CHALMERS
UNIVERSITY OF TECHNOLOGY

Numerical model reduction for FE-analysis of the viscoplasticity problem

Master's thesis in Applied Mechanics

DANIEL HÅRD
HENRIK SVENSSON

DEPARTMENT OF INDUSTRIAL AND MATERIALS SCIENCE

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

Master's thesis 2021

Numerical model reduction for FE-analysis of the viscoplasticity problem

DANIEL HÅRD
HENRIK SVENSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Industrial and Materials Science
Division of Material and Computational Mechanics
Chalmers University of Technology
Gothenburg, Sweden 2021

Numerical model reduction for FE-analysis of the viscoplasticity problem
DANIEL HÅRD
HENRIK SVENSSON

Supervisors: Fredrik Larsson, Kenneth Runesson and Fredrik Ekre, CHALMERS, IMS
Ralf Jänicke, TU Braunschweig

Examiner: Fredrik Larsson, IMS

Master's Thesis 2021
Department of Industrial and Materials Science
Division of Material and Computational Mechanics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: The first three approximation modes for viscoplastic strain component $\boldsymbol{\epsilon}_{11}^{\text{vp}}$ in a cantilever beam.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2021

Numerical model reduction for FE-analysis of the viscoplasticity problem
DANIEL HÅRD
HENRIK SVENSSON
Department of Industrial and Materials Science
Chalmers University of Technology

Abstract

Some numerical problems require, despite modern computational power, a lot of time to solve. The nonlinear viscoplasticity model is one such problem, where the plasticity in a material depends on the rate at which the load is applied. A method for reducing the system and decreasing the simulation time would therefore be an advantage.

This thesis aims to adopt one such method to reduce the computational cost for the viscoplasticity problem and evaluate it for some test cases. A mixed weak form together with the Finite Element Method (FEM) on monolithic form is established. Thereby, displacements and viscoplastic strains are solved for simultaneously rather than in the standard nested fashion. Proper Orthogonal Decomposition (POD) is performed on snapshots of the viscoplastic strains from a set of finite element training simulations carried out in an offline phase. The Nonuniform Transformation Field Analysis (NTFA) approach expresses the displacements in a corresponding reduced basis. The numerical computations have been implemented in Julia and tested in 2D for varying load combinations.

It was shown that it was possible to reduce the solve time and still obtain good approximations of the solution. However, there is a crucial dependency on the training, with higher accuracy for targeted simulations similar to the training. Still, the robustness of the procedure was illustrated by near monotonic error convergence. Although more research would be needed, the results show promise for the development of highly efficient approximations of the viscoplasticity problem.

Keywords: viscoplasticity, finite element method, numerical model reduction, proper orthogonal decomposition

Preface and acknowledgements

This thesis is the final part of the master's programme Applied Mechanics at Chalmers University of Technology. It was carried out during the spring semester of 2021 at the division of Material and Computational Mechanics, department of Industrial and Materials Science.

We would like to thank our main supervisor Fredrik Larsson for his excellent guidance and support throughout the project. We would also like to thank our co-supervisors Kenneth Runesson, Ralf Jänicke and Fredrik Ekre for their involvement and key insights. They have all provided good explanations for different terms and methods. Fredrik Ekre has introduced us to the programming language Julia and provided help when we had questions. All of them have shown a big interest in the progression of the thesis and helped us many times to reach an understanding to move forward.

Daniel and Henrik, Gothenburg, June 2021

Contents

Nomenclature

Abbreviation

BC	Boundary Condition
CST	Constant Strain Triangle
FEM	Finite Element Method
LST	Linear Strain Triangle
MOR	Model Order Reduction
NMR	Numerical Model Reduction
NTFA	Nonuniform Transformation Field Analysis
PGD	Proper Generalized Decomposition
POD	Proper Orthogonal Decomposition
RMS	Root mean square
ROM	Reduced Order Modelling
RVE	Representative Volume Element
TFA	Transformation Field Analysis

Numerical model reduction

$\hat{\boldsymbol{\epsilon}}^{\mathrm{vp}}$	Reduced basis for viscoplastic strains
$\hat{\boldsymbol{u}}$	Reduced basis for inelastic part of displacements
$\boldsymbol{u}_0$	Elastic part of the displacements
ξ	Mode activity function
g	Correlation matrix

Solid Mechanics

$\boldsymbol{\epsilon}$	Total strain
$\boldsymbol{\epsilon}^{\mathrm{e}}$	Elastic strain
Γ	Boundary
Ω	Domain

$\boldsymbol{\sigma}$	Stress
$\boldsymbol{u}$	Displacement
$\boldsymbol{b}$	Body force
$\boldsymbol{t}$	Traction
G	Shear modulus
K	Bulk modulus

Tensors

E	4th order elastic stiffness tensor
$\mathsf{I}^{\text{sym}}_{\text{dev}}$	4th order deviatoric minor-symmetric identity tensor
I^{sym}	4th order minor-symmetric identity tensor
$\boldsymbol{I}$	2nd order identity tensor

Viscoplasticity

$\boldsymbol{\epsilon}^{\text{vp}}$	Viscoplastic strain
$\eta[\Phi]$	Overstress function
λ	Plastic multiplier
σ_e	Von Mises yield effective stress
σ_y	Yield stress
Φ	Yield function
t^*	natural relaxation time

List of Figures

List of Tables

1

Introduction

1.1 Background

Modern computational power has meant that more complex problems can be investigated numerically than was previously possible. Despite this, there is still a need to speedup the numerical analyses for many problems in science and engineering through model reduction, e.g., multi-query problems such as design and optimization [1]. In these cases, a large number of solutions are traditionally needed for varying the model parameters. Applying model reduction to design optimization was done by, e.g, Amsallem et al. [2]. Another important application in numerical analysis, where speedup is necessary, is microscale analysis. This is when the response from Representative Volume Elements (RVE) on the microscale is sought for varying macroscopic loading, see, e.g, Jänicke et al. [3] and Ekre et al. [4].

In linear structural dynamics, it is very common to adopt model reduction in terms of modal superposition [5]. In its most standard format, the original finite element problem is replaced by a small set of decoupled scalar equations for the predominant displacement modes. For nonlinear problems, a similar approach can be adopted. The degrees of freedom for the system can be reduced by considering only the most important modes of the solution. However, for a nonlinear problem the procedure for computing the mode shapes becomes more involved. Furthermore, the reduced system will be of that of a fully coupled nonlinear set of equations that needs to be solved iteratively.

Reducing a complicated model by introducing a reduced basis and solving for fewer unknowns will in this thesis be called Numerical Model Reduction (NMR), in the literature also called Reduced Order Modelling (ROM) and Model Order Reduction (MOR). The reason for using the notation NMR is to highlight that numerical methods are used to obtain the reduced basis, instead of changing the underlying model. A general method to obtain a reduced basis, applicable to nonlinear problems, is Proper Orthogonal Decomposition (POD) and was used for elastic-viscoplastic composites by Roussette et al. [6]. Materials consisting of one or multiple phases can be assumed to have a uniform basis function in each phase, called Transformation Field Analysis (TFA), a concept introduced by Dvorak and Benveniste [7]. A similar approach, Nonuniform Transformation Field Analysis (NTFA), was introduced by Michel and Suqet [8] as a more general approach to TFA, where the basis functions can be nonuniform instead. Combining these two methods, NTFA and POD, was

studied by Fritzen et al. [9] for viscoelasticity in composites and by Jänicke et al. [3] for poroelasticity.

This thesis considers model reduction applied to viscoplasticity. Viscoplasticity is a type of plasticity model that is rate-dependent, meaning the deformations depends on the rate at which the loads are applied [10]. It can for example be applied to metals, especially at higher temperatures [11]. This material model is highly non-linear and solving for it can take a considerable time, therefore is reducing the solve time of interest.

Some methods, not investigated in this thesis, but still of relevance in the broader research into model reduction are Proper Generalized Decomposition (PGD), hyperreduction and error control. PGD is a related method to POD, but with some big differences [12]. It decomposes a function as a sum of products between simpler component functions. This allows, e.g., boundary conditions and material properties to be treated as coordinates in the problem and therefore accounted for in the decomposition. Hyper-reduction reduces the solve time for the model reduction further by only looking at a subset of the quadrature points when assembling the reduced system, see Ryckelynck [13]. Finally, for a reduction method to be reliably used, which would be necessary for industrial use, an error estimation needs to be established for the model reduction of the problem, see Ekre et al. [4]. This would then mean error control could be used in the solution procedure.

1.2 Aim and scope

The aim with this thesis is to apply Numerical Model Reduction (NMR) using Proper Orthogonal Decomposition (POD) to viscoplasticity. The reduction method will be evaluated for some numerical examples in terms of solve time, robustness, and solution accuracy.

Specific goals are to:

- Formulate viscoplasticity in mixed weak form
- Establish NMR problem
- Implement the numerical procedure
- Design a test case for evaluating the performance
- Analyse the performance in terms of accuracy, robustness, and computational cost.

The material is assumed to be isotropic and deformations limited to small strains in a quasi-static process. Furthermore, the displacements are also assumed to be two dimensional, using the plane strain condition.

2

FE-analysis of viscoplasticity

In this chapter, the theory for viscoplasticity is presented, specifically perfect viscoplasticity, meaning no hardening or softening. First the constitutive model for viscoplasticity is presented. Then the equations on strong and mixed weak form. Then the 2D model used and lastly the FE-format of the equations with a monolithic approach. The assumptions made are small strain, plane strain, isotropic material and a quasi-static process.

2.1 Viscoplasticity

A material exposed to loads will experience a certain stress level. If the stress is low the response will be elastic, meaning that the material will revert back to it's initial state when the loads are removed. If the stress is high enough the material will start to plasticly deform, meaning that when the loads are removed it will not revert back to the initial state [11]. Instead it will end up in a new state, different from the initial.

In the viscoplastic model, the point at which the material starts to plasticly deform is when the effective stress σ_e is above the yield stress σ_y. Also important with the viscoplastic model is that it is rate-dependent, as showed in Figure 2.1. A very slow loading will closely follow the lower curve, which is the case of perfect plasticity. Increasing the strain rate will increase the stress above the initial yield stress σ_y. The higher the strain rate is, the higher the stress will be. A description of viscoplasticity can be found in [10] and references therein. Viscoplasticity can to some extent be applied for most metals, especially at temperatures higher than about a third of the material's melting point [11].

2.1.1 Yield function

The quasi-static yield function with no hardening in a multi-axial case is

$$\Phi = \sigma_e - \sigma_y, \quad \text{with} \quad \sigma_e = \sqrt{\frac{3}{2}}|\boldsymbol{\sigma}_{\text{dev}}| \tag{2.1}$$

where σ_e is the von Mises yield effective stress. Note that if $\Phi < 0$, the material will have an elastic response and if $\Phi > 0$, the response will be plastic.

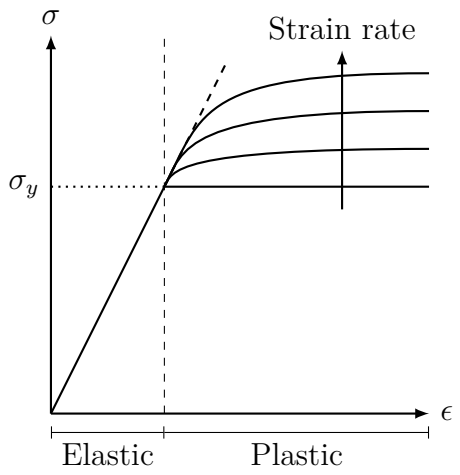


Figure 2.1: Illustrative example of the stress-strain curve for viscoplasticity. A higher strain rate will, when in the plastic region, increase the stress level in the material. A very slow loading will result in perfect plasticity, which is the lowest curve.

