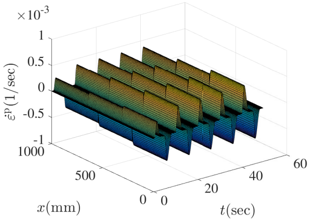
LATIN approach for fatigue damage computations

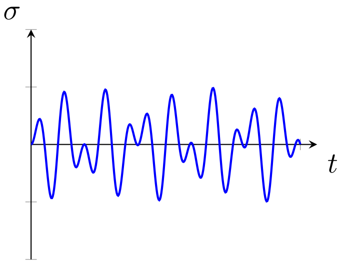
S. Alameddin†, M. Bhattacharyya†, A. Fau†,
U. Nackenhorst†, D. Néron‡, P. Ladevèze‡
† IBNM, Leibniz Universität Hannover
‡ LMT, ENS Cachan, CNRS, Université Paris Saclay
09. March 2017



Outline

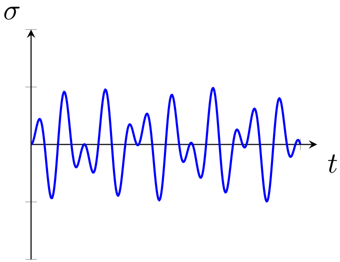
Motivation

- Cyclic loading

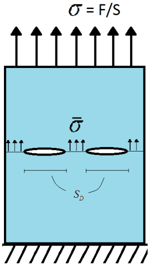


Motivation

- Cyclic loading

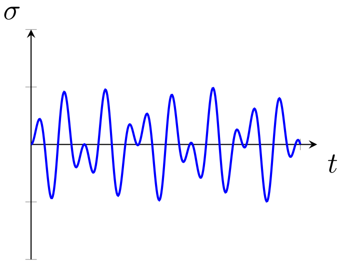


- Damage



Motivation

- Cyclic loading



- Virtual experiments
- Continuum damage framework
- Millions of cycles
- Computationally expensive

Model order reduction (MOR) techniques

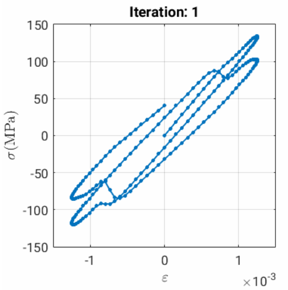
Model order reduction (MOR) techniques

- RB
- Proper orthogonal decomposition POD
- PGD

LATIN

Large time increment method [Ladevèze, 1999]

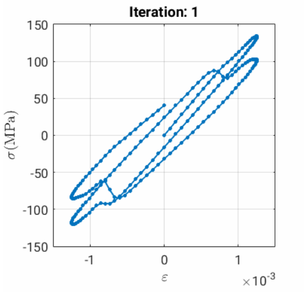
- At each iteration
  - an approximation on the **whole time domain** is obtained.
  - the balance equation is solved as a linearised problem



LATIN

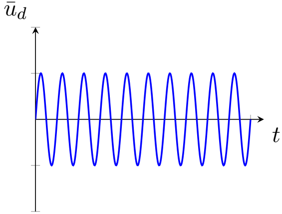
Large time increment method [Ladevèze, 1999]

- At each iteration
  - an approximation on the **whole time domain** is obtained.
  - the balance equation is solved as a linearised problem



- convenient to apply MOR

LATIN



Uniaxial visco-plastic bar with cyclic loading without considering dynamic effects

$$\psi(\varepsilon,\alpha) = \frac{1}{2}\,E\,(\varepsilon)^2 + \frac{1}{2}\,C\,(\alpha)^2$$

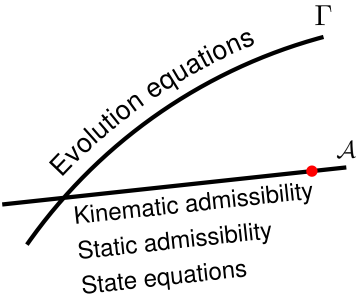
State equations: $\sigma = E\,\varepsilon^{\mathrm{e}}, \quad \beta = C\,\alpha$

$$\phi^{\mathrm{p}}(\sigma,\beta) = \frac{k}{n+1}\langle|\sigma-\beta| + \frac{a}{C}\,\beta^2 - \sigma_{\mathrm{y}}\rangle_+^{n+1}$$

