributeError: issim
module `extraModules.solver.nonLin' : residual scale factor = 3.9758e+01
module `extraModules.solver.nonLin' : iter = 1, scaled residual = 8.9753e-02
module `extraModules.solver.nonLin' : iter = 2, scaled residual = 1.9729e-02
module `extraModules.solver.nonLin' : iter = 3, scaled residual = 6.0983e-04
module `extraModules.solver.nonLin' : iter = 4, scaled residual = 6.3835e-08
The Newton-Raphson solver converged in 4 iterations
```

Convergence rate

It has been mathematically proved, full NR converges quadratically closed to the solution.

In reality …

```
AttributeError: issim
module `extraModules.solver.nonLin' : residual scale factor = 3.9758e+01
module `extraModules.solver.nonLin' : iter = 1, scaled residual = 8.9753e-02
module `extraModules.solver.nonLin' : iter = 2, scaled residual = 1.9729e-02
module `extraModules.solver.nonLin' : iter = 3, scaled residual = 6.0983e-04
module `extraModules.solver.nonLin' : iter = 4, scaled residual = 6.3835e-08
The Newton-Raphson solver converged in 4 iterations
```

Convergence rate

It has been mathematically proved, full NR converges quadratically closed to the solution.

In reality …

```
AttributeError: issim
module `extraModules.solver.nonLin' : residual scale factor = 3.9758e+01
module `extraModules.solver.nonLin' : iter = 1, scaled residual = 8.9753e-02
module `extraModules.solver.nonLin' : iter = 2, scaled residual = 1.9729e-02
module `extraModules.solver.nonLin' : iter = 3, scaled residual = 6.0983e-04
module `extraModules.solver.nonLin' : iter = 4, scaled residual = 6.3835e-08
The Newton-Raphson solver converged in 4 iterations
```

quadratic

