CIEM5110-2: FEM, lecture 4.2

Plastic hinges and arc-length

Frans van der Meer and Iuri Rocha

TUDelft

Agenda

Another lecture on nonlinear analysis of frames

- Material nonlinearity in the form of plastic hinges
- Arclength solution method
- Comparison with analytical rigid-plastic solution from *Stability* unit

CIEM5110-2 workshops and lectures

	(Theory)	BarModel (MUDE)	SolidModel (1.2)	TimoshenkoModel (2.1)	FrameModel (4.1)
SolverModule	(1.2)	2.2	2.2	3.2	3.2
NonlinModule	(3.1)		6.1		4.1 + **4.2** + 5.1
ArclenModule	(**4.2**)				**4.2**
LinBuckModule	(4.1)				4.1 + 5.1
ModeShapeModule	(6.2)		7.1		7.1 + 8.1
ExplicitTimeModule	(6.2)				7.2 + 8.1
NewmarkModule	(6.2)				7.2 + 8.1

TU Delft

Nonlinear finite element analysis

Nonlinear system of equations

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

with

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

- Material nonlinearity ($\boldsymbol{\sigma}(\mathbf{a})$) and/or geometric nonlinearity ($\mathbf{B}(\mathbf{a})$)
- Incremental-iterative analysis: “time stepping”
- Displacement control or load control
- Newton-Raphson scheme to solve nonlinear system of equations
- Arclength method for snapback and for post-peak response with proportional loads

Nonlinear finite element analysis

Nonlinear system of equations

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

with

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

- Material nonlinearity ($\boldsymbol{\sigma}(\mathbf{a})$) and/or geometric nonlinearity ($\mathbf{B}(\mathbf{a})$)
- Incremental-iterative analysis: “time stepping”
- Displacement control or load control
- Newton-Raphson scheme to solve nonlinear system of equations
- Arclength method for snapback and for post-peak response with proportional loads
  Two arc-length parameters: $\Delta\ell$ and β

$$\Delta\ell = \sqrt{\Delta\mathbf{a} \cdot \Delta\mathbf{a} + \beta^2 \Delta\mathbf{f} \cdot \Delta\mathbf{f}}$$

Plastic hinges in `pyJive`

The `framemodel` can deal with plastic hinges:

- Set `plastic = True`
- Define `Mp` as additional material parameter
- Use `subtype = linear` or `subtype = nonlin` for geometrically linear/nonlinear analysis

Plastic hinges in `pyJive`

The `framemodel` can deal with plastic hinges:

- Set `plastic = True`
- Define `Mp` as additional material parameter
- Use `subtype = linear` or `subtype = nonlin` for geometrically linear/nonlinear analysis

When bending moment exceeds plastic moment, a plastic hinge is added

- Additional rotational degree of freedom is introduced
- External forces are added (pair of opposite bending moments)

Plastic hinges in `pyJive`

The `framemodel` can deal with plastic hinges:

- Set `plastic = True`
- Define `Mp` as additional material parameter
- Use `subtype = linear` or `subtype = nonlin` for geometrically linear/nonlinear analysis

When bending moment exceeds plastic moment, a plastic hinge is added

- Additional rotational degree of freedom is introduced
- External forces are added (pair of opposite bending moments)

The code could be made more complete by adding

- Possibility of unloading of plastic hinge
- Influence of normal force on M_p
- Development of plasticity in nonlinear $M(\kappa)$-relation

Plastic hinges in solution algorithm: single time step

A check is added to the end of the algorithm for a single time step.