2.1.2 Perzyna formulation

The Perzyna formulation of the evolution of the viscoplastic strain $\boldsymbol{\epsilon}^{\text{vp}}$ is the flow rule

$$\dot{\boldsymbol{\epsilon}}^{\text{vp}} = \boldsymbol{g} = \lambda \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} = \lambda \boldsymbol{f}$$

where the plastic multiplier λ in the Perzyna formulation is

$$\lambda = \frac{1}{t^*} \eta[\Phi] \geq 0$$

where t^* is the natural relaxation time and $\eta[\Phi]$ a nondimensional overstress function.

$$\eta[\Phi] = \begin{cases} 0 & \text{if } \Phi < 0 \\ > 0 & \text{if } \Phi > 0 \end{cases}$$

If the yield function Φ is on the form as in Equation 2.1 then the flow direction $\boldsymbol{f}$ becomes

$$\boldsymbol{f} = \frac{3}{2} \frac{\boldsymbol{\sigma}_{\text{dev}}}{\sigma_e}$$

which leads to $\boldsymbol{\epsilon}^{\text{vp}}$ being deviatoric through the flow rule.

2.1.3 Norton model

The Norton model for the overstress function $\eta[\Phi]$ is

$$\eta[\Phi] = \left(\frac{\langle \Phi \rangle}{\sigma_c} \right)^{n_c} \geq 0$$

where n_c and σ_c both are parameters. $\langle\rangle$ denotes the Macaulay brackets, which is defined as

$$\langle x \rangle = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$

in the Perzyna formulation this leads to

$$\boldsymbol{g} = \frac{3}{2t^*} \left(\frac{\langle \sigma_e - \sigma_y \rangle}{\sigma_c} \right)^{n_c} \frac{\boldsymbol{\sigma}_{\text{dev}}}{\sigma_e} \tag{2.2}$$

and

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} = \begin{cases} 0 & \sigma_e \leq \sigma_y \\ \dfrac{3(\sigma_e - \sigma_y)^{n_c}}{2t^* \sigma_e \sigma_c^{n_c}} \left[\mathbf{I}^{\text{sym}}_{\text{dev}} + \dfrac{3}{2\sigma_e^2} \left(\dfrac{n_c \sigma_e}{\sigma_e - \sigma_y} - 1 \right) \boldsymbol{\sigma}_{\text{dev}} \otimes \boldsymbol{\sigma}_{\text{dev}} \right] & \sigma_e > \sigma_y \end{cases} \tag{2.3}$$

where the derivation of $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}}$ can be found in Appendix A.1.

2.1.4 Bingham model

Setting $n_c = 1$ and $\sigma_c = 3G$ in the Norton model results in the simpler Bingham model

$$\eta[\Phi] = \frac{\langle \Phi \rangle}{3G}.$$

Inserting the values on n_c and σ_c into Equations 2.2 and 2.3 results in the corresponding $\boldsymbol{g}$ and $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}}$ respectively.

2.2 Strong form

Assuming a quasi-static process, the governing equations are [14]

$$\begin{cases} -\boldsymbol{\sigma} \cdot \boldsymbol{\nabla} = \boldsymbol{b} & \text{in } \Omega \\ \boldsymbol{u} = \boldsymbol{u}_{\text{P}} & \text{on } \Gamma_{\text{D}} \\ \boldsymbol{t} = \boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{t}_{\text{P}} & \text{on } \Gamma_{\text{N}} \end{cases} \tag{2.4}$$

where the loads and boundary conditions (BC) can be seen represented in Figure 2.2 on the so-called continuum potato. The stress $\boldsymbol{\sigma}$ depends on the elastic strain $\boldsymbol{\epsilon}^{\text{e}}$, which in turn depends on the total strain $\boldsymbol{\epsilon}[\boldsymbol{u}]$ and the viscoplastic strain $\boldsymbol{\epsilon}^{\text{vp}}$

$$\boldsymbol{\sigma} = \mathbf{E} : \boldsymbol{\epsilon}^{\text{e}} = \mathbf{E} : (\boldsymbol{\epsilon}[\boldsymbol{u}] - \boldsymbol{\epsilon}^{\text{vp}})$$

where the assumption of small strain has been made and $\mathbf{E}$ is the fourth order isotropic symmetric elastic stiffness tensor defined by

$$\mathbf{E} = 2G\mathbf{I}^{\text{sym}}_{\text{dev}} + K\boldsymbol{I} \otimes \boldsymbol{I}$$

where G is the shear modulus and K the bulk modulus. The 4th order deviatoric minor-symmetric identity tensor $\mathbf{I}^{\text{sym}}_{\text{dev}}$, the 4th order minor-symmetric identity tensor $\mathbf{I}^{\text{sym}}$, and the 2nd order identity tensor $\boldsymbol{I}$ are given by

$$\begin{aligned}\mathbf{I}^{\text{sym}}_{\text{dev}} &:= \mathbf{I}^{\text{sym}} - \frac{1}{3}\boldsymbol{I}\otimes\boldsymbol{I}\\ \mathbf{I}^{\text{sym}} &:= \frac{1}{2}[\delta_{ik}\delta_{jl}+\delta_{il}\delta_{jk}]\boldsymbol{e}_i\otimes\boldsymbol{e}_j\otimes\boldsymbol{e}_k\otimes\boldsymbol{e}_l\\ \boldsymbol{I} &:= \delta_{ij}\boldsymbol{e}_i\otimes\boldsymbol{e}_j.\end{aligned}$$

The viscoplastic strain is an effect of the material plasticly deforming for large loading. For a viscoplastic model, the plastic deformation is rate-dependent and can be described by the viscoplastic flow rule on the form

$$\dot{\boldsymbol{\epsilon}}^{\text{vp}} = \boldsymbol{g}(\boldsymbol{\sigma}(\boldsymbol{\epsilon}[\boldsymbol{u}] - \boldsymbol{\epsilon}^{\text{vp}})) \quad \text{in } \Omega \tag{2.5}$$

where $\boldsymbol{\epsilon}^{\text{vp}} = \mathbf{0}$ in the initial state. See Section 2.1 for more details on $\boldsymbol{g}$.

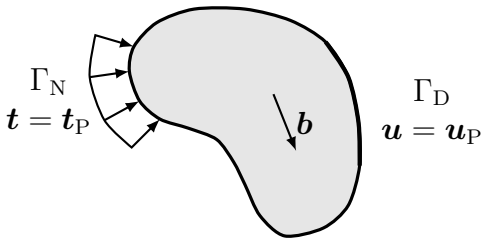


Figure 2.2: The continuum potato with boundary conditions and loads.

2.3 Mixed weak form

The mixed weak form can be derived by multiplying the Equations 2.4 and 2.5 on strong form with $\delta\boldsymbol{u}$ and $\delta\boldsymbol{\epsilon}^{\text{vp}}$ respectively and integrating over the domain Ω. Then applying Gauss theorem to the first equation resulting in [14]

$$\int_\Omega \boldsymbol{\epsilon}[\delta\boldsymbol{u}] : \boldsymbol{\sigma}\,\mathrm{d}\Omega = \int_\Omega \delta\boldsymbol{u}\cdot\boldsymbol{b}\,\mathrm{d}\Omega + \int_{\partial\Omega}\delta\boldsymbol{u}\cdot\boldsymbol{t}\,\mathrm{d}\Gamma \tag{2.6}$$

$$\int_\Omega \delta\boldsymbol{\epsilon}^{\text{vp}} : \dot{\boldsymbol{\epsilon}}^{\text{vp}}\,\mathrm{d}\Omega = \int_\Omega \delta\boldsymbol{\epsilon}^{\text{vp}} : \boldsymbol{g}\,\mathrm{d}\Omega\,. \tag{2.7}$$

2.3.1 Residual

The residual of the equations is the difference between the left- and right-hand sides. This means that the residual is zero for a solution.

The residual $R^{\text{u}}(\boldsymbol{u}, \boldsymbol{\epsilon}^{\text{vp}}; \delta\boldsymbol{u})$ of Equation 2.6 is defined as

$$R^{\text{u}}(\boldsymbol{u}, \boldsymbol{\epsilon}^{\text{vp}}; \delta\boldsymbol{u}) := a^u(\boldsymbol{u}; \delta\boldsymbol{u}) - c(\boldsymbol{\epsilon}^{\text{vp}}; \delta\boldsymbol{u}) - l(\delta\boldsymbol{u}) \tag{2.8}$$

where

$$a^u(\boldsymbol{u};\delta\boldsymbol{u}) = \int_\Omega \boldsymbol{\epsilon}[\delta\boldsymbol{u}] : \mathbf{E} : \boldsymbol{\epsilon}[\boldsymbol{u}]\,\mathrm{d}\Omega$$
$$c(\boldsymbol{\epsilon}^{\mathrm{vp}};\delta\boldsymbol{u}) = \int_\Omega \boldsymbol{\epsilon}[\delta\boldsymbol{u}] : \mathbf{E} : \boldsymbol{\epsilon}^{\mathrm{vp}}\,\mathrm{d}\Omega$$
$$l(\delta\boldsymbol{u}) = \int_\Omega \delta\boldsymbol{u}\cdot\boldsymbol{b}\,\mathrm{d}\Omega + \int_{\partial\Omega}\delta\boldsymbol{u}\cdot\boldsymbol{t}\,\mathrm{d}\Gamma$$

The residual for equation 2.7 can be written on a similar format, starting with first rewriting the equations as

$$a^{\mathrm{vp}}(\dot{\boldsymbol{\epsilon}}^{\mathrm{vp}};\delta\boldsymbol{\epsilon}^{\mathrm{vp}}) - b(\boldsymbol{u},\boldsymbol{\epsilon}^{\mathrm{vp}};\delta\boldsymbol{\epsilon}^{\mathrm{vp}}) = 0.$$

where

$$a^{\mathrm{vp}}(\dot{\boldsymbol{\epsilon}}^{\mathrm{vp}};\delta\boldsymbol{\epsilon}^{\mathrm{vp}}) := \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \dot{\boldsymbol{\epsilon}}^{\mathrm{vp}}\,\mathrm{d}\Omega$$
$$b(\boldsymbol{u},\boldsymbol{\epsilon}^{\mathrm{vp}};\delta\boldsymbol{\epsilon}^{\mathrm{vp}}) := \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \boldsymbol{g}\,\mathrm{d}\Omega\,.$$

The time derivative is discretised according to the backward Euler method [15]

$$\dot{\boldsymbol{\epsilon}}^{\mathrm{vp}} = \frac{\boldsymbol{\epsilon}^{\mathrm{vp}} - {}^{(n)}\boldsymbol{\epsilon}^{\mathrm{vp}}}{\Delta t}$$
$$\Rightarrow a^{\mathrm{vp}}(\dot{\boldsymbol{\epsilon}}^{\mathrm{vp}};\delta\boldsymbol{\epsilon}^{\mathrm{vp}}) = \frac{1}{\Delta t}\Big(a^{\mathrm{vp}}(\boldsymbol{\epsilon}^{\mathrm{vp}};\delta\boldsymbol{\epsilon}^{\mathrm{vp}}) - a^{\mathrm{vp}}\Big({}^{(n)}\boldsymbol{\epsilon}^{\mathrm{vp}};\delta\boldsymbol{\epsilon}^{\mathrm{vp}}\Big)\Big)$$

where ${}^{(n)}\boldsymbol{\epsilon}^{\mathrm{vp}}$ is the viscoplastic strain from the previous time step and $\boldsymbol{\epsilon}^{\mathrm{vp}}$ is the viscoplastic strain in the current time step that is being solved for.