Evolution equations: $\dot{\varepsilon}^{\mathrm{p}} = \dfrac{\partial\phi^{\mathrm{p}}}{\partial\sigma}, \quad \dot{\alpha} = -\dfrac{\partial\phi^{\mathrm{p}}}{\partial\beta}$

LATIN iteration

- Linear initialisation



LATIN iteration

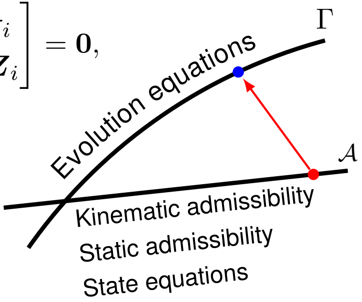
- Non-linear local step

$$\begin{bmatrix} \dot{\hat{\varepsilon}}^{\mathrm{p}}_{i+1/2} - \dot{\varepsilon}^{\mathrm{p}}_{i} \\ -(\dot{\hat{\boldsymbol{X}}}_{i+1/2} - \dot{\boldsymbol{X}}_{i}) \end{bmatrix} + H^{+} \begin{bmatrix} \hat{\sigma}_{i+1/2} - \sigma_{i} \\ \hat{\boldsymbol{Z}}_{i+1/2} - \boldsymbol{Z}_{i} \end{bmatrix} = \boldsymbol{0},$$

- Solve the evolution equations

$$\dot{\varepsilon}^{\mathrm{p}} = \frac{\partial \phi^{\mathrm{p}}}{\partial \sigma}, \quad \dot{\alpha} = -\frac{\partial \phi^{\mathrm{p}}}{\partial \beta}$$

- Suitable for parallelisation



LATIN iteration

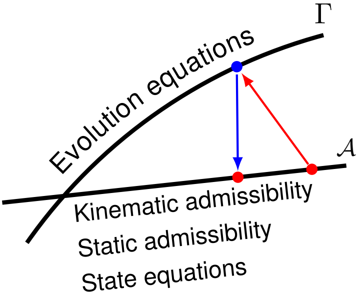
- Linear global step

$$\begin{bmatrix} \dot{\varepsilon}^{\mathrm{p}}_{i+1} - \dot{\hat{\varepsilon}}^{\mathrm{p}}_{i+1/2} \\ -(\dot{\boldsymbol{X}}_{i+1} - \dot{\hat{\boldsymbol{X}}}_{i+1/2}) \end{bmatrix} - H^{-} \begin{bmatrix} \sigma_{i+1} - \hat{\sigma}_{i+1/2} \\ \boldsymbol{Z}_{i+1} - \hat{\boldsymbol{Z}}_{i+1/2} \end{bmatrix} = \mathbf{0}.$$

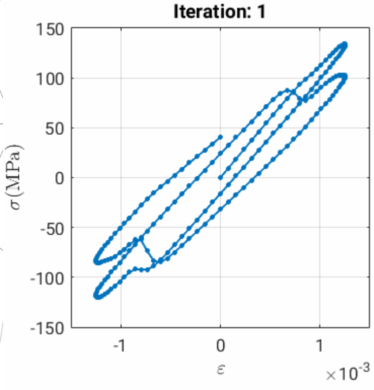
- Solve the equilibrium equation $\rightarrow$ PGD
- Integrate iteratively over time and space domains

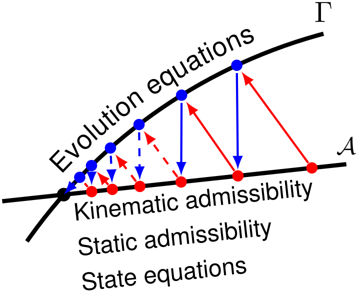
$$\Delta\varepsilon^{\mathrm{p}}(x,t) \approx \sum_{i=1}^{n} \bar{\varepsilon}^{\mathrm{p}}(x)\ \lambda(t)$$