See lines 11-16

Require: Solution from previous time step $\mathbf{a}^n$

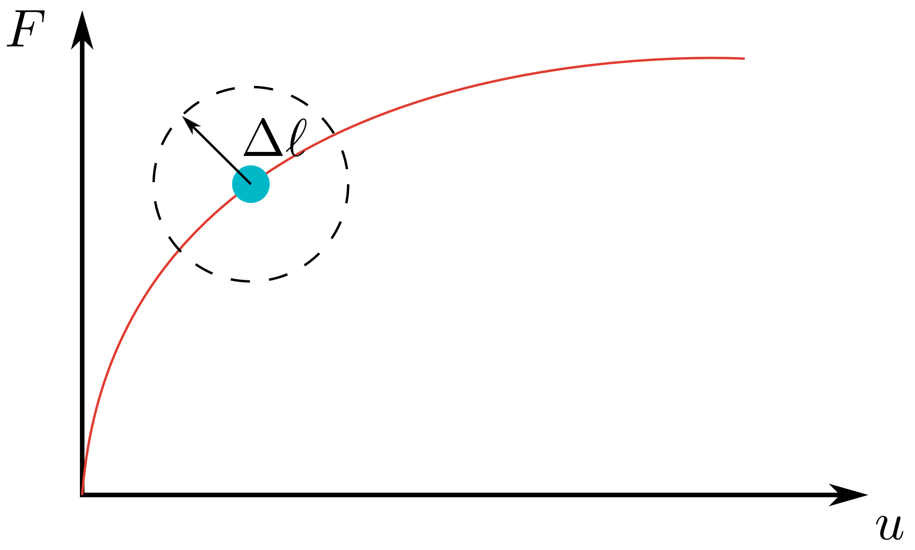
Require: Nonlinear relation $\mathbf{f}_{\text{int}}(\mathbf{a})$ with $\mathbf{K}(\mathbf{a}) = \frac{\partial \mathbf{f}_{\text{int}}}{\partial \mathbf{a}}$

```
1:  Get new external force vector: f_ext^{n+1}
2:  Initialize new solution at old one: a^{n+1} = a^n
3:  Compute internal force and stiffness: f_int^{n+1}(a^{n+1}), K^{n+1}(a^{n+1})
4:  Evaluate residual: r = f_ext^{n+1} − f_int^{n+1}
5:  repeat
6:      Solve linear system of equations: K^{n+1} Δa = r
7:      Update solution: a^{n+1} = a^{n+1} + Δa
8:      Compute internal force and stiffness: f_int^{n+1}(a^{n+1}), K^{n+1}(a^{n+1})
9:      Evaluate residual: r = f_ext^{n+1} − f_int^{n+1}
10: until |r| < tolerance
11: if max(abs(M)) > M_p then
12:     Insert plastic hinge at the position of max(abs(M))
13:     Go back to 3.
14: else
15:     Go to next time step
16: end if
```

Arc-length control — linearization

Redefine the external load vector:

$$\mathbf{r}\left(\mathbf{a}\right) = \mathbf{f}_{\text{ext}} - \mathbf{f}_{\text{int}}\left(\mathbf{a}\right) \quad \Rightarrow \quad \mathbf{r}\left(\mathbf{a}, \lambda\right) = \lambda\hat{\mathbf{f}} - \mathbf{f}_{\text{int}}\left(\mathbf{a}\right)$$



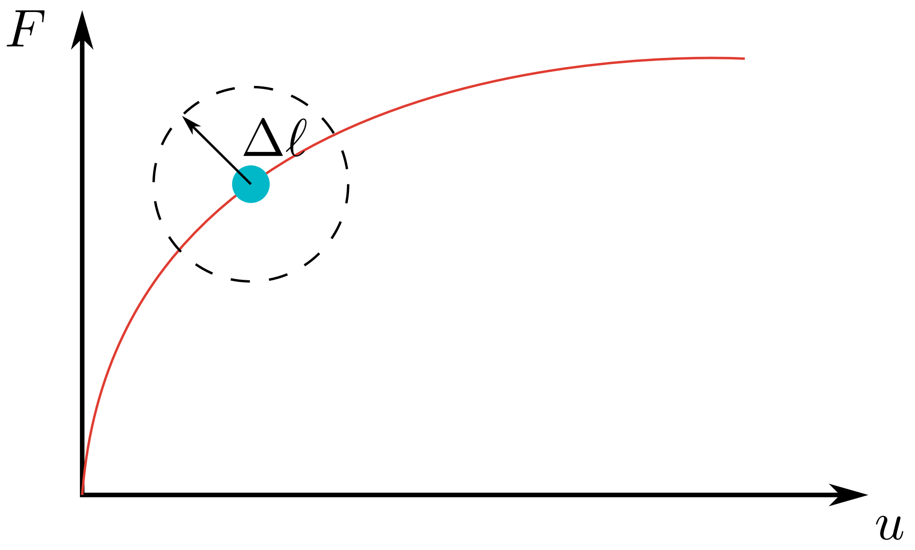
Arc-length control — linearization

Redefine the external load vector:

$$\mathbf{r}\left(\mathbf{a}\right) = \mathbf{f}_{\text{ext}} - \mathbf{f}_{\text{int}}\left(\mathbf{a}\right) \quad \Rightarrow \quad \mathbf{r}\left(\mathbf{a}, \lambda\right) = \lambda\hat{\mathbf{f}} - \mathbf{f}_{\text{int}}\left(\mathbf{a}\right)$$

Too many unknowns, so introduce a new constraint equation:

$$g\left(\Delta\mathbf{a}, \Delta\lambda, \Delta\ell\right) = 0$$



Arc-length control – linearization

Redefine the external load vector:

$$\mathbf{r}\left(\mathbf{a}\right) = \mathbf{f}_{\text{ext}} - \mathbf{f}_{\text{int}}\left(\mathbf{a}\right) \quad \Rightarrow \quad \mathbf{r}\left(\mathbf{a}, \lambda\right) = \lambda\hat{\mathbf{f}} - \mathbf{f}_{\text{int}}\left(\mathbf{a}\right)$$

Too many unknowns, so introduce a new constraint equation:

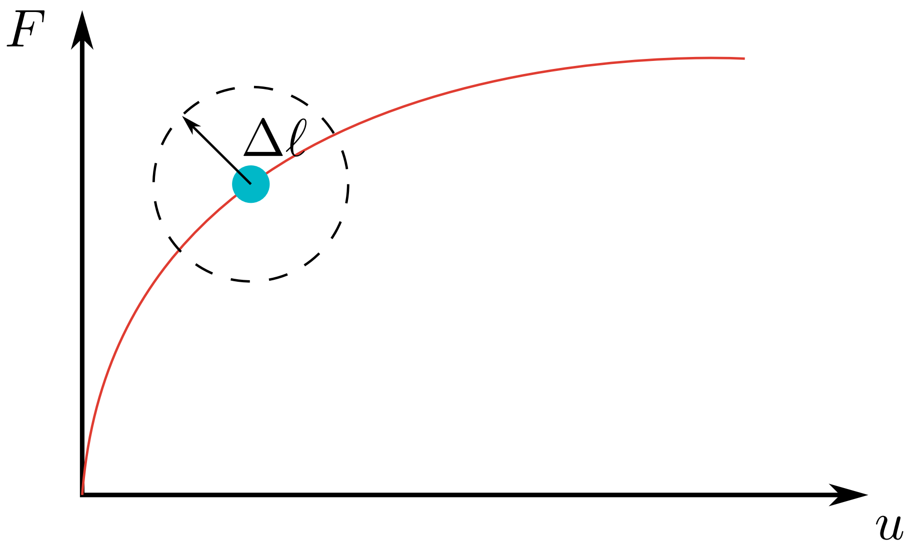
$$g\left(\Delta\mathbf{a}, \Delta\lambda, \Delta\ell\right) = 0$$

The linearization then changes:

$$\begin{bmatrix} \mathbf{K} & -\hat{\mathbf{f}} \\ \mathbf{h}^{\mathrm{T}} & s \end{bmatrix} \begin{bmatrix} \Delta\mathbf{a} \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} \mathbf{r} \\ -g \end{bmatrix}$$

where:

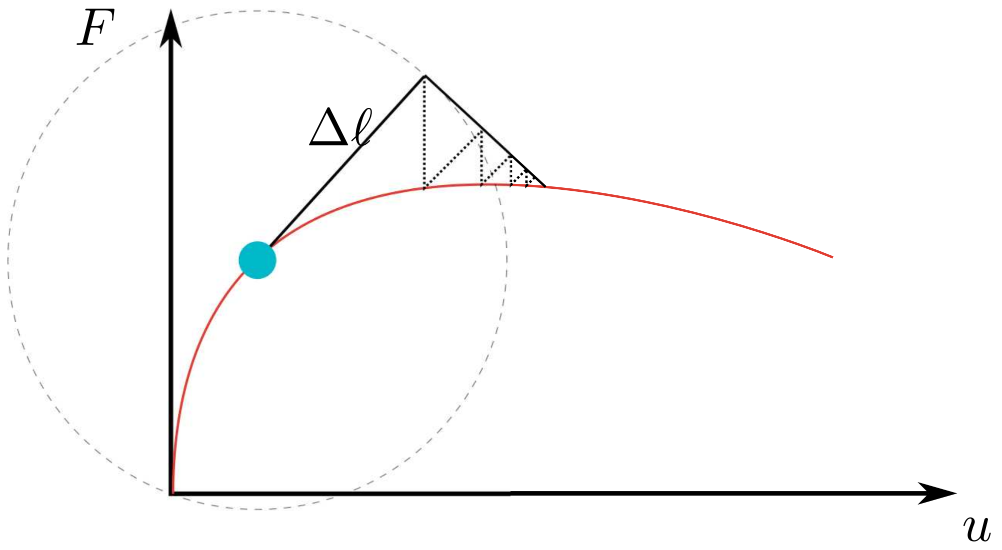
$$\mathbf{h} = \frac{\partial g}{\partial \mathbf{a}} \qquad s = \frac{\partial g}{\partial \lambda}$$