This leads then to

$$a^{\mathrm{vp}}(\boldsymbol{\epsilon}^{\mathrm{vp}};\delta\boldsymbol{\epsilon}^{\mathrm{vp}}) - a^{\mathrm{vp}}\Big({}^{(n)}\boldsymbol{\epsilon}^{\mathrm{vp}};\delta\boldsymbol{\epsilon}^{\mathrm{vp}}\Big) - \Delta t\, b(\boldsymbol{u},\boldsymbol{\epsilon}^{\mathrm{vp}};\delta\boldsymbol{\epsilon}^{\mathrm{vp}}) = 0$$

and the residual is defined as

$$R^{\boldsymbol{\epsilon}^{\mathrm{vp}}}(\boldsymbol{u},\boldsymbol{\epsilon}^{\mathrm{vp}};\delta\boldsymbol{\epsilon}^{\mathrm{vp}}) := a^{\mathrm{vp}}(\boldsymbol{\epsilon}^{\mathrm{vp}};\delta\boldsymbol{\epsilon}^{\mathrm{vp}}) - a^{\mathrm{vp}}\Big({}^{(n)}\boldsymbol{\epsilon}^{\mathrm{vp}};\delta\boldsymbol{\epsilon}^{\mathrm{vp}}\Big) - \Delta t\, b(\boldsymbol{u},\boldsymbol{\epsilon}^{\mathrm{vp}};\delta\boldsymbol{\epsilon}^{\mathrm{vp}}) \qquad (2.9)$$

A solution for $\boldsymbol{u}$ and $\boldsymbol{\epsilon}^{\mathrm{vp}}$ satisfies the system

$$R^{\mathrm{u}}(\boldsymbol{u},\boldsymbol{\epsilon}^{\mathrm{vp}};\delta\boldsymbol{u}) = 0$$
$$R^{\boldsymbol{\epsilon}^{\mathrm{vp}}}(\boldsymbol{u},\boldsymbol{\epsilon}^{\mathrm{vp}};\delta\boldsymbol{\epsilon}^{\mathrm{vp}}) = 0.$$

2.3.2 Linearisation

The linearisation of the the equations is done by taking the Gateaux derivatives of the residuals with respect to the two fields $\boldsymbol{u}$ and $\boldsymbol{\epsilon}^{\mathrm{vp}}$. The derivations of the Gateaux derivative of the residuals is done in the Appendix A.2. The resulting derivatives are presented after the Gateaux derivative is defined.

The Gateaux derivative of a functional F w.r.t. $\boldsymbol{u}$ in the direction $\mathrm{d}\boldsymbol{u}$ is defined as

$$F'(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \mathrm{d}\boldsymbol{u}) := \frac{\mathrm{d}F(\boldsymbol{u} + \gamma \mathrm{d}\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}})}{\mathrm{d}\gamma}\Big|_{\gamma=0}.$$

The Gateaux derivative of the residual R^{u} w.r.t $\boldsymbol{u}$ in the direction $\mathrm{d}\boldsymbol{u}$ is

$$R_u^{\mathrm{u}'}(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{u}, \mathrm{d}\boldsymbol{u}) = a^u(\delta\boldsymbol{u}, \mathrm{d}\boldsymbol{u}).$$

The Gateaux derivative of the residual R^{u} w.r.t $\boldsymbol{\epsilon}^{\mathrm{vp}}$ in the direction $\mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}$ is

$$R_{\boldsymbol{\epsilon}^{\mathrm{vp}}}^{\mathrm{u}'}(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{u}, \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}) = -c(\delta\boldsymbol{u}, \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}).$$

The Gateaux derivative of the residual $R^{\boldsymbol{\epsilon}^{\mathrm{vp}}}$ w.r.t $\boldsymbol{u}$ in the direction $\mathrm{d}\boldsymbol{u}$ is

$$R_u^{\boldsymbol{\epsilon}^{\mathrm{vp}\prime}}(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}}, \mathrm{d}\boldsymbol{u}) = -\Delta t \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} : \mathsf{E} : \boldsymbol{\epsilon}[\mathrm{d}\boldsymbol{u}]\, \mathrm{d}\Omega\,.$$

where $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}}$ can be seen in Equation 2.3.

The Gateaux derivative of the residual $R^{\boldsymbol{\epsilon}^{\mathrm{vp}}}$ w.r.t $\boldsymbol{\epsilon}^{\mathrm{vp}}$ in the direction $\mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}$ is

$$R_{\boldsymbol{\epsilon}^{\mathrm{vp}}}^{\boldsymbol{\epsilon}^{\mathrm{vp}\prime}}(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}}, \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}) = \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}\, \mathrm{d}\Omega + \Delta t \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} : \mathsf{E} : \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}\, \mathrm{d}\Omega$$

2.3.3 Newton's method

Newton's method is used to iteratively solve equations [14]. For the viscoplastic problem with the sought fields $\boldsymbol{u}$ and $\boldsymbol{\epsilon}^{\mathrm{vp}}$, the updates to the solutions are

$$\begin{aligned}\boldsymbol{u}^{(k+1)} &= \boldsymbol{u}^{(k)} + \Delta\boldsymbol{u} \\ \boldsymbol{\epsilon}^{\mathrm{vp}(k+1)} &= \boldsymbol{\epsilon}^{\mathrm{vp}(k)} + \Delta\boldsymbol{\epsilon}^{\mathrm{vp}}\end{aligned}$$

where $\Delta\boldsymbol{u}$ and $\Delta\boldsymbol{\epsilon}^{\mathrm{vp}}$ are obtained as solutions of the system

$$\begin{aligned} R_u^{\mathrm{u}'}\Big(\boldsymbol{u}^{(k)}, \boldsymbol{\epsilon}^{\mathrm{vp}(k)}; \delta\boldsymbol{u}, \Delta\boldsymbol{u}\Big) + R_{\boldsymbol{\epsilon}^{\mathrm{vp}}}^{\mathrm{u}'}\Big(\boldsymbol{u}^{(k)}, \boldsymbol{\epsilon}^{\mathrm{vp}(k)}; \delta\boldsymbol{u}, \Delta\boldsymbol{\epsilon}^{\mathrm{vp}}\Big) &= -R^{\mathrm{u}}\Big(\boldsymbol{u}^{(k)}, \boldsymbol{\epsilon}^{\mathrm{vp}(k)}; \delta\boldsymbol{u}\Big) \\ R_u^{\boldsymbol{\epsilon}^{\mathrm{vp}\prime}}\Big(\boldsymbol{u}^{(k)}, \boldsymbol{\epsilon}^{\mathrm{vp}(k)}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}}, \Delta\boldsymbol{u}\Big) + R_{\boldsymbol{\epsilon}^{\mathrm{vp}}}^{\boldsymbol{\epsilon}^{\mathrm{vp}\prime}}\Big(\boldsymbol{u}^{(k)}, \boldsymbol{\epsilon}^{\mathrm{vp}(k)}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}}, \Delta\boldsymbol{\epsilon}^{\mathrm{vp}}\Big) &= -R^{\boldsymbol{\epsilon}^{\mathrm{vp}}}\Big(\boldsymbol{u}^{(k)}, \boldsymbol{\epsilon}^{\mathrm{vp}(k)}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}}\Big). \end{aligned} \tag{2.10}$$

2.4 2 dimensional analysis

The plane strain condition is one form of the 2D formulation of the governing equations [16]. A slice of a long beam with uniform cross section is extracted, as in Figure 2.3. Loading is only applied in the transverse direction and not in the lengthwise direction. The two ends are also constrained in the lengthwise direction. This means that the lengthwise direction is fixed, and the assumption is that displacements only occur in the transverse directions. The displacements are then given by

$$\boldsymbol{u} = u_1\boldsymbol{e}_1 + u_2\boldsymbol{e}_2.$$

This leads to the out of plane components $\epsilon_{13} = \epsilon_{23} = \epsilon_{33} = 0$. This also implies that $\sigma_{13} = \sigma_{23} = 0$, but importantly is $\sigma_{33} \neq 0$ [16]. This is because the constraint $\epsilon_{33} = 0$ needs to be maintained. Assuming isotropy leads to

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_{11} & \epsilon_{12} & 0 \\ \epsilon_{12} & \epsilon_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & 0 \\ \sigma_{12} & \sigma_{22} & 0 \\ 0 & 0 & \sigma_{33} \end{bmatrix}$$

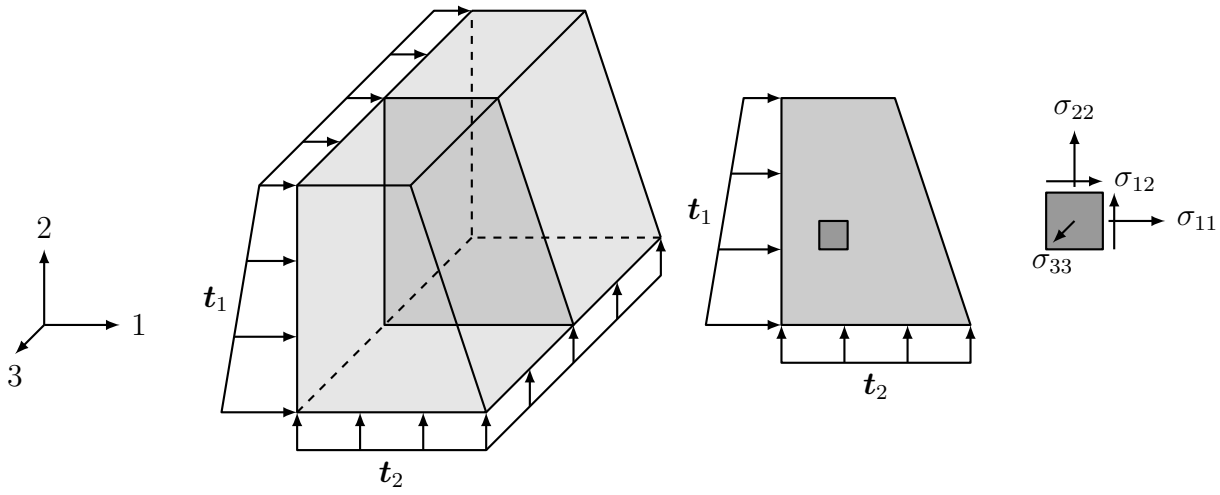


Figure 2.3: The plane strain condition is a 2D slice in a long beam with both short sides constrained in the lengthwise direction. Three stress components will be in the plane of the slice and one component orthogonal to the plane.

It is important to note that these assumptions are only valid for the total strain $\boldsymbol{\epsilon}$. The viscoplasticity model described in Section 2.1 is deviatoric and therefore must the trace of $\boldsymbol{\epsilon}^{\text{vp}}$ be zero. This means that the viscoplastic strain is

$$\boldsymbol{\epsilon}^{\text{vp}} = \begin{bmatrix} \epsilon_{11}^{\text{vp}} & \epsilon_{12}^{\text{vp}} & 0 \\ \epsilon_{12}^{\text{vp}} & \epsilon_{22}^{\text{vp}} & 0 \\ 0 & 0 & -(\epsilon_{11}^{\text{vp}} + \epsilon_{22}^{\text{vp}}) \end{bmatrix}$$

and, hence, the elastic strain component $\boldsymbol{\epsilon}_{33}^{\text{e}} \neq 0$.

