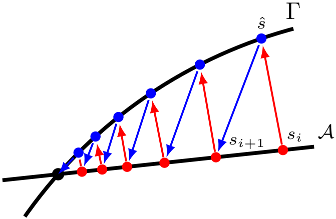
Model order reduction for fatigue analysis

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

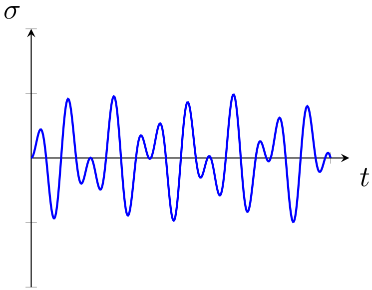
08. August 2018



Deutsche Forschungsgemeinschaft
IRTG-1627

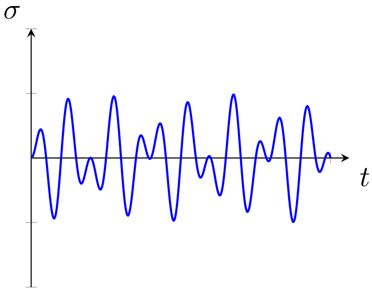
Fatigue damage

- Fluctuating loads



Fatigue damage

- Fluctuating loads

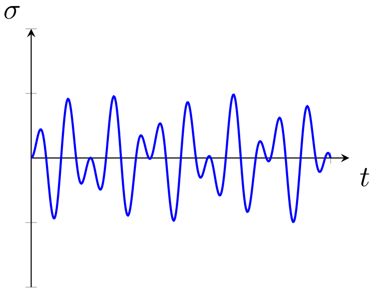


- Material degradation

Image by alegri / 4freephotos.com

Fatigue damage

- Fluctuating loads

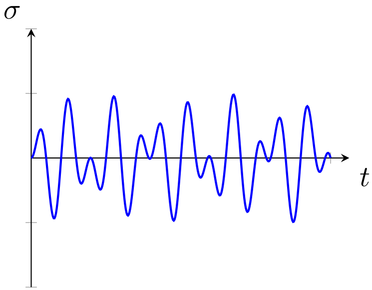


- Material degradation

Image by alegri / 4freephotos.com

Fatigue damage

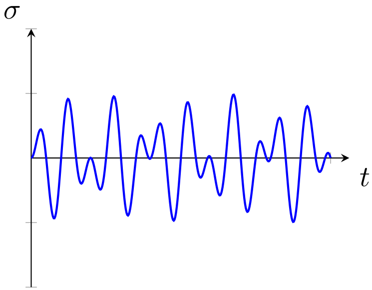
- Fluctuating loads



- Virtual experiments
- Continuum damage model
- Millions of cycles
- Macro crack initiation
- Computationally expensive

Fatigue damage

- Fluctuating loads



- Virtual experiments
- Continuum damage model
- Millions of cycles
- Macro crack initiation
- Computationally expensive

Model order reduction (MOR) techniques

State of the art

- An approach to include damage in a LATIN-PGD framework ☺
- Two-time scale approach to tackle large number of cycles ☺

- Limited to specific damage models ☹
- Inefficient for high damage values ☹
- Not easy to include variable amplitude loads ☹
  - without the two-time scale:
    - no time savings and many modes are generated

Goals

- Generalised formulation for different nonlinear material models
    - Tackle high damage values
    - Efficient in comparison with classical damage approaches
- Variable amplitude loading
- Decouple the problem dimensionality from the high fidelity one

Mechanical problem

Equilibrium equation

- Static admissibility

$$\nabla\cdot\boldsymbol{\sigma} + \boldsymbol{b} = \boldsymbol{0} \text{ in } \Omega$$

$$u = \bar{u} \text{ in } \partial\Omega_{\mathrm{D}}$$

- Kinematic admissibility

$$\boldsymbol{\varepsilon} = \nabla^{\mathrm{s}} u$$

Nonlinear material model

- State equations

$$\boldsymbol{\sigma} = \frac{\partial\psi}{\partial\boldsymbol{\varepsilon}^{\mathrm{e}}}$$

$$\boldsymbol{\beta} = \frac{\partial\psi}{\partial\boldsymbol{\alpha}}$$

$$Y = -\frac{\partial\psi}{\partial D}$$

- Evolution equations

$$\dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} = \frac{\partial\phi^{\mathrm{p}}}{\partial\sigma}$$

$$\dot{\alpha} = -\frac{\partial\phi^{\mathrm{p}}}{\partial\beta}$$

$$\dot{D} = \frac{\partial\phi^{\mathrm{d}}}{\partial D}$$

Solution algorithm

- Many LATIN algorithms since 1985
- A combination with some modifications
  - First stage (local)
    - Solve all local equations (possibly in parallel), given initial conditions
  - Second stage (global)
    - Solve all global equations
  - Data flow between these stages

$$(\hat{\boldsymbol{\sigma}} - \boldsymbol{\sigma}_i) + \mathbb{H}^+ (\hat{\boldsymbol{\varepsilon}} - \boldsymbol{\varepsilon}_i) = \mathbf{0}$$

$$(\boldsymbol{\sigma}_{i+1} - \hat{\boldsymbol{\sigma}}) - \mathbb{H}^- (\boldsymbol{\varepsilon}_{i+1} - \hat{\boldsymbol{\varepsilon}}) = \mathbf{0}$$