Convergence criteria

Residual
$$\frac{||\mathbf{r}||}{||\mathbf{f}_{\text{ext}}||} \leq \epsilon$$

Energy
$$\frac{\mathbf{f}_{\text{int}}^{\text{T}} \delta \mathbf{a}}{\mathbf{f}_{\text{int},1}^{\text{T}} \delta \mathbf{a}_1} \leq \epsilon$$

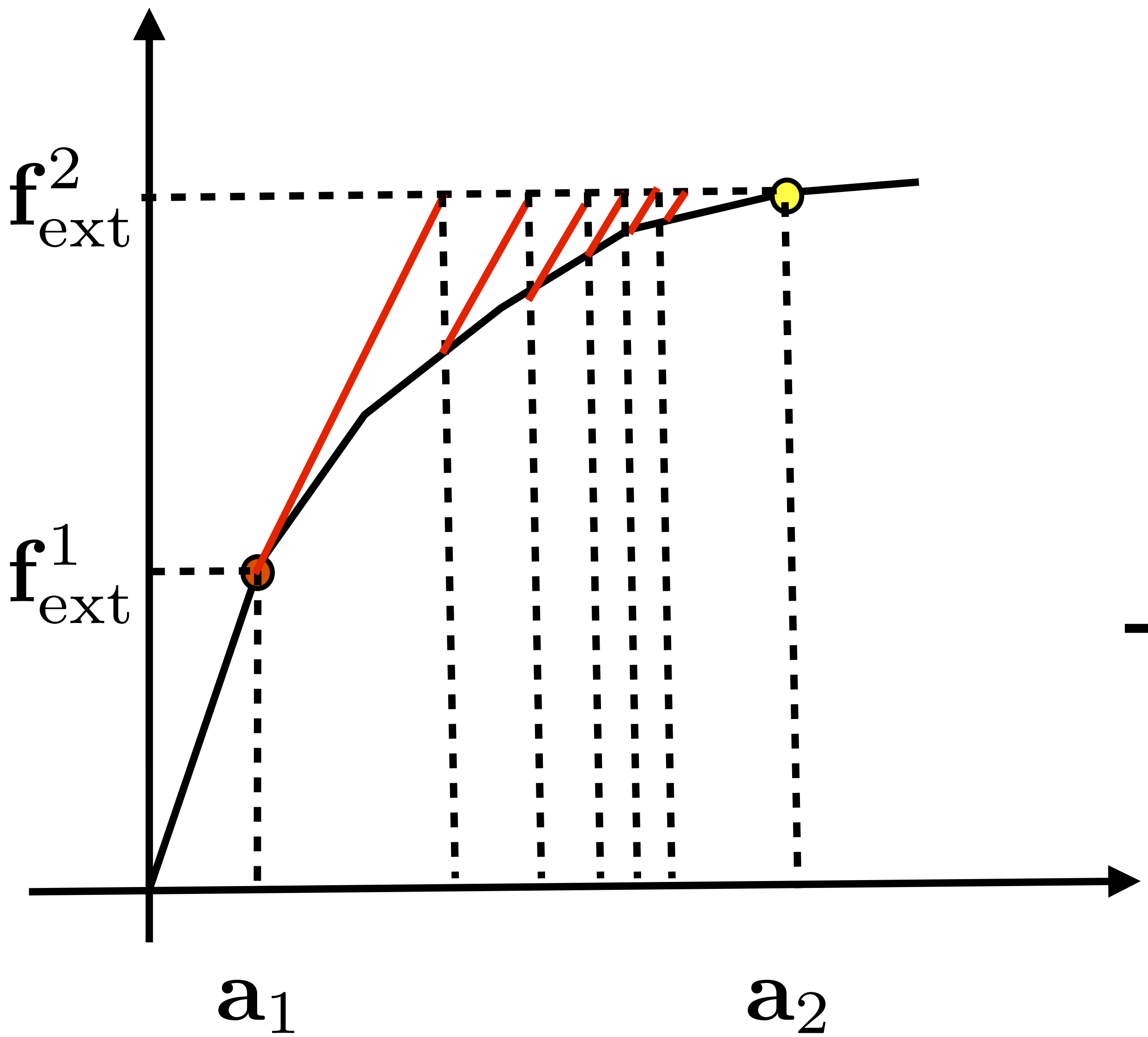
Displacement
$$\frac{||\delta \mathbf{a}||}{\max \ ||\delta \mathbf{a}_i||} \leq \epsilon$$

$$||\mathbf{a}|| = \sqrt{\sum_{i=1}^{n} a_i^2}$$

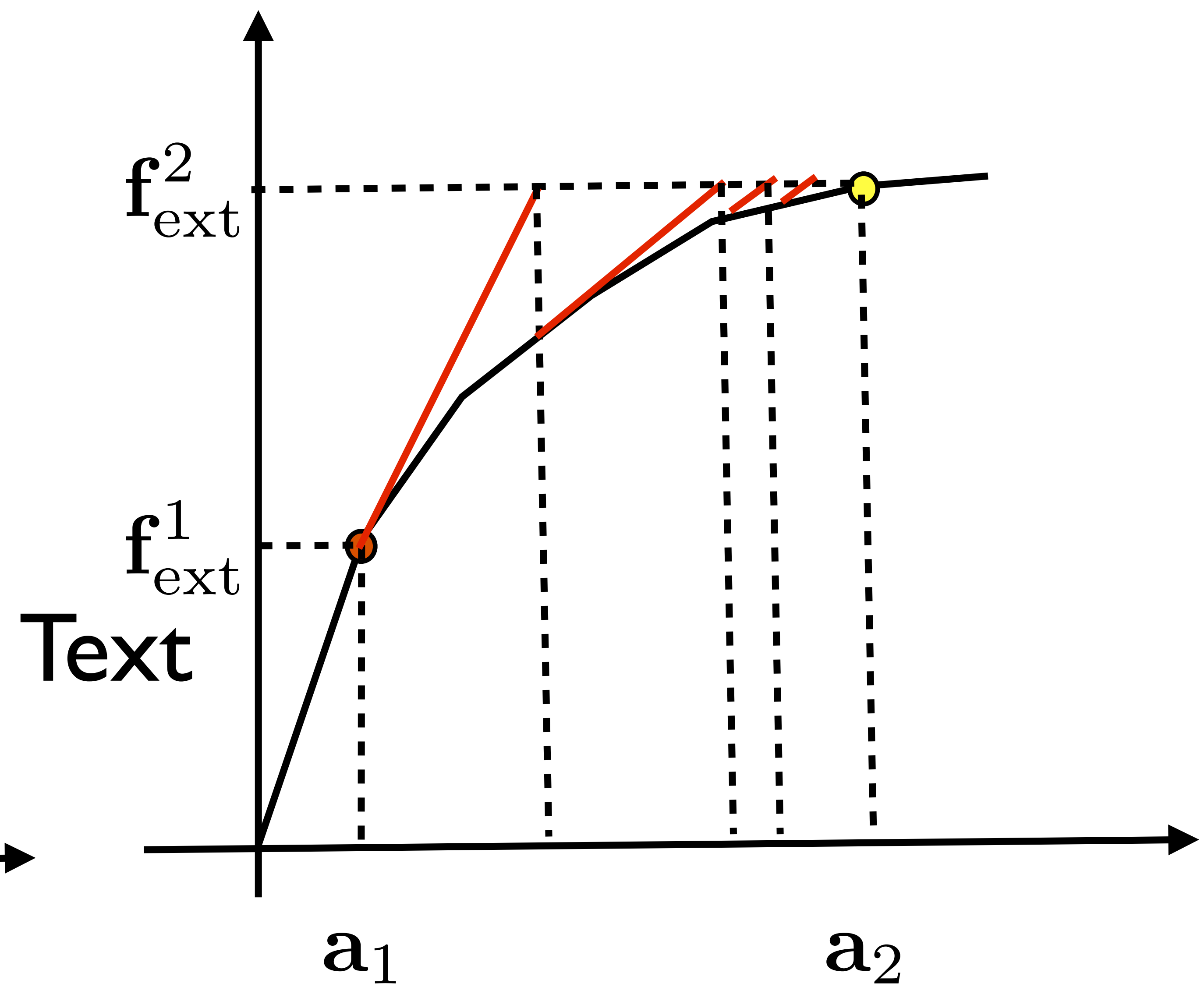
relative norms

tolerance is problem dependent

Modified NR methods



Initial stiffness method



Modified Newton-Raphson

more iterations per step

without consistent material tangent

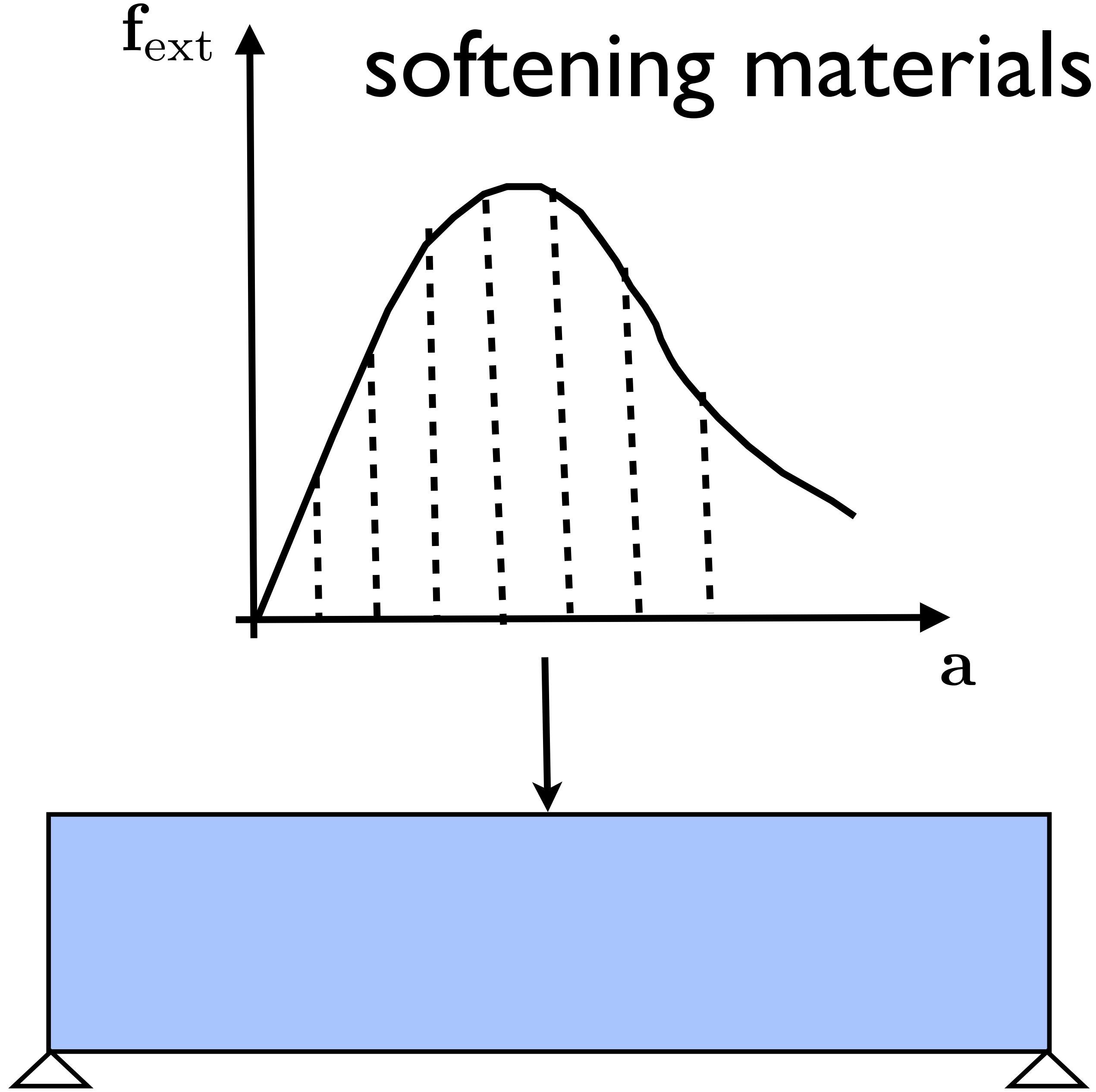
Direct displacement control

$$\mathbf{f}_{\text{int}}(\mathbf{a}) = \mathbf{0}$$

$$a_5 = \lambda \bar{a}, \quad \lambda = 0, 1, 2, \ldots$$