2.5 FEM

The Finite Element Method (FEM) is a method for solving partial differential equations numerically. It is based on discretizing a complicated, continuous domain into a mesh consisting of smaller, simpler parts, usually triangles or quadrilaterals. In this thesis triangles are used.

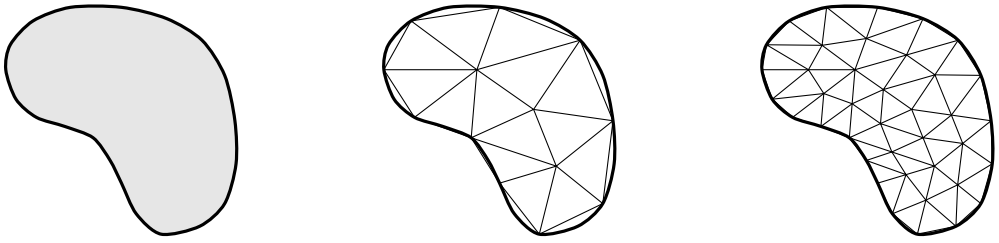
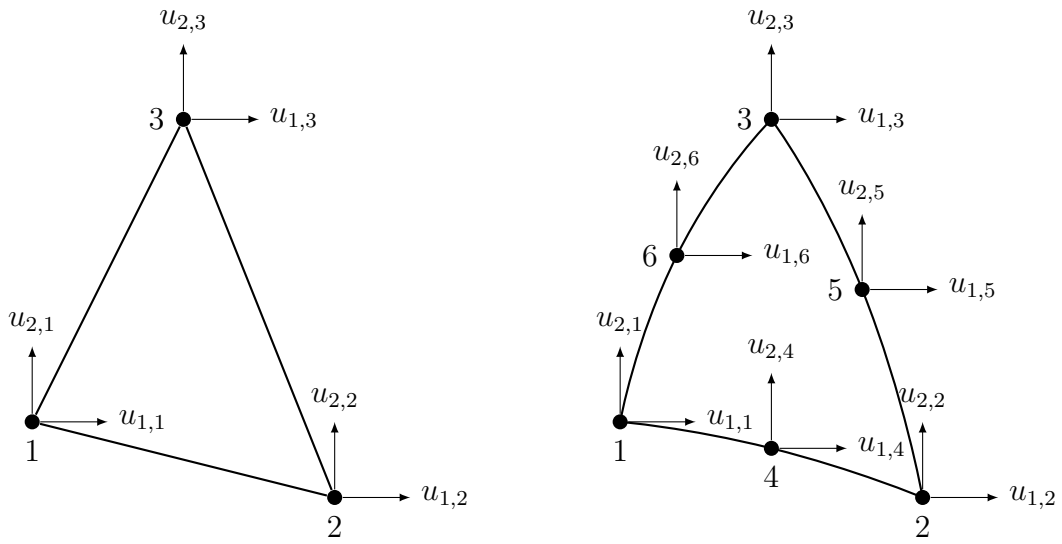


Figure 2.4: A domain is discretized into triangles to approximate the geometry. Smaller triangles means better capture of the geometry. A finer mesh will also give a solution with a smaller error.

The elements from the discretization of the domain is described by a set of basis functions. The simplest of which is linear basis functions. This results in a triangle with constant strain and is called Constant Strain Triangle (CST) [16]. As depicted in Figure 2.5a, the CST element only has nodes in the corners and straight edges. This results in six degrees of freedom. This type of triangle usually is worse at capturing the geometry and the solution will have a larger error.

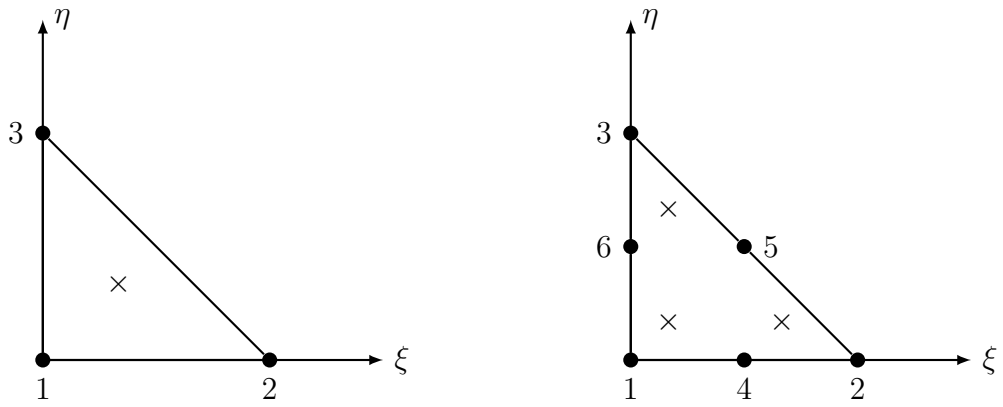
Second order basis functions will result in the strain being linear and is called a Linear Strain Triangle (LST). It is depicted in Figure 2.5b and has six nodes, three at the corners and three along the edges [14]. This results in a total of 12 degrees of freedom. This type of element is much better at capturing the geometry than the CST element, because of the second order basis functions can allow the edges to be curved. Because the strain in the LST element is linear it will also give a more accurate solution than the CST.



(a) CST element. **(b)** LST element.

Figure 2.5: Triangular element types. On the left triangle with linear basis functions and on right with quadratic basis functions.

The geometry of the elements shown in Figure 2.5 are not easy to integrate over. Therefore a coordinate transformation can be used to formulate easier geometries, which are called isoparametric elements. Triangular isoparametric elements are shown in Figure 2.6.



(a) CST isoparametric element. **(b)** LST isoparametric element.

Figure 2.6: Isoparametric elements with the quadrature points required for CST and LST.

To approximate an integral over an isoparametric element, numerical integration can be used through a quadrature formula on the form [15]

$$\int_{\Omega_\xi} f(\boldsymbol{\xi})\,\mathrm{d}\Omega_\xi \approx \sum_{k=1}^{n} \omega_k f(\boldsymbol{\xi}_k)$$

where $\boldsymbol{\xi}_k$ are quadrature points in the element, ω_k weights and n the number of quadrature points. Quadrature points are also known as Gauss points. If f is a polynomial, the approximation will be exact for a sufficiently high n. To integrate the CST element, one quadrature point is needed. To integrate the LST element, three quadrature points are needed instead. Coordinates and weights can be seen in Table 2.1 [17].

Table 2.1: Quadrature points and weights.

	CST	LST
$\boldsymbol{\xi}_i$	(1/3,1/3)	(1/6,1/6) (2/3,1/6) (1/6,2/3)
ω_i	1/2	1/6 1/6 1/6

2.5.1 FEM-form of equations

The residuals and Jacobians from the weak form in Section 2.3 will be written on FEM-form. The equation systems is written with matrices and the matrices will

be constructed by adding the contribution from every element. The contribution is written on tensor form, and because of that, the Julia-package `Tensors.jl` is used. The Julia-package Ferrite is used to supply the code with shape functions, shape gradient, etc. The system will be solved using a monolithic approach, meaning that the fields $\boldsymbol{u}$ and $\boldsymbol{\epsilon}^{\text{vp}}$ will be solved for directly. This is because snapshots of the viscoplastic strains $\boldsymbol{\epsilon}^{\text{vp}}$ are needed in the model reduction, see Section 3. Therefore, they need to be accessible at each time step. This is not possible with a conventional implementation, where only a solution for $\boldsymbol{u}$ is obtained and $\boldsymbol{\epsilon}^{\text{vp}}$ is stored as an internal variable.

2.5.2 Basis functions

The assumptions of small strain, plane strain and isotropy, see Section 2.4, will be reflected on the basis function. The fields $\boldsymbol{u}$ and $\boldsymbol{\epsilon}^{\text{vp}}$ will be approximated with basis functions. The viscoplastic strain is deviatoric and can be expressed as

$$\begin{aligned}
\boldsymbol{\epsilon}^{\text{vp}} &= \begin{bmatrix} \epsilon_{11}^{\text{vp}} & \epsilon_{12}^{\text{vp}} & 0 \\ \epsilon_{12}^{\text{vp}} & \epsilon_{22}^{\text{vp}} & 0 \\ 0 & 0 & -(\epsilon_{11}^{\text{vp}} + \epsilon_{22}^{\text{vp}}) \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \epsilon_{11}^{\text{vp}} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \epsilon_{22}^{\text{vp}} + \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \epsilon_{12}^{\text{vp}} \\
&= \boldsymbol{N}_1^{\text{vp}} \epsilon_{11}^{\text{vp}} + \boldsymbol{N}_2^{\text{vp}} \epsilon_{22}^{\text{vp}} + \boldsymbol{N}_3^{\text{vp}} \epsilon_{12}^{\text{vp}}.
\end{aligned}$$

The tensors $\boldsymbol{N}_1^{\text{vp}}$, $\boldsymbol{N}_2^{\text{vp}}$, and $\boldsymbol{N}_3^{\text{vp}}$ forms the basis for the viscoplastic strain in each Gauss point. This leads to the convention ϵ_{12} instead of $2\epsilon_{12}$, which is the usual convention when writing equations on Voigt form.

The viscoplatic strain in Gauss point q_a for an element $\boldsymbol{\epsilon}^{(\text{vp,e})}$ can be approximated as

$$\boldsymbol{\epsilon}_{q_a}^{(\text{vp,e})} \approx \sum_{j=1}^{3 \cdot n_{Gauss}} \boldsymbol{N}_j^{(\text{vp},e)} \left(\underline{a}^{(\text{vp},e)} \right)_j$$

where n_{Gauss} is the number of Gauss points for the element, $\boldsymbol{N}_j^{(\text{vp},e)}$ are shape functions and $\left(\underline{a}^{(\text{vp},e)} \right)_j$ is Gauss point values of the viscoplastic strain.