Arc-length control — constraints

Linearized constraint, modified linearization

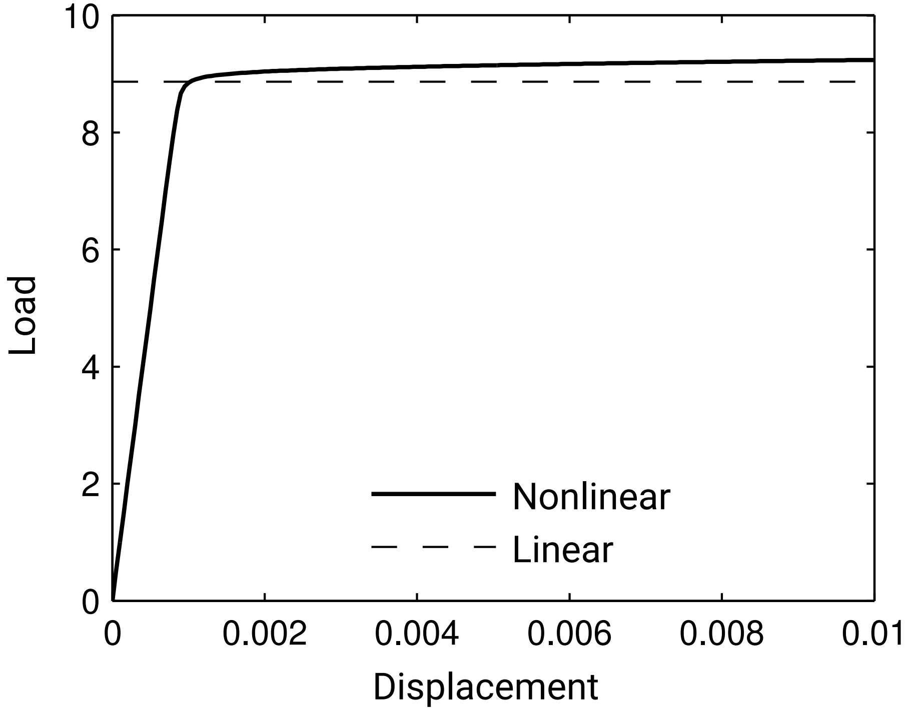
- Implemented in pyJive

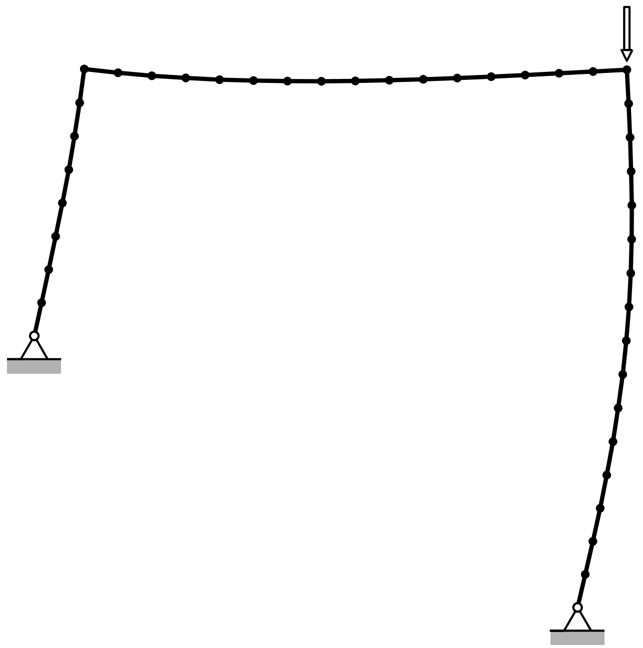


$$g = \Delta \mathbf{a}_0^{\mathrm{T}} \Delta \mathbf{a}_{j+1} + \beta^2 \Delta\lambda_0 \Delta\lambda_{j+1} \hat{\mathbf{f}}^{\mathrm{T}} \hat{\mathbf{f}} - \Delta\ell^2$$

Plastic analysis: frame results

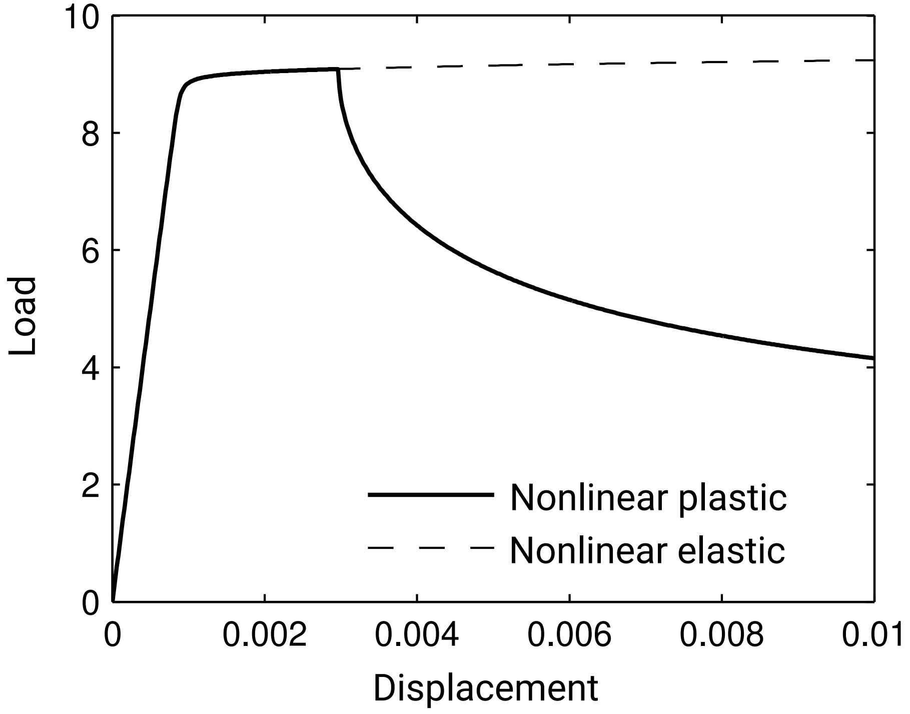
First load case (treated in elastic FEM lecture)