Argyris (1965)

Batoz, Dhatt (1979)



For each load step:

First iteration:

$$\mathbf{K}\delta\mathbf{a}_{(1)} = -\mathbf{f}_{\text{int}}$$

$$\delta a_5 = \bar{a}$$

From second iteration:

$$\mathbf{K}\delta\mathbf{a}_{(i)} = -\mathbf{f}_{\text{int}}$$

$$\delta a_5 = 0$$

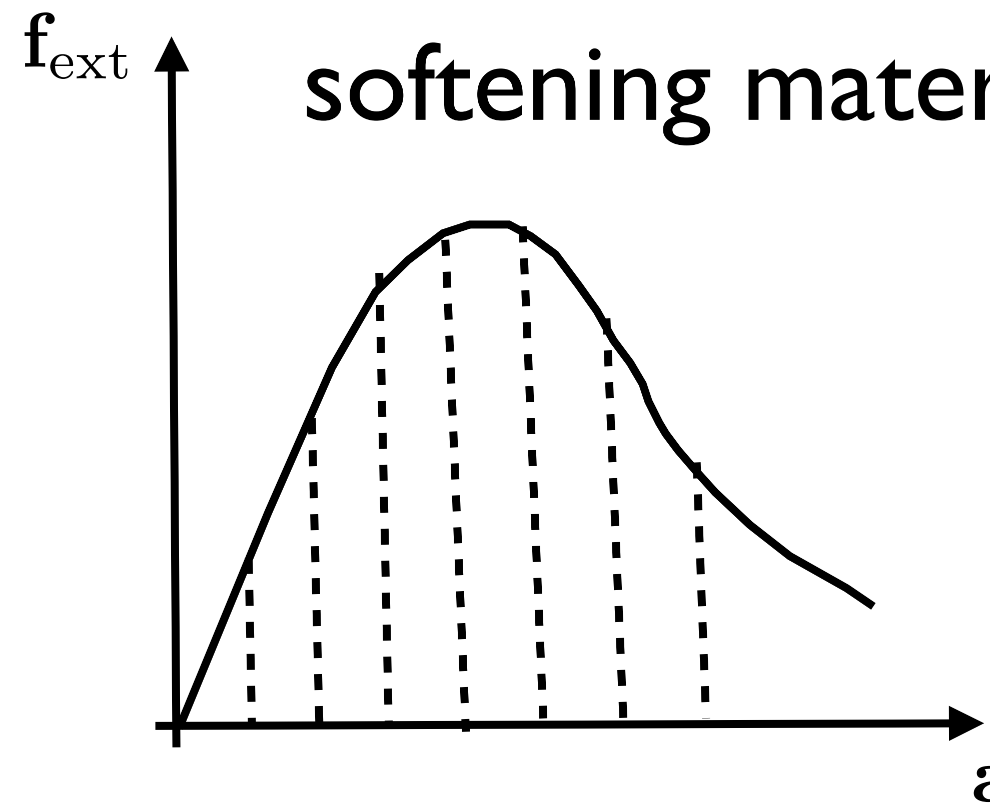
Direct displacement control

$$\mathbf{f}_{\text{int}}(\mathbf{a}) = \mathbf{0}$$

$$a_5 = \lambda \bar{a}, \quad \lambda = 0, 1, 2, ...$$

Argyris (1965)

Batoz, Dhatt (1979)



For each load step:

First iteration:

$$\mathbf{K}\delta\mathbf{a}_{(1)} = -\mathbf{f}_{\text{int}}$$

$$\delta a_5 = \bar{a}$$

From second iteration:

$$\mathbf{K}\delta\mathbf{a}_{(i)} = -\mathbf{f}_{\text{int}}$$

$$\delta a_5 = 0$$

Direct displacement control

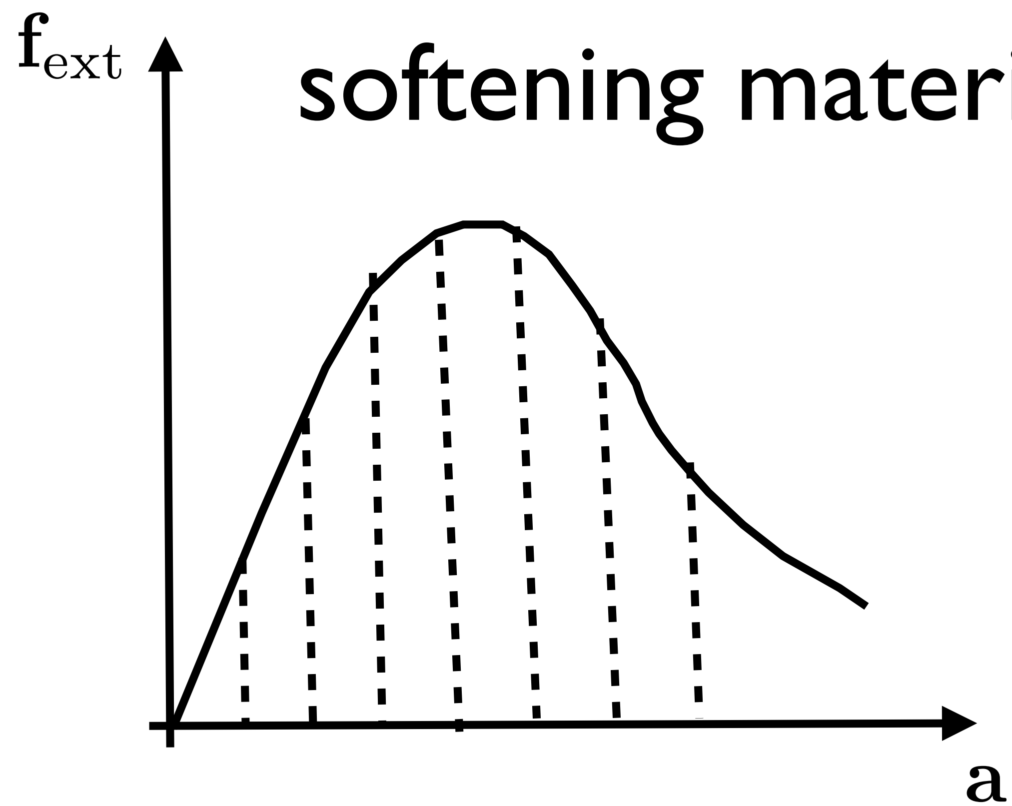