A CST element has one Gauss point, $n_{Gauss} = 1$, and therefore is $\underline{a}^{(\text{vp},e)}$

$$\underline{a}^{(\text{vp},e)} = \begin{bmatrix} \epsilon_{11,1}^{\text{vp}} \\ \epsilon_{22,1}^{\text{vp}} \\ \epsilon_{12,1}^{\text{vp}} \end{bmatrix}$$

and the shape functions $\boldsymbol{N}_j^{(\text{vp},e)}$ are

$$\boldsymbol{N}_1^{(\text{vp},e)} = \delta_{q_a,q_1} \boldsymbol{N}_1^{\text{vp}} \quad \boldsymbol{N}_2^{(\text{vp},e)} = \delta_{q_a,q_1} \boldsymbol{N}_2^{\text{vp}} \quad \boldsymbol{N}_3^{(\text{vp},e)} = \delta_{q_a,q_1} \boldsymbol{N}_3^{\text{vp}}$$

where δ_{q_a,q_1} is

$$\delta_{q_a,q_1} = \begin{cases} 1, & \text{if } q_a = q_1 \\ 0, & \text{otherwise.} \end{cases}$$

q_a is the current Gauss point and q_1 is the Gauss point. A LST element has three Gauss points, $n_{Gauss} = 3$, and will therefore have q_1, q_2, and q_3. The shape functions are then

$$\begin{aligned}
\boldsymbol{N}_1^{(\mathrm{vp},e)} &= \delta_{q_a,q_1}\boldsymbol{N}_1^{\mathrm{vp}} & \boldsymbol{N}_2^{(\mathrm{vp},e)} &= \delta_{q_a,q_1}\boldsymbol{N}_2^{\mathrm{vp}} & \boldsymbol{N}_3^{(\mathrm{vp},e)} &= \delta_{q_a,q_1}\boldsymbol{N}_3^{\mathrm{vp}} \\
\boldsymbol{N}_4^{(\mathrm{vp},e)} &= \delta_{q_a,q_2}\boldsymbol{N}_1^{\mathrm{vp}} & \boldsymbol{N}_5^{(\mathrm{vp},e)} &= \delta_{q_a,q_2}\boldsymbol{N}_2^{\mathrm{vp}} & \boldsymbol{N}_6^{(\mathrm{vp},e)} &= \delta_{q_a,q_2}\boldsymbol{N}_3^{\mathrm{vp}} \\
\boldsymbol{N}_7^{(\mathrm{vp},e)} &= \delta_{q_a,q_3}\boldsymbol{N}_1^{\mathrm{vp}} & \boldsymbol{N}_8^{(\mathrm{vp},e)} &= \delta_{q_a,q_3}\boldsymbol{N}_2^{\mathrm{vp}} & \boldsymbol{N}_9^{(\mathrm{vp},e)} &= \delta_{q_a,q_3}\boldsymbol{N}_3^{\mathrm{vp}}.
\end{aligned}$$

With the previously mention assumptions in this section, the displacement for an element $\boldsymbol{u}^{(e)}$ can be approximated as

$$\boldsymbol{u}^{(e)} \approx \sum_{i=1}^{n_{dofs}} \boldsymbol{N}_i^{(u,e)} \left(\underline{a}^{(u,e)}\right)_i$$

where n_{dofs} is the number of degrees of freedom for the displacement in an element, $\boldsymbol{N}_i^{(u,e)}$ are shape functions, and $\left(\underline{a}^{(u,e)}\right)_i$ are nodal values of the displacement.

For a CST element $n_{dofs} = 6$, and therefore is $\underline{a}^{(u,e)}$

$$\underline{a}^{(u,e)} = \begin{bmatrix} u_{1,1} \\ u_{2,1} \\ u_{1,2} \\ u_{2,2} \\ u_{1,3} \\ u_{2,3} \end{bmatrix}$$

and the shape functions $\boldsymbol{N}_i^{(u,e)}$ are

$$\begin{aligned}
\boldsymbol{N}_1^{(u,e)} &= N_1\boldsymbol{e}_1 & \boldsymbol{N}_2^{(u,e)} &= N_1\boldsymbol{e}_2 & \boldsymbol{N}_3^{(u,e)} &= N_2\boldsymbol{e}_1 \\
\boldsymbol{N}_4^{(u,e)} &= N_2\boldsymbol{e}_2 & \boldsymbol{N}_5^{(u,e)} &= N_3\boldsymbol{e}_1 & \boldsymbol{N}_6^{(u,e)} &= N_3\boldsymbol{e}_2
\end{aligned}$$

where isoparametric scalar shape functions are

$$N_1 = 1 - \xi - \eta \quad N_2 = \xi \quad N_3 = \eta.$$

The displacement gradient can be approximated as

$$\begin{aligned}
\boldsymbol{u}^{(e)} \otimes \boldsymbol{\nabla} &\approx \left(\sum_{i=1}^{n_{dofs}} \boldsymbol{N}_i^{(u,e)} \left(\underline{a}^{(u,e)}\right)_i\right) \otimes \boldsymbol{\nabla} \\
&\approx \sum_{i=1}^{n_{dofs}} \left(\underline{a}^{(u,e)}\right)_i \left(\boldsymbol{N}_i^{(u,e)} \otimes \boldsymbol{\nabla}\right) \\
&\approx \sum_{i=1}^{n_{dofs}} \left(\underline{a}^{(u,e)}\right)_i \left(\boldsymbol{N}_i^{(u,e)} \otimes \boldsymbol{\nabla}_\xi\right) \cdot \boldsymbol{J}_\xi^{-1}
\end{aligned}$$

where the Jacobian $\boldsymbol{J}_\xi$, is received from the transform to the isoparametric coordinate system and is

$$\boldsymbol{J}_\xi = \sum_{i=1}^{\frac{n_{dofs}}{2}} \mathbf{x}_i \otimes \boldsymbol{\nabla}_\xi \boldsymbol{N}_i^{(u,e)}$$

where $\mathbf{x}_i$ is the coordinates for the nodes in the element.

From that, $(\boldsymbol{\epsilon}[\boldsymbol{u}])^e$ can be calculated

$$\begin{aligned}
(\boldsymbol{\epsilon}[\boldsymbol{u}])^e &= \frac{1}{2}\Big((\boldsymbol{u}^{(e)} \otimes \boldsymbol{\nabla}) + (\boldsymbol{u}^{(e)} \otimes \boldsymbol{\nabla})^T\Big) \\
&\approx \frac{1}{2}\sum_{i=1}^{n_{dofs}} \Big(\underline{a}^{(u,e)}\Big)_i \Big(\boldsymbol{N}_i^{(u,e)} \otimes \boldsymbol{\nabla}_\xi\Big) \cdot \boldsymbol{J}_\xi^{-1} + \frac{1}{2}\sum_{i=1}^{n_{dofs}} \Big(\underline{a}^{(u,e)}\Big)_i \Big(\Big(\boldsymbol{N}_i^{(u,e)} \otimes \boldsymbol{\nabla}_\xi\Big) \cdot \boldsymbol{J}_\xi^{-1}\Big)^T \\
&\approx \sum_{i=1}^{n_{dofs}} \frac{1}{2}\Big[\Big(\Big(\boldsymbol{N}_i^{(u,e)} \otimes \boldsymbol{\nabla}_\xi\Big) \cdot \boldsymbol{J}_\xi^{-1}\Big) + \Big(\Big(\boldsymbol{N}_i^{(u,e)} \otimes \boldsymbol{\nabla}_\xi\Big) \cdot \boldsymbol{J}_\xi^{-1}\Big)^T\Big] \Big(\underline{a}^{(u,e)}\Big)_i \\
&\approx \sum_{i=1}^{n_{dofs}} \mathbf{B}_i \Big(\underline{a}^{(u,e)}\Big)_i
\end{aligned}$$

2.5.3 Residual formulation

The residuals in Section 2.3 is written on tensor form and will be expressed in FEM-form, which means using the approximations for $\boldsymbol{u}$ and $\boldsymbol{\epsilon}^{\text{vp}}$.

We recall the residual R^{u} in Equation 2.8,

$$\begin{aligned}
R^{\text{u}}(\boldsymbol{u}, \boldsymbol{\epsilon}^{\text{vp}}; \delta\boldsymbol{u}) &= a^u(\boldsymbol{u}; \delta\boldsymbol{u}) - c(\boldsymbol{\epsilon}^{\text{vp}}; \delta\boldsymbol{u}) - l(\delta\boldsymbol{u}) \\
&= \int_\Omega \boldsymbol{\epsilon}[\delta\boldsymbol{u}] : \mathbf{E} : \boldsymbol{\epsilon}[\boldsymbol{u}]\, \mathrm{d}\Omega - \int_\Omega \boldsymbol{\epsilon}[\delta\boldsymbol{u}] : \mathbf{E} : \boldsymbol{\epsilon}^{\text{vp}}\, \mathrm{d}\Omega - \int_\Omega \delta\boldsymbol{u} \cdot \boldsymbol{b}\, \mathrm{d}\Omega - \int_{\partial\Omega} \delta\boldsymbol{u} \cdot \boldsymbol{t}\, \mathrm{d}\Gamma
\end{aligned}$$

which on FEM-form can be written as

$$\underline{R}^{\text{u}} = \underline{K}\,\underline{a}^{(u)} - \underline{H}\,\underline{a}^{(\text{vp})} - \underline{f}.$$

The matrices in respective element formulation is

$$\begin{aligned}
(\underline{K})_{ij}^e &= \int_{\Omega_\xi^e} \boldsymbol{B}_i : \mathbf{E} : \boldsymbol{B}_j\, \mathrm{d}\Omega_\xi \\
(\underline{H})_{ij}^e &= \int_{\Omega_\xi^e} \boldsymbol{B}_i : \mathbf{E} : \boldsymbol{N}_j^{(\text{vp},e)}\, \mathrm{d}\Omega_\xi \\
\Big(\underline{f}\Big)_i^e &= \int_{\Omega_\xi^e} \boldsymbol{N}_i^{(u,e)} \cdot \boldsymbol{b}\, \mathrm{d}\Omega_\xi + \int_{\partial\Omega_\xi^e} \boldsymbol{N}_i^{(u,e)} \cdot \boldsymbol{t}\, \mathrm{d}\Gamma_\xi
\end{aligned}$$

$\underline{a}^{(u)}$ and $\underline{a}^{(\mathrm{vp})}$ are

$$\underline{a}^{(u)} = \begin{bmatrix} u_{1,1} \\ u_{2,1} \\ u_{1,2} \\ u_{2,2} \\ \vdots \\ u_{1,N_{dofs}} \\ u_{2,N_{dofs}} \end{bmatrix} \qquad \underline{a}^{(\mathrm{vp})} = \begin{bmatrix} \epsilon^{\mathrm{vp}}_{11,1} \\ \epsilon^{\mathrm{vp}}_{22,1} \\ \epsilon^{\mathrm{vp}}_{12,1} \\ \vdots \\ \epsilon^{\mathrm{vp}}_{11,N_{Gauss}} \\ \epsilon^{\mathrm{vp}}_{22,N_{Gauss}} \\ \epsilon^{\mathrm{vp}}_{12,N_{Gauss}} \end{bmatrix}$$

where N_{dofs} is the total degrees of freedom in the system and N_{Gauss} the total number of Gauss points in the system. It is important to note that the convention ϵ_{12} is used instead of $2\epsilon_{12}$. This is due to the use of the package `Tensors.jl` in Julia to evaluate the elementwise contributions.

The residual $R^{\boldsymbol{\epsilon}^{\mathrm{vp}}}$ in Equation 2.9 is

$$\begin{aligned} R^{\boldsymbol{\epsilon}^{\mathrm{vp}}}(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}}) &= a^{\mathrm{vp}}(\boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}}) - a^{\mathrm{vp}}\left({}^{(n)}\boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}}\right) - \Delta t\, b(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}}) \\ &= \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \boldsymbol{\epsilon}^{\mathrm{vp}}\, \mathrm{d}\Omega - \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : {}^{(n)}\boldsymbol{\epsilon}^{\mathrm{vp}}\, \mathrm{d}\Omega - \Delta t \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \boldsymbol{g}\, \mathrm{d}\Omega \end{aligned}$$

which on FEM-form can be written

$$\underline{R}^{\boldsymbol{\epsilon}^{\mathrm{vp}}} = \underline{M}^{(\mathrm{vp})}\, \underline{a}^{(\mathrm{vp})} - \underline{M}^{(\mathrm{vp})\ (\mathrm{n})}\underline{a}^{(\mathrm{vp})} - \Delta t\, \underline{P}.$$

The matrices in respective element formulation is

$$\begin{aligned} \left(\underline{M}^{(\mathrm{vp})}\right)^e_{ij} &= \int_{\Omega^e_\xi} \boldsymbol{N}^{(\mathrm{vp},e)}_i : \boldsymbol{N}^{(\mathrm{vp},e)}_j\, \mathrm{d}\Omega_\xi \\ (\underline{P})^e_i &= \int_{\Omega^e_\xi} \boldsymbol{N}^{(\mathrm{vp},e)}_i : \boldsymbol{g}\, \mathrm{d}\Omega_\xi\,. \end{aligned}$$

A solution satisfies the total residual if

$$\underline{R}^{FEM} = \begin{bmatrix} \underline{R}^{\mathrm{u}} \\ \underline{R}^{\boldsymbol{\epsilon}^{\mathrm{vp}}} \end{bmatrix} = \begin{bmatrix} \underline{0} \\ \underline{0} \end{bmatrix}.$$

2.5.4 Linearisation

The Jacobians in Section 2.3 is written on tensor form and will be expressed in FEM-form.