  - Iterate until convergence

The global stage

- The solution is approximated by a finite sum of separated functions (low-rank approximation)

$$u(x,t) = \sum_{i=1}^{N} \lambda_i(t)\ \bar{u}_i(x)$$

- Static admissibility

$$\left(\int_{[0,T]} \lambda^2 \,\mathrm{d}t \right) \int_{\Omega} \nabla v^* \mathbb{C} \nabla v \ \mathrm{d}\Omega = - \int_{\Omega} \nabla v^* \left(\int_{[0,T]} \lambda\, \hat{f} \,\mathrm{d}t \right) \mathrm{d}\Omega$$

$$\int_{[0,T]} \lambda^* \left(\int_{\Omega} \nabla v\ \mathbb{C}\ \nabla v \,\mathrm{d}\Omega \right) \lambda\ \mathrm{d}t = - \int_{[0,T]} \lambda^* \left(\int_{\Omega} \nabla v\ \hat{f} \,\mathrm{d}\Omega \right) \mathrm{d}t$$

Algorithmic point of view

- One cycle

$$\gamma \, \underline{\underline{K}} \, \underline{v} = \underline{F} \qquad (n \times n)$$

$$\underline{\underline{A}} \, \underline{\lambda} = \underline{b} \qquad (n_t \times n_t)$$

- Multiple cycles with variable load amplitudes
  - The initial stiffness
  - The elastic solution
  - An initial guess

Algorithmic point of view

- One cycle

$$\gamma \, \underline{\underline{K}} \, \underline{v} = \underline{F} \qquad (n \times n)$$

$$\underline{\underline{A}} \, \underline{\lambda} = \underline{b} \qquad (n_t \times n_t)$$

- Multiple cycles with variable load amplitudes
  - The initial stiffness

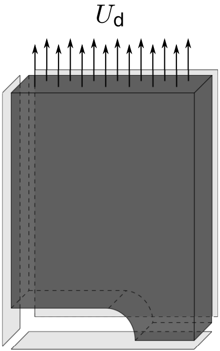
    compute only once
  - The elastic solution

    compute only once
  - An initial guess

    from the previous cycle

Numerical results

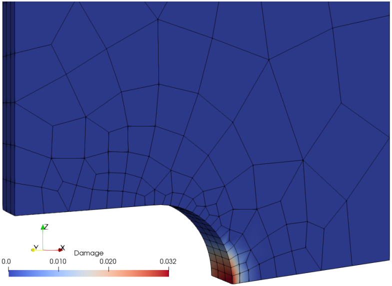
- Validation



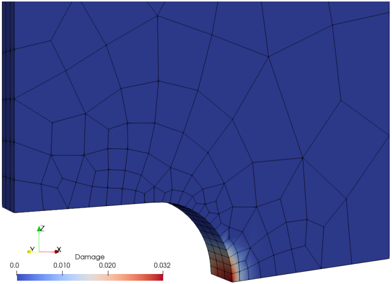
A plate with a central groove subjected to cyclic loading

(Cr-Mo steel at $580°\mathrm{C}$)