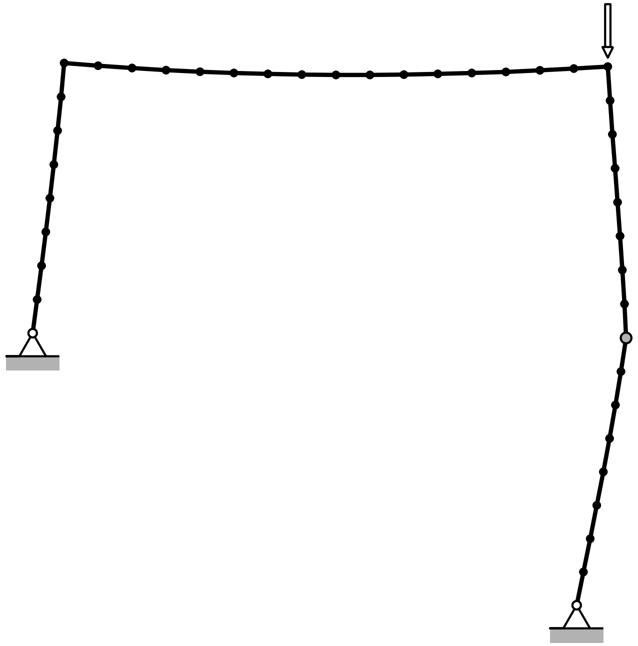




Plastic analysis: frame results

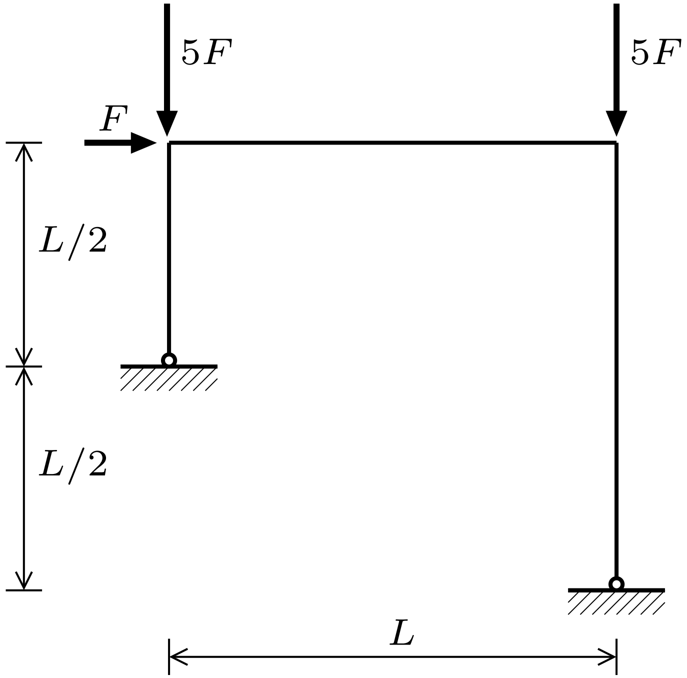
Results after adding plasticity





Plastic analysis: frame results

Second load case (treated in plasticity lecture): Including lateral load