The Jacobian $R^{\mathrm{u}'}_u$ is

$$R^{\mathrm{u}'}_u(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{u}, \mathrm{d}\boldsymbol{u}) = \int_\Omega \boldsymbol{\epsilon}[\delta\boldsymbol{u}] : \mathbf{E} : \boldsymbol{\epsilon}[\mathrm{d}\boldsymbol{u}]\, \mathrm{d}\Omega$$

which on FEM-form is

$$\underline{R}_u^{\mathrm{u}'} = \underline{K}\,\mathrm{d}\underline{a}^{(u)}.$$

The Jacobian $R_{\boldsymbol{\epsilon}^{\mathrm{vp}}}^{\mathrm{u}'}$ is

$$R_{\boldsymbol{\epsilon}^{\mathrm{vp}}}^{\mathrm{u}'} = -\int_\Omega \boldsymbol{\epsilon}[\delta\boldsymbol{u}] : \mathbf{E} : \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}\,\mathrm{d}\Omega$$

which on FEM-form is

$$\underline{R}_{\boldsymbol{\epsilon}^{\mathrm{vp}}}^{\mathrm{u}'} = -\underline{H}\,\mathrm{d}\underline{a}^{(\mathrm{vp})}.$$

The Jacobian $R_u^{\boldsymbol{\epsilon}^{\mathrm{vp}\prime}}$ is

$$R_u^{\boldsymbol{\epsilon}^{\mathrm{vp}\prime}}(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}}, \mathrm{d}\boldsymbol{u}) = -\Delta t \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} : \mathbf{E} : \boldsymbol{\epsilon}[\mathrm{d}\boldsymbol{u}]\,\mathrm{d}\Omega$$

which can be written on FEM-form as

$$\underline{R}_u^{\boldsymbol{\epsilon}^{\mathrm{vp}\prime}} = -\Delta t\,\underline{S}\,\mathrm{d}\underline{a}^{(u)}.$$

The matrix $\underline{S}$ for an element is

$$(\underline{S})_{ij}^e = \int_{\Omega_\xi^e} \boldsymbol{N}_i^{(\mathrm{vp},e)} : \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} : \mathbf{E} : \boldsymbol{B}_j\,\mathrm{d}\Omega_\xi\,.$$

The Jacobian $R_{\boldsymbol{\epsilon}^{\mathrm{vp}}}^{\boldsymbol{\epsilon}^{\mathrm{vp}\prime}}$ is

$$R_{\boldsymbol{\epsilon}^{\mathrm{vp}}}^{\boldsymbol{\epsilon}^{\mathrm{vp}\prime}}(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}}, \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}) = \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}\,\mathrm{d}\Omega + \Delta t \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} : \mathbf{E} : \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}\,\mathrm{d}\Omega$$

which on FEM-form is

$$\underline{R}_{\boldsymbol{\epsilon}^{\mathrm{vp}}}^{\boldsymbol{\epsilon}^{\mathrm{vp}\prime}} = \underline{M}^{(\mathrm{vp})}\,\Delta\underline{a}^{(\mathrm{vp})} + \Delta t\,\underline{U}\,\mathrm{d}\underline{a}^{(\mathrm{vp})}.$$

The matrix $\underline{U}$ for an element is

$$(\underline{U})_{ij}^e = \int_{\Omega_\xi^e} \boldsymbol{N}_i^{(\mathrm{vp},e)} : \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} : \mathbf{E} : \boldsymbol{N}_j^{(\mathrm{vp},e)}\,\mathrm{d}\Omega_\xi\,.$$

2.5.5 Newton's method

Inserting the FEM-approximations into Newton's method on weak form in Equation 2.10 gives the equation system

$$\begin{bmatrix} \underline{K} & -\underline{H} \\ -\Delta t\underline{S} & \left[\underline{M}^{(\mathrm{vp})} + \Delta t\underline{U}\right] \end{bmatrix} \begin{bmatrix} \Delta\underline{a}^{(u)} \\ \Delta\underline{a}^{(\mathrm{vp})} \end{bmatrix} = -\begin{bmatrix} \underline{K}\,\underline{a}^{(u)} - \underline{H}\,\underline{a}^{(\mathrm{vp})} - \underline{f} \\ \underline{M}^{(\mathrm{vp})}\underline{a}^{(\mathrm{vp})} - \underline{M}^{(\mathrm{vp})\,(\mathrm{n})}\underline{a}^{(\mathrm{vp})} - \Delta t\underline{P} \end{bmatrix}$$

which can be solved for the vector $\begin{bmatrix} \Delta\underline{a}^{(u)} \\ \Delta\underline{a}^{(\mathrm{vp})} \end{bmatrix}$ and then each of the vectors $\Delta\underline{a}^{(u)}$ and $\Delta\underline{a}^{(\mathrm{vp})}$ can be extracted.

3

Numerical Model Reduction

A system with a large amount of degrees of freedom can be very computationally heavy and take a long time to solve. Finding techniques for reducing the amount of degrees of freedom can the have a huge advantage. This is called Numerical Model Reduction (NMR), but in the literature also known as Model Order Reduction (MOR) and Reduced Order Modelling (ROM). In this thesis, the notation NMR is used to emphasize that numerical methods are used to reduce the system. An overview of the methods used for the model reduction can be found [6, 3] and it is specifically based on the Nonuniform Transformation Field Analysis (NTFA) approach [8].

3.1 Reduced basis ansatz

The reduction technique used is based on separation of variables as

$$\boldsymbol{\epsilon}^{\mathrm{vp}}(\boldsymbol{x},t) \approx \sum_{a=1}^{N_R} \hat{\boldsymbol{\epsilon}}_a^{\mathrm{vp}}(\boldsymbol{x})\xi_a(t) \tag{3.1}$$

where the spatial and time domains are decomposed, $\hat{\boldsymbol{\epsilon}}_a^{\mathrm{vp}}(\boldsymbol{x})$ are the spatial mode functions and $\xi_a(t)$ are time dependent mode activity functions. N_R is the number of modes used and is usually much smaller than the total number of Gauss points $N_R \ll N_{Gauss}$.

The spatial modes are fixed functions in the spatial domain and the mode activity functions controls which spatial modes are active for each time and can thus vary over time. As a consequence of this, the variational form of $\boldsymbol{\epsilon}^{\mathrm{vp}}(\boldsymbol{x},t)$ is

$$\delta\boldsymbol{\epsilon}^{\mathrm{vp}}(\boldsymbol{x},t) \approx \sum_{a=1}^{N_R} \hat{\boldsymbol{\epsilon}}_a^{\mathrm{vp}}(\boldsymbol{x})\delta\xi_a(t). \tag{3.2}$$

The approximations in Equations 3.1 and 3.2 inserted into the weak forms in Equations 2.6 and 2.7 leads to

$$\int_\Omega \boldsymbol{\epsilon}[\delta\boldsymbol{u}] : \mathbf{E} : \boldsymbol{\epsilon}[\boldsymbol{u}]\,\mathrm{d}\Omega - \sum_{a=1}^{N_R}\int_\Omega \boldsymbol{\epsilon}[\delta\boldsymbol{u}] : \mathbf{E} : \hat{\boldsymbol{\epsilon}}_a^{\mathrm{vp}}\,\mathrm{d}\Omega\,\xi_a = \int_\Omega \delta\boldsymbol{u}\cdot\boldsymbol{b}\,\mathrm{d}\Omega + \int_{\partial\Omega}\delta\boldsymbol{u}\cdot\boldsymbol{t}\,\mathrm{d}\Gamma \tag{3.3}$$

$$\sum_{a=1}^{N_R}\delta\xi_a\left(\int_\Omega \hat{\boldsymbol{\epsilon}}_a^{\mathrm{vp}} : \left[\sum_{b=1}^{N_R}\hat{\boldsymbol{\epsilon}}_b^{\mathrm{vp}}\dot{\xi}_b\right]\mathrm{d}\Omega - \int_\Omega \hat{\boldsymbol{\epsilon}}_a^{\mathrm{vp}} : \boldsymbol{g}\left(\boldsymbol{\sigma}\left(\boldsymbol{\epsilon}[\boldsymbol{u}] - \sum_{b=1}^{N_R}\hat{\boldsymbol{\epsilon}}_b^{\mathrm{vp}}\xi_b\right)\right)\mathrm{d}\Omega\right) = 0. \tag{3.4}$$

In the NTFA approach, superposition of the displacements is

$$\boldsymbol{u}(\boldsymbol{x},t) \approx \boldsymbol{u}_0(\boldsymbol{x},t) + \sum_{a=1}^{N_R} \hat{\boldsymbol{u}}_a(\boldsymbol{x})\xi_a(t)$$

where $\boldsymbol{u}_0$ is the elastic displacement, and the summation of $\hat{\boldsymbol{u}}_a(\boldsymbol{x})\xi_a(t)$ is the contribution to the displacements due to plasticity. Note that the same mode activity functions $\xi_a(t)$ as for $\hat{\boldsymbol{\epsilon}}^{\text{vp}}$ are used. This is because $\boldsymbol{\epsilon}[\boldsymbol{u}]$ is linear, and therefore can $\xi_a(t)$ be factored out to describe both $\hat{\boldsymbol{u}}_a$ and $\hat{\boldsymbol{\epsilon}}_a^{\text{vp}}$ simultaneously. Inserting the superposition in Equation 3.3 and assuming

$$\begin{aligned}
\int_\Omega \boldsymbol{\epsilon}[\delta\boldsymbol{u}] : \mathbf{E} : \boldsymbol{\epsilon}[\boldsymbol{u}_0]\,\mathrm{d}\Omega &= \int_\Omega \delta\boldsymbol{u}\cdot\boldsymbol{b}\,\mathrm{d}\Omega + \int_{\partial\Omega} \delta\boldsymbol{u}\cdot\boldsymbol{t}\,\mathrm{d}\Gamma \\
\sum_{a=1}^{N_R}\left(\int_\Omega \boldsymbol{\epsilon}[\delta\boldsymbol{u}] : \mathbf{E} : \boldsymbol{\epsilon}[\hat{\boldsymbol{u}}_a]\,\mathrm{d}\Omega - \int_\Omega \boldsymbol{\epsilon}[\delta\boldsymbol{u}] : \mathbf{E} : \hat{\boldsymbol{\epsilon}}_a^{\text{vp}}\,\mathrm{d}\Omega\right)\xi_a &= 0 \\
\Rightarrow \int_\Omega \boldsymbol{\epsilon}[\delta\boldsymbol{u}] : \mathbf{E} : \boldsymbol{\epsilon}[\hat{\boldsymbol{u}}_a]\,\mathrm{d}\Omega &= \int_\Omega \boldsymbol{\epsilon}[\delta\boldsymbol{u}] : \mathbf{E} : \hat{\boldsymbol{\epsilon}}_a^{\text{vp}}\,\mathrm{d}\Omega\,, \quad a = 1,...,N_R
\end{aligned} \tag{3.5}$$

means that Equation 3.3 is satisfied. This means that the elastic displacements $\boldsymbol{u}_0$ can be solved as a linear elastic problem, and each $\hat{\boldsymbol{u}}_a$ is obtained exactly by solving a system using $\hat{\boldsymbol{\epsilon}}_a^{\text{vp}}$.