$$\mathbf{f}_{\text{int}}(\mathbf{a}) = \mathbf{0}$$

$$a_5 = \lambda \bar{a}, \quad \lambda = 0, 1, 2, \ldots$$

Argyris (1965)

Batoz, Dhatt (1979)

softening materials



For each load step:

First iteration:

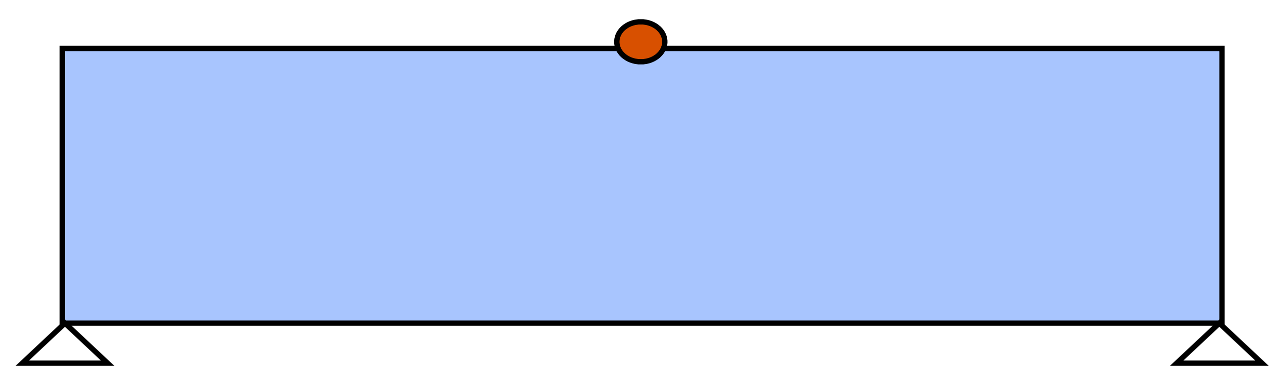
$$\mathbf{K}\delta\mathbf{a}_{(1)} = -\mathbf{f}_{\text{int}}$$

$$\delta a_5 = \bar{a}$$

From second iteration:

$$\mathbf{K}\delta\mathbf{a}_{(i)} = -\mathbf{f}_{\text{int}}$$

$$\delta a_5 = 0$$



Direct displacement control

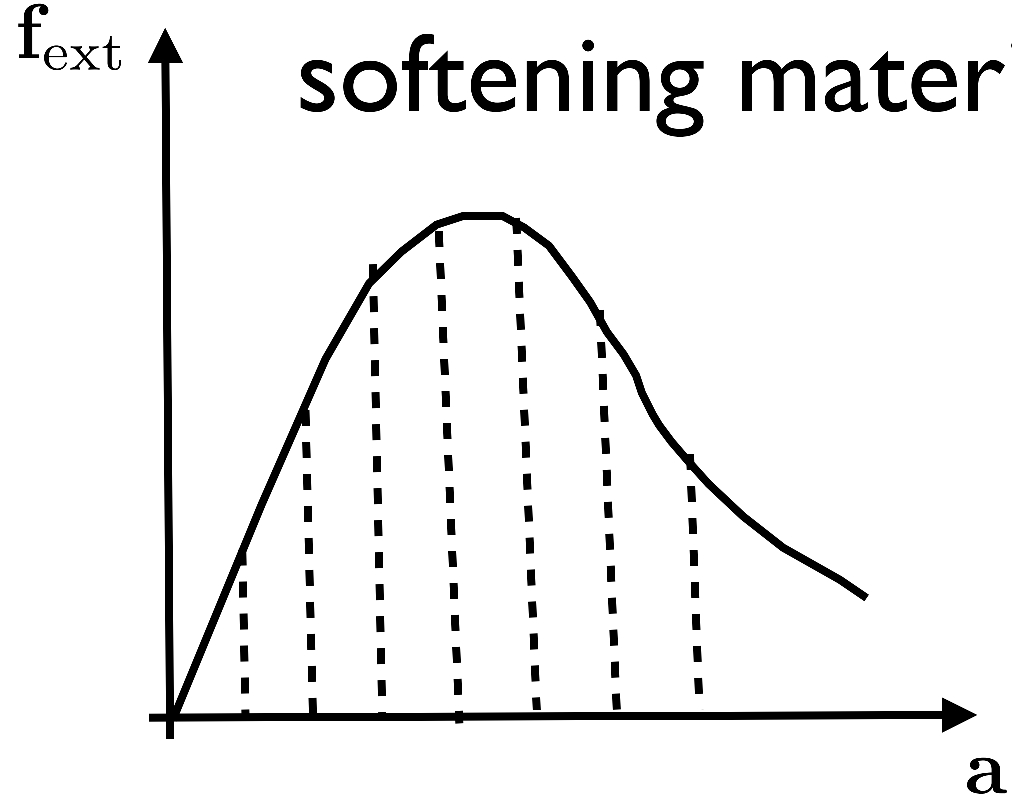
$$\mathbf{f}_{\text{int}}(\mathbf{a}) = \mathbf{0}$$

$$a_5 = \lambda \bar{a}, \quad \lambda = 0, 1, 2, ...$$

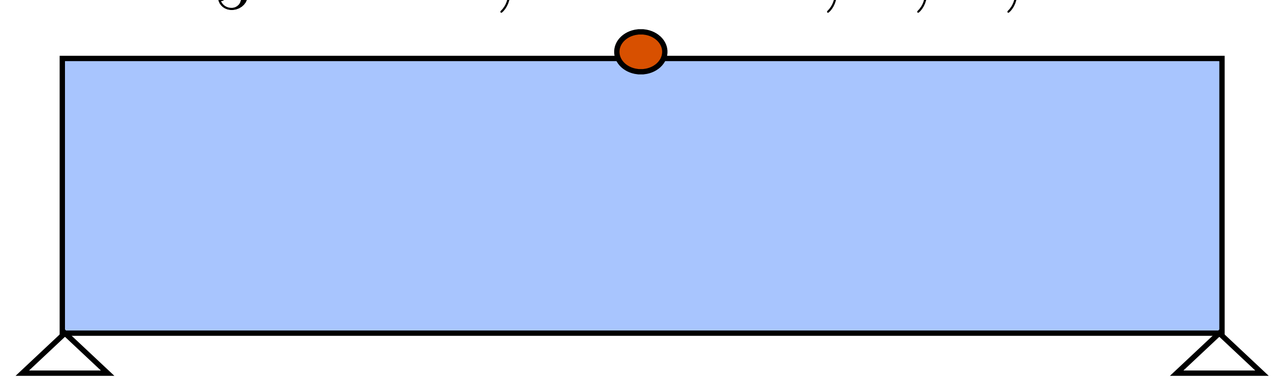
Argyris (1965)

Batoz, Dhatt (1979)

softening materials



$$a_5 = \lambda \bar{a}, \quad \lambda = 0, 1, 2, ...$$



For each load step:

First iteration:

$$\mathbf{K}\delta\mathbf{a}_{(1)} = -\mathbf{f}_{\text{int}}$$

$$\delta a_5 = \bar{a}$$

From second iteration:

$$\mathbf{K}\delta\mathbf{a}_{(i)} = -\mathbf{f}_{\text{int}}$$

$$\delta a_5 = 0$$

Divergence issues

```
module `extraModules.solver.nonLin' : residual scale factor = 4.2697e+01
module `extraModules.solver.nonLin' : iter = 1, scaled residual = 1.1246e-01
module `extraModules.solver.nonLin' : iter = 2, scaled residual = 5.5655e-02
module `extraModules.solver.nonLin' : iter = 3, scaled residual = 3.1334e-02
module `extraModules.solver.nonLin' : iter = 4, scaled residual = 3.9274e-02
module `extraModules.solver.nonLin' : iter = 5, scaled residual = 5.4948e-02
module `extraModules.solver.nonLin' : iter = 6, scaled residual = 3.3929e-02
module `extraModules.solver.nonLin' : iter = 7, scaled residual = 4.6461e-02
module `extraModules.solver.nonLin' : iter = 8, scaled residual = 2.7252e-02
module `extraModules.solver.nonLin' : iter = 9, scaled residual = 4.7695e-02
module `extraModules.solver.nonLin' : iter = 10, scaled residual = 4.1914e-02
module `extraModules.solver.nonLin' : iter = 11, scaled residual = 1.1236e-01
module `extraModules.solver.nonLin' : iter = 12, scaled residual = 6.0917e-02
module `extraModules.solver.nonLin' : iter = 13, scaled residual = 5.6249e-02
module `extraModules.solver.nonLin' : iter = 14, scaled residual = 3.4289e-02
module `extraModules.solver.nonLin' : iter = 15, scaled residual = 4.4957e-02
module `extraModules.solver.nonLin' : iter = 16, scaled residual = 2.7413e-02
```

the system is failed structurally or numerically???

Elasto-plastic models

Linear visoelasticity