- Update time functions
- If needed add only one pair per iteration



LATIN iteration





Established for viscoplasticity, contact and large deformations problems [ref]

Incorporating damage

State equations: $\sigma = E\,(1-D)\,\varepsilon^{\mathrm{e}}$

Nonlinear admissibility manifold

$$\dot{\varepsilon^{\mathrm{p}}} = \dot{\varepsilon} - \left(\frac{\dot{\sigma}}{E\,(1-D)} + \sigma\,\frac{\dot{D}}{E\,(1-D)^2} \right).$$

Work around

Linearise the admissibility manifold

by solving $\sigma = E\,(1-D)\,\varepsilon^{\mathrm{e}}$ locally

Incorporating damage

$$\varepsilon^{\mathrm{e}} - \hat{\varepsilon}^{\mathrm{e}} = E^{-1}\,(\sigma - \hat{\sigma})$$

$$\Delta\sigma = \sigma_{i+1} - \sigma_i = E\,\Delta\varepsilon^{\mathrm{e}} - \Delta R$$

$$\Delta R = (\sigma_i - \sigma_{i+1/2}) - E\,(\varepsilon^{\mathrm{e}}_i - \varepsilon^{\mathrm{e}}_{i+1/2})$$

Introduce

$$\Delta\sigma = \Delta\sigma' + \Delta\tilde{\sigma} \qquad \Delta\varepsilon = \Delta\varepsilon' + \Delta\tilde{\varepsilon}$$

$$\Delta\sigma' + \Delta\tilde{\sigma} = E\,(\Delta\varepsilon' - \Delta\varepsilon^{\mathrm{p}}) + E\,(\Delta\tilde{\varepsilon} - E^{-1}\Delta R)$$

PGD

$$\Delta\varepsilon^{\mathrm{p}}(x,t) \approx \sum_{i=1}^{n} \bar{\varepsilon}^{\mathrm{p}}(x)\,\lambda(t) \qquad \Delta\sigma(x,t) \approx \sum_{i=1}^{n} \bar{\sigma}(x)\,\lambda(t)$$

Numerical example

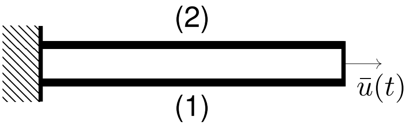
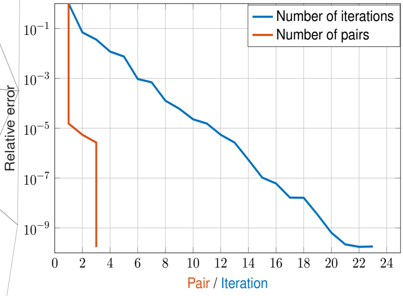
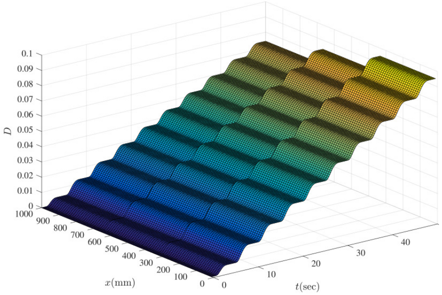


Figure: Two bars undergoing a cyclic loading.