Inserting the superposition of $\boldsymbol{u}$ in Equation 3.4 yields a system of ODEs in ξ and time

$$\begin{aligned}
\sum_{b=1}^{N_R}\int_\Omega \hat{\boldsymbol{\epsilon}}_a^{\text{vp}} : \hat{\boldsymbol{\epsilon}}_b^{\text{vp}}\,\mathrm{d}\Omega\,\dot{\xi}_b - \int_\Omega \hat{\boldsymbol{\epsilon}}_a^{\text{vp}} : \boldsymbol{g}\left(\boldsymbol{\sigma}\left(\boldsymbol{\epsilon}\left[\boldsymbol{u}_0 + \sum_{b=1}^{N_R}\hat{\boldsymbol{u}}_b\xi_b\right] - \sum_{b=1}^{N_R}\hat{\boldsymbol{\epsilon}}_b^{\text{vp}}\xi_b\right)\right)\mathrm{d}\Omega = 0, \quad a = 1,...,N_R \\
\underline{M}\dot{\underline{\xi}} - \underline{f}(\underline{\xi}) = 0
\end{aligned}$$

where

$$\begin{aligned}
(\underline{M})_{ab} &= \int_\Omega \hat{\boldsymbol{\epsilon}}_a^{\text{vp}} : \hat{\boldsymbol{\epsilon}}_b^{\text{vp}}\,\mathrm{d}\Omega \\
\left(\underline{f}\right)_a &= \int_\Omega \hat{\boldsymbol{\epsilon}}_a^{\text{vp}} : \boldsymbol{g}\left(\boldsymbol{\sigma}\left(\boldsymbol{\epsilon}[\boldsymbol{u}_0] + \sum_{b=1}^{N_R}(\boldsymbol{\epsilon}[\hat{\boldsymbol{u}}_b] - \hat{\boldsymbol{\epsilon}}_b^{\text{vp}})\xi_b\right)\right)\mathrm{d}\Omega \\
\underline{\xi} &= \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_{N_R} \end{bmatrix}.
\end{aligned}$$

3.2 Newton's method

The time derivative $\dot{\xi}$ is discretized according to backward Euler method [15]

$$\dot{\xi} = \frac{\xi - {}^{(n)}\xi}{\Delta t}$$

where ${}^{(n)}\xi$ is the values of the mode activity functions from the previous time step and ξ is the values in the current time step, which is what is solved for. This leads to the residual

$$\underline{R}^{NMR} := \underline{M}\underline{\xi} - \underline{M}^{(n)}\underline{\xi} - \Delta t \underline{f}(\underline{\xi}).$$

The variational formulation of $\underline{f}$ is needed for calculating the Jacobian. It is

$$\begin{aligned} d\underline{f} &= \int_\Omega \hat{\boldsymbol{\epsilon}}_a^{\text{vp}} : \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} : \mathbf{E} : \sum_{b=1}^{N_R} (\boldsymbol{\epsilon}[\hat{\boldsymbol{u}}_b] - \hat{\boldsymbol{\epsilon}}_b^{\text{vp}}) \Delta\xi_b \, \mathrm{d}\Omega \\ &= \underline{F}\Delta\underline{\xi} \end{aligned}$$

where

$$(\underline{F})_{ab} = \int_\Omega \hat{\boldsymbol{\epsilon}}_a^{\text{vp}} : \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} : \mathbf{E} : (\boldsymbol{\epsilon}[\hat{\boldsymbol{u}}_b] - \hat{\boldsymbol{\epsilon}}_b^{\text{vp}}) \, \mathrm{d}\Omega$$

note that here $\hat{\boldsymbol{\epsilon}}^{\text{vp}}$ is a second order tensor, which need to be taken into account when extracting the snapshots and constructing the mode functions. This means that the factor 2 in $2\epsilon_{12}$ in Voigt notation is not used in this formulation.
The Jacobian is

$$\underline{J} = \underline{M} - \Delta t \underline{F}$$

and the Newton iterations

$$\begin{aligned} \underline{J}\Delta\underline{\xi} &= -\underline{R}^{NMR} \\ \underline{\xi}^{(k+1)} &= \underline{\xi}^{(k)} + \Delta\underline{\xi} \end{aligned}$$

becomes

$$\begin{aligned} \left[\underline{M} - \Delta t \underline{F}\right]\Delta\underline{\xi} &= -\left[\underline{M}\underline{\xi} - \underline{M}^{(n)}\underline{\xi} - \Delta t \underline{f}(\underline{\xi})\right] \\ \underline{\xi}^{(k+1)} &= \underline{\xi}^{(k)} + \Delta\underline{\xi}. \end{aligned}$$

3.3 Proper Orthogonal Decomposition

An orthogonal basis can be constructed by the method Proper Orthogonal Decomposition (POD). An overview of the method used can be found in [6, 3]. From the FEM-solution, $\boldsymbol{\epsilon}^{\text{vp}}(\boldsymbol{x}, t)$ is obtained. From it, N_S snapshots can be extracted and stored in $\tilde{\underline{\boldsymbol{\epsilon}}}^{\text{vp}}$, where each column represents values of the viscoplastic strains at a specific time step. POD then takes the set of N_S column vectors $\tilde{\underline{\boldsymbol{\epsilon}}}_s^{\text{vp}}$, where $s = 1, ..., N_S$, and extracts a new set of orthogonal basis vectors. The correlation matrix is constructed as

$$g_{st} = [\tilde{\underline{\boldsymbol{\epsilon}}}_s^{\text{vp}}]^T \tilde{\underline{\boldsymbol{\epsilon}}}_t^{\text{vp}}.$$

For g, the eigenvalue problem $(g_{st} - \lambda\delta_{st})v_t = 0$ is solved for eigenvalues λ and eigenvectors $\underline{v}$. The eigenvalues are sorted in decreasing order and values $|\lambda_s| < \beta|\lambda_1|$

are removed. This results in a reduced basis with N_R eigenvalues.

The orthogonal basis $\hat{\underline{\boldsymbol{\epsilon}}}^{\mathrm{vp}}$ is constructed by

$$\hat{\underline{\boldsymbol{\epsilon}}}_a^{\mathrm{vp}} = \sum_{s=1}^{N_S} (\underline{v}_a)_s \tilde{\underline{\boldsymbol{\epsilon}}}_s^{\mathrm{vp}}, \quad a = 1, 2, ..., N_R \tag{3.6}$$

where $\underline{v}_a$ is the eigenvector associated with eigenvalue λ_a. Because $\hat{\underline{\boldsymbol{\epsilon}}}^{\mathrm{vp}}$ is an orthogonal basis

$$[\hat{\underline{\boldsymbol{\epsilon}}}_a^{\mathrm{vp}}]^T \hat{\underline{\boldsymbol{\epsilon}}}_b^{\mathrm{vp}} = \begin{cases} \lambda_a, & \text{if } a = b \\ 0, & \text{else} \end{cases}$$

where it is important to note that the basis is orthogonal, but not orthonormal.

4

Implementation

Simulations were done on one or a couple training cases. From these, snapshots were extracted to then perform POD and NMR. The same snapshots were then used to simulate some test cases and compare the FEM-solution to the NMR-solution.

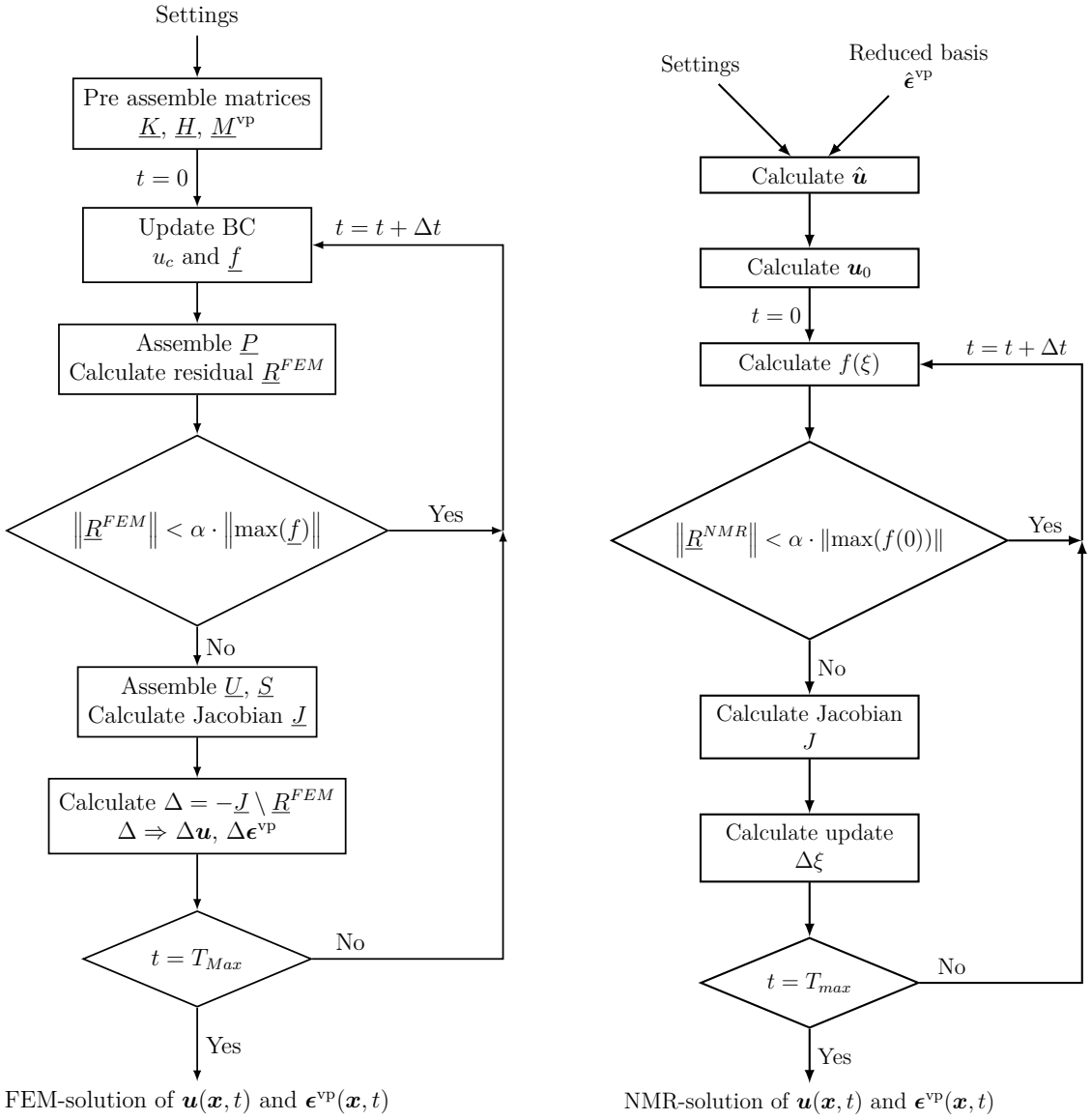
4.1 Program structure

The structure for the solving procedure for FEM and NMR can be seen in Figure 4.1. Both have a similar procedure and use the same settings. The settings include choice of geometry, applied boundary conditions, loads and the material parameters. One additional setting for NMR is the number of modes to solve with.