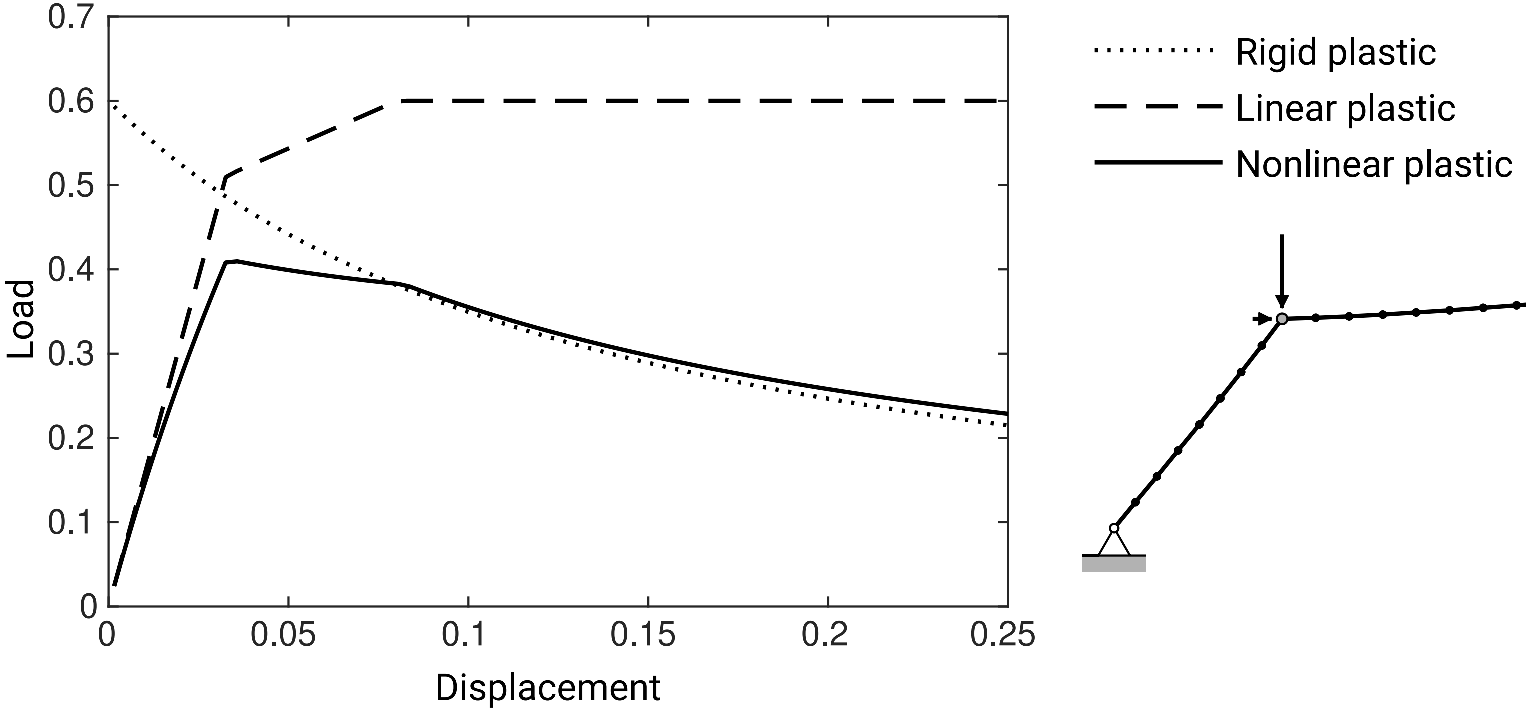


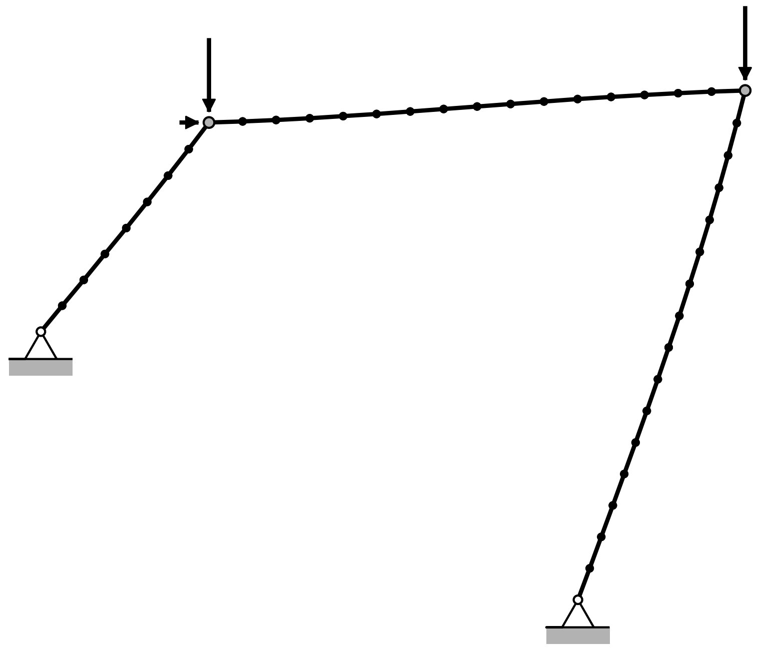
Plastic analysis: frame results

Second load case (treated in plasticity lecture): Including lateral load

Results from rigid-plastic hand calculation:

$$F = \frac{3M_\mathrm{p}}{L}\frac{1}{1+14\frac{1}{3}\theta}$$

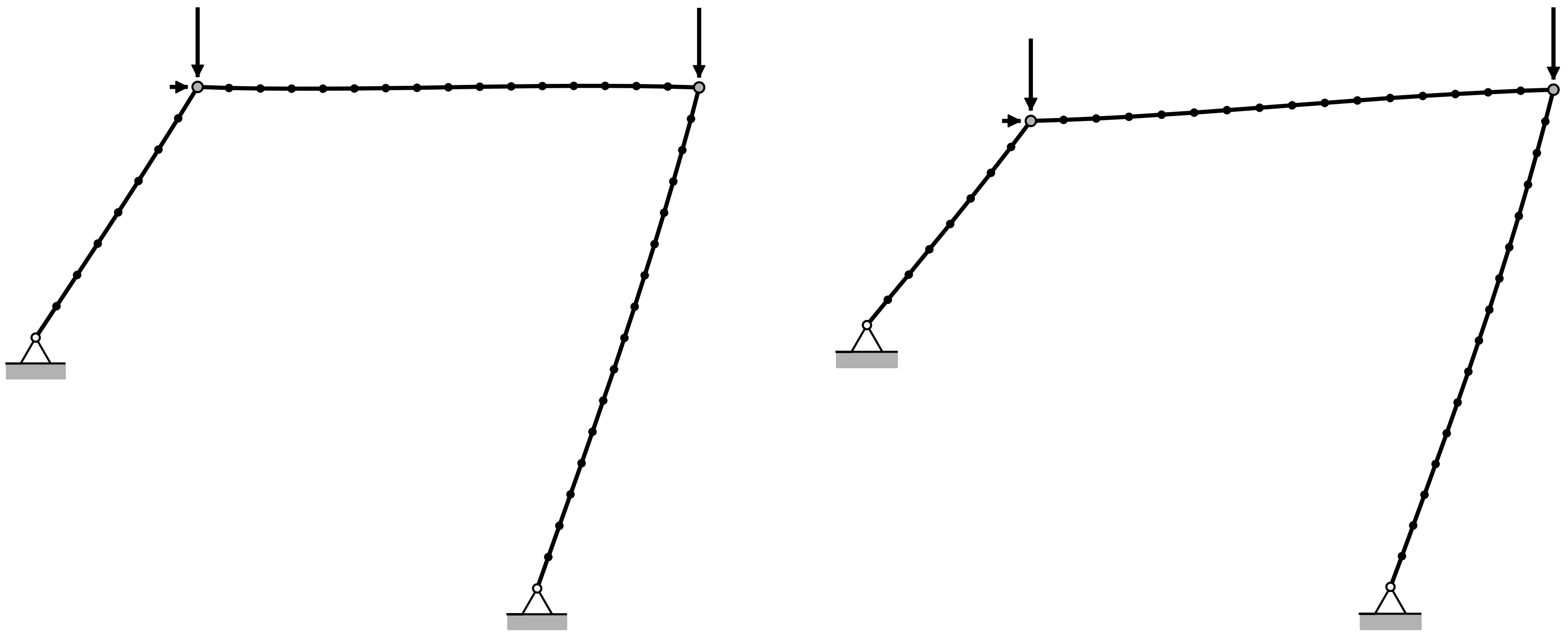




Plastic analysis: frame results

Second load case (treated in plasticity lecture): Including lateral load

Comparing geometrically linear/nonlinear collapse mechanism



Final considerations

Different approaches to the problem of structural collapse

- Linear buckling analysis
- Geometrically nonlinear elastic FEA
- Linear plastic hand calculation
- Nonlinear (2nd order) plastic hand calculation
- Full nonlinear FEA

Essential choices for running nonlinear FEA

- Watch out for bifurcations
- Choose appropriate increments (force/displacement/arclength)
- Consistent linearization