Numerical results



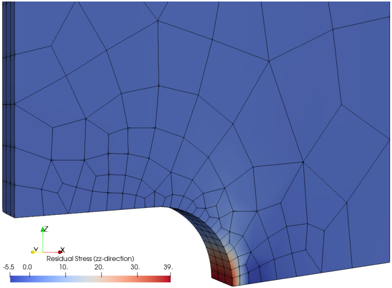
Newton-Raphson



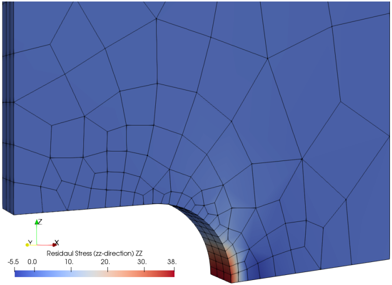
LATIN-PGD

Damage contour after at the last time step

Numerical results



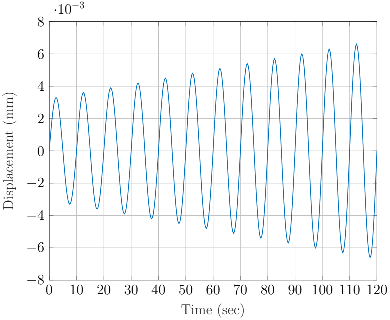
Newton-Raphson

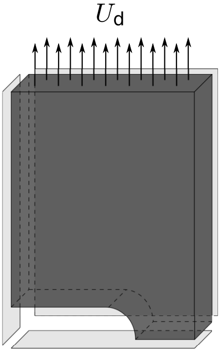


LATIN-PGD

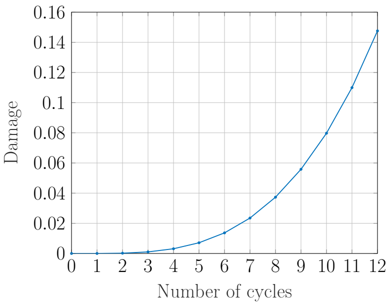
Residual stress distribution after removing the load