Numerical example

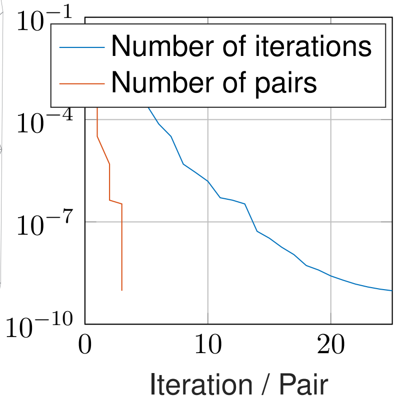


Error evolution

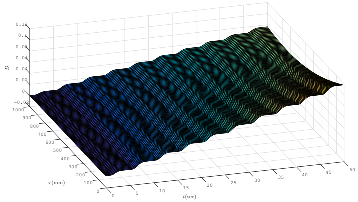


Damage evolution

Numerical example

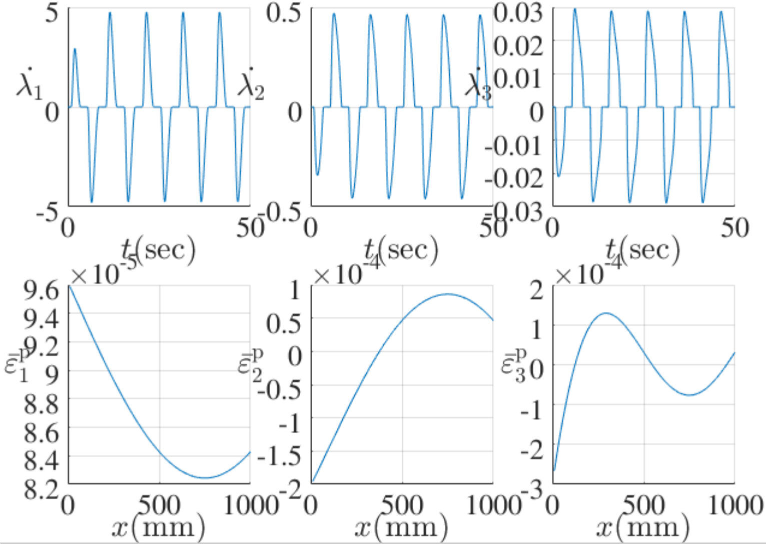


Error evolution

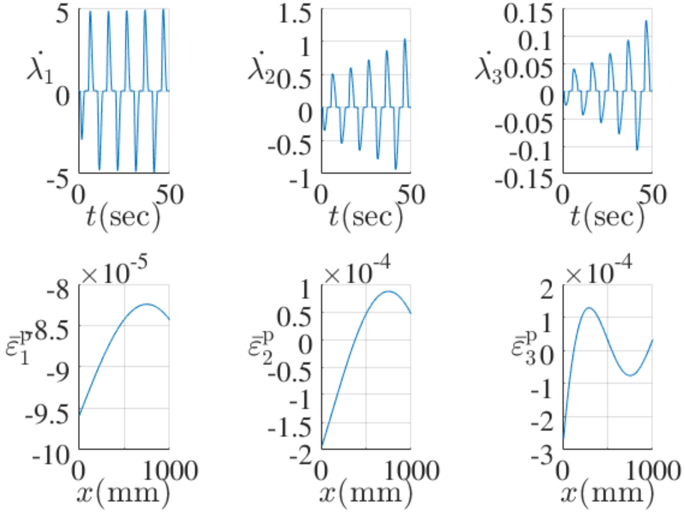


Damage evolution

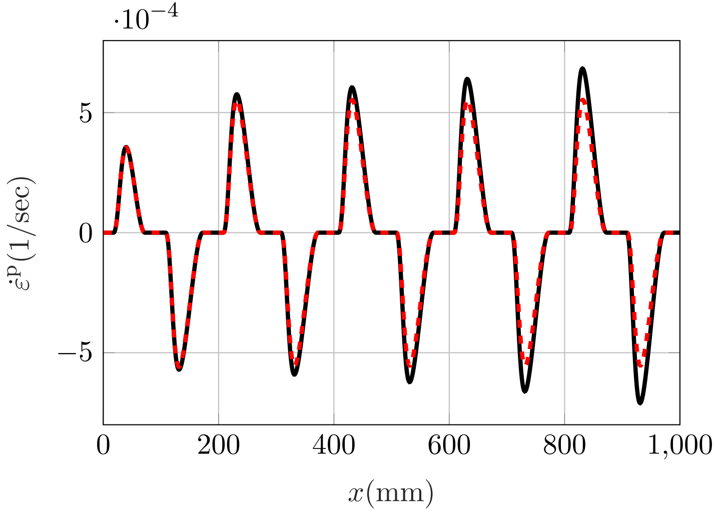
Numerical example



Numerical example



Numerical example



Conclusion

- We have a MOR to compute fatigue damage

Future steps

- Develop MOR in time
- Different amplitudes, frequencies and random loadings

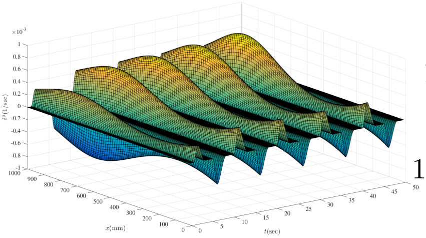
Thank you for your attention!

Model order reduction

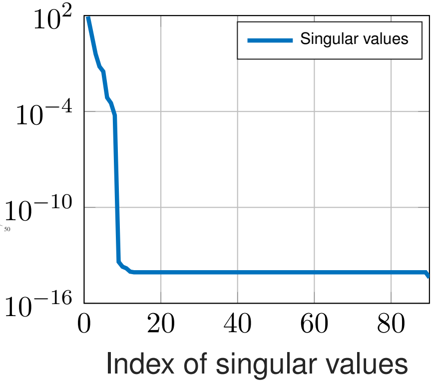
High fidelity problem $\rightarrow$

the plastic strain evolution

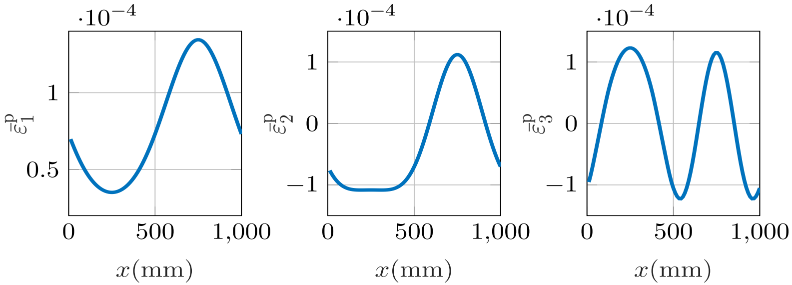
Singular Values Decomposition



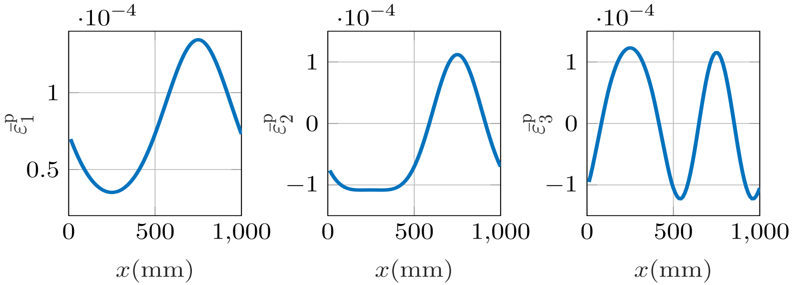
Given solution



Model order reduction



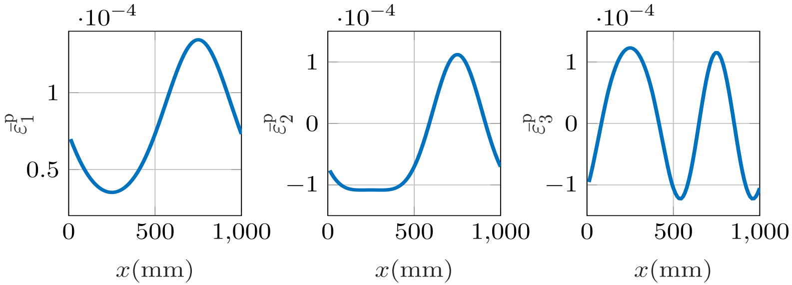
Model order reduction



POD → Solve to get the time functions

- Cheap online stage but the space functions are fixed.

Model order reduction



POD → Solve to get the time functions

- Cheap online stage but the space functions are fixed.

Could we generate space and time functions on the fly?

Model order reduction

Separation of variables [Proper Generalised decomposition (PGD), Ladevèze 1999]

The quantities of interest are defined over the whole time-space domain

$$\varepsilon^{\mathrm{p}}(x,t) \approx \sum_{i=1}^{n} \bar{\varepsilon}^{\mathrm{p}}(x)\ \lambda(t)$$

- Intrusive method
- Integrals over the generalised coordinates

Need for a convenient framework to utilise PGD