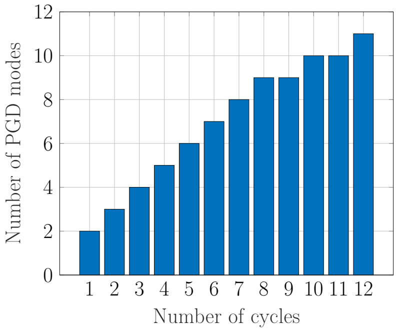
Variable amplitude



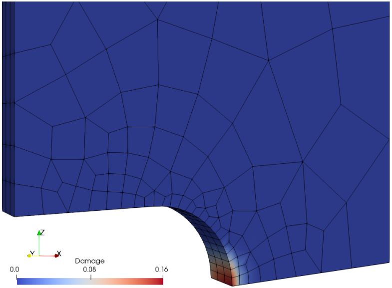


Variable amplitude

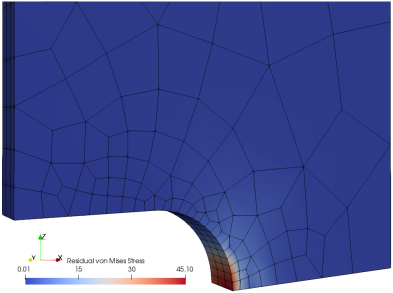




Variable amplitude

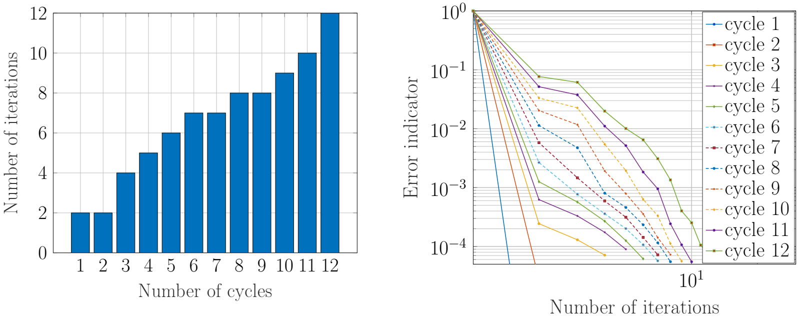


Damage distribution

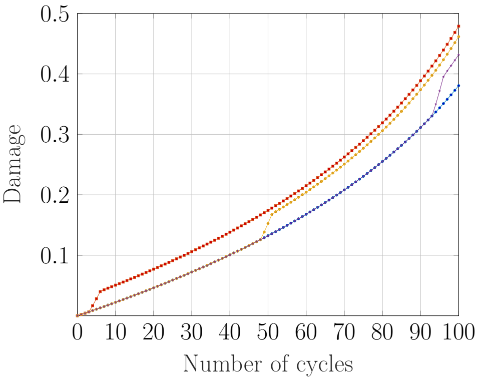


Von Mises stress distribution

Variable amplitude

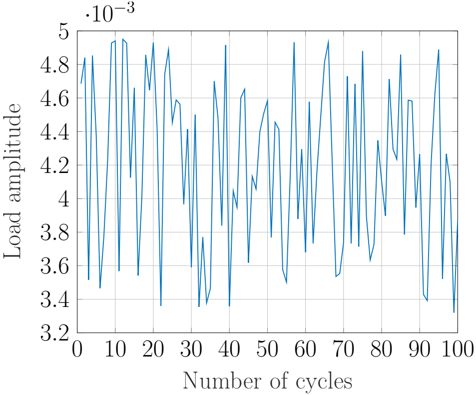


Overloads

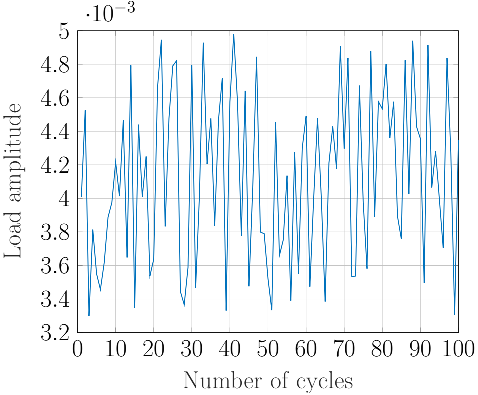


0.5
0.4
0.3
0.2
0.1
Damage
0
10
20
30
40
50
60
70
80
90
100
Number of cycles

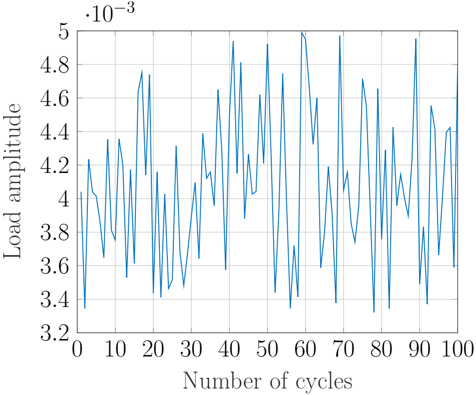
Random amplitudes



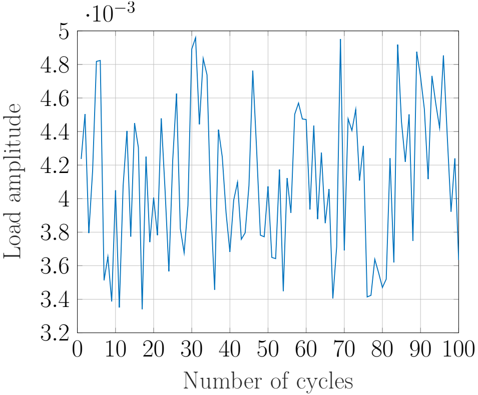
Random amplitudes



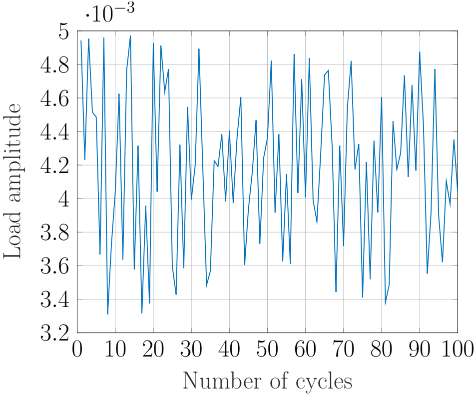
Random amplitudes



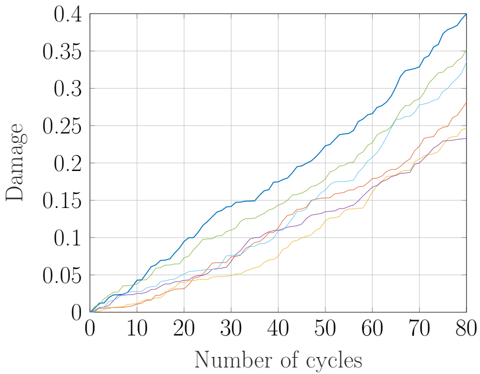
Random amplitudes



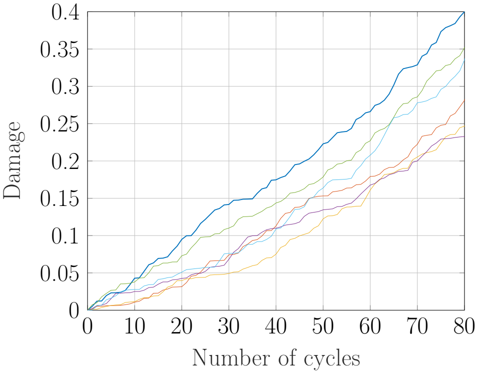
Random amplitudes



Random amplitudes



Random amplitudes



Speed-up factor $\sim$ 50

Conclusion and future research

- Efficient cycle by cycle simulation
- Works for LCF and HCF

- POD approach
- Time adaptivity
- Modes selection
- Faster orthogonalisation
- Hyper-reduction

Conclusion and future research

- Efficient cycle by cycle simulation
- Works for LCF and HCF

- POD approach
- Time adaptivity
- Modes selection
- Faster orthogonalisation
- Hyper-reduction

Thank you for your attention