The procedure for FEM can be seen in Figure 4.1a. Some of the matrices don't depend on the solution and can therefore be assembled before the time stepping. These are $\underline{K}$, $\underline{H}$ and $\underline{M}^{\text{vp}}$. In the beginning of each time step, the current BC is applied and the constrained displacements $\boldsymbol{u}_c$ and load vector $\underline{f}$ created. The vector $\underline{P}$ is assembled and the residual $\underline{R}^{FEM}$ calculated. If the norm of the residual show convergence, then the next time step is started. If it has not converged, the matrices $\underline{U}$ and $\underline{S}$ are assembled. They are then used to calculate the Jacobian J, which is used to solve for the update Δ, which in turn leads to the updates $\Delta\boldsymbol{u}$ and $\Delta\boldsymbol{\epsilon}^{\text{vp}}$. If it is the last time step, a solution for $\boldsymbol{u}(\boldsymbol{x}, t)$ and $\boldsymbol{\epsilon}^{\text{vp}}(\boldsymbol{x}, t)$ has been reached, otherwise it continues with the next time step.

The procedure for NMR can be seen in Figure 4.1b and is similar to the procedure for FEM. It requires both the same settings as for FEM, but also the number of modes N_R and the reduced basis $\hat{\boldsymbol{\epsilon}}^{\text{vp}}$. The procedure is split into two parts, one "offline stage" and one "online stage". The offline stage includes things that can be performed one time and then reused multiple times. For the current problem, this is using POD on extracted snapshots to calculate $\hat{\boldsymbol{\epsilon}}^{\text{vp}}$, and solving for the modes $\hat{\boldsymbol{u}}$ using Equation 3.5. The online stage is the actual problem solving stage, and it makes use of the precomputed mode functions calculated during the offline stage. Because the actual problem can vary, calculating $\boldsymbol{u}_0$ and solving for $\xi(t)$ are part of the online stage, although $\boldsymbol{u}_0$ could be part of the offline stage for parameterized loads $\boldsymbol{b}$ and $\boldsymbol{t}_{\text{P}}$. The matrix M doesn't depend on ξ and can therefore be assembled before the time stepping. In the beginning of each time step, $f(\xi)$ and the residual $\underline{R}^{NMR}$ are calculated. If convergence, the next time step is started. If not, the Jacobian is calculated to then get the updates $\Delta\xi$. If it is the last time step, a

NMR-solution for $\boldsymbol{u}(\boldsymbol{x},t)$ and $\boldsymbol{\epsilon}^{\text{vp}}(\boldsymbol{x},t)$ has been reached. Otherwise it continues with the next time step.



(a) Solving procedure for FEM. **(b)** Solving procedure for NMR.

Figure 4.1: The solving procedures for FEM and NMR. Both have a similar structure.

To obtain the basis functions $\hat{\boldsymbol{\epsilon}}^{\text{vp}}$ needed for the NMR procedure, POD described in Figure 4.2 was used. For snapshots $\tilde{\underline{\boldsymbol{\epsilon}}}^{\text{vp}}$, obtained from the FEM-solution, eigenvalues λ_n can be calculated. Only eigenvalues with a relative size larger than the specified cut-off β are chosen, and subsequently only the corresponding eigenvectors were used. To construct the reduced basis $\hat{\boldsymbol{\epsilon}}^{\text{vp}}$, Equation 3.6 was used.

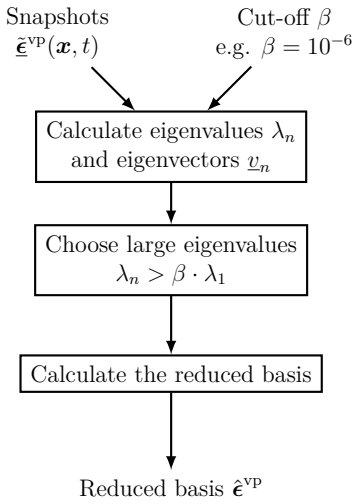


Figure 4.2: General structure of the POD computation.

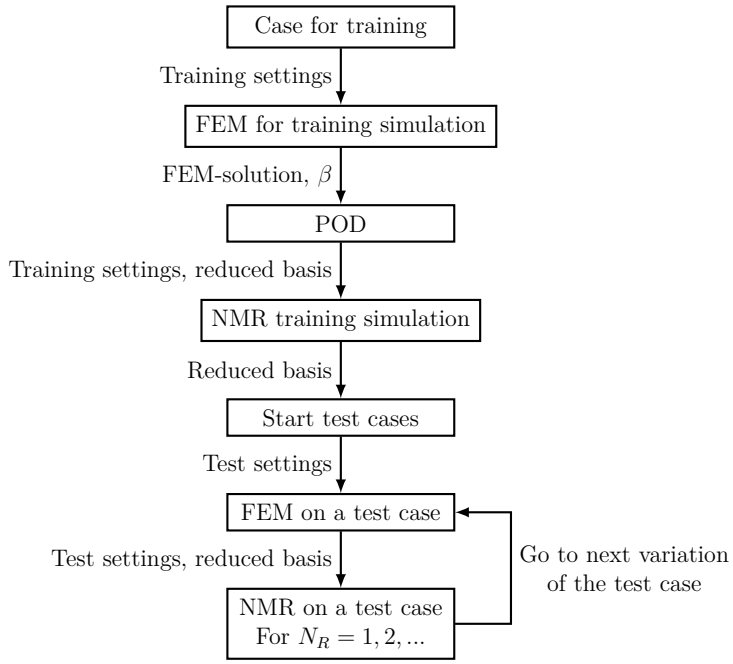


Figure 4.3: General structure of the testing. For a specified training simulation a FEM-solution is obtained. POD is performed and reduced basis extracted for the NMR calculation. Then, for a set of test cases, FEM and NMR is repeated to test how good the reduction is.

The total solving procedure can be seen in Figure 4.3. It starts with defining one or a few training cases, which was solved using FEM. Snapshots from each solution was saved for later use. Then, for each case respectively, POD and NMR were calculated. This step is not necessary for the later steps, but can be important to see how well NMR performs when directly decomposed.

One or more test cases are then solved with FEM, to obtain a comparative solution

in each case. These tests are generally more complicated than the training cases. For each test case, POD is performed by combining the snapshots computed previously during the training, see Section 4.5 for more details. From POD a reduced basis is obtained, for which NMR is solved. The FEM and NMR-solutions can then be compared and evaluated, partly based on the methods described in Section 4.2.

4.2 Comparing results

To compare and evaluate the performance of NMR, some standard methods were needed. One was the time to obtain a FEM-solution compared to the time for the online stage in NMR.

The tip displacements over time were used to compare FEM and NMR. This was done by integrating the displacements on the right edge. Also the relative displacement error between FEM and NMR-solutions for the tip displacements were used. These were calculated with

$$\begin{aligned} u_{tip,1}^{rel}(t) &= \frac{u_{tip,1}^{NMR}(t) - u_{tip,1}^{FEM}(t)}{\max\limits_{t}\left(\left|u_{tip,1}^{FEM}(t)\right|\right)} \\ u_{tip,2}^{rel}(t) &= \frac{u_{tip,2}^{NMR}(t) - u_{tip,2}^{FEM}(t)}{\max\limits_{t}\left(\left|u_{tip,2}^{FEM}(t)\right|\right)} \end{aligned} \tag{4.1}$$

where the error could be both positive or negative and change over time.

For a loading ending with zero traction, some residual stresses would be left within the material if it plasticised. To compare FEM and NMR a root mean square (RMS) error was used

$$\sigma_e^{RMS} = \sqrt{\frac{1}{\Omega}\int_\Omega \left(\sigma_e^{NMR} - \sigma_e^{FEM}\right)^2 \mathrm{d}\Omega} \tag{4.2}$$

this means that the error σ_e^{RMS} will always be positive.

4.3 Newton iterations

When solving the systems of equations for both FEM and NMR, the norm of the residual was used to determine convergence. But due to FEM and NMR having very different number of unknowns, only comparing the norm of the residual vector with a fixed value is not appropriate. Instead a relative method was used. For FEM, the convergence criteria was comparing the norm of the residual to the norm of the load vector at the time step with the largest traction

$$\|\underline{R}^{FEM}\| < \alpha \|f_{max}\|$$

where the factor α was a small number, in this thesis it was set to $\alpha = 10^{-6}$.

For NMR a similar approach was used. Because the equivalent load vector $f(\xi)$

depends on ξ, it is not possible beforehand to know what the maximum value of $f(\xi)$ is. Therefore the assumption $\xi = 0$ together with the $\boldsymbol{u}_0$ with largest summation of $|\boldsymbol{u}_0|$ nodal values were used to construct $f(0)_{max}$. The convergence criteria was then

$$\|\underline{R}^{NMR}\| < \alpha \|f(0)_{max}\|$$

were the same α as for FEM was used.

One way of reducing the number of Newton iterations is to try to guess the solution of $\boldsymbol{u}$ and $\boldsymbol{\epsilon}^{\text{vp}}$ in the next time step. This might create a better starting point, and thus reducing the number of Newton iterations. The easiest method is to just use the solution in current time step as the starting point. Another method is to use the current solution and the previous solution as

$$\begin{aligned} {}^{(n+2)}\boldsymbol{u} &= {}^{(n+1)}\boldsymbol{u} + \left({}^{(n+1)}\boldsymbol{u} - {}^{(n)}\boldsymbol{u}\right) \\ {}^{(n+2)}\boldsymbol{\epsilon}^{\text{vp}} &= {}^{(n+1)}\boldsymbol{\epsilon}^{\text{vp}} + \left({}^{(n+1)}\boldsymbol{\epsilon}^{\text{vp}} - {}^{(n)}\boldsymbol{\epsilon}^{\text{vp}}\right) \end{aligned}$$

where ${}^{(n+1)}\boldsymbol{u}$ and ${}^{(n+1)}\boldsymbol{\epsilon}^{\text{vp}}$ are the current solutions, ${}^{(n)}\boldsymbol{u}$ and ${}^{(n)}\boldsymbol{\epsilon}^{\text{vp}}$ the solutions from the previous time step and ${}^{(n+2)}\boldsymbol{u}$ and ${}^{(n+2)}\boldsymbol{\epsilon}^{\text{vp}}$ are the guessed starting points for the next time step. It's important to note that this is something performed after Newton's method arrived at a solution in the current time step, and not to be confused with the iterative solutions during the Newton iterations, $\boldsymbol{u}^{(k)}$ and $\boldsymbol{\epsilon}^{\text{vp}(k)}$. This method was not used to solve for $\boldsymbol{u}$ and $\boldsymbol{\epsilon}^{\text{vp}}$, because of numerical solving issues. For them, just the current solutions were used as the starting points. In NMR, the guessing method used was

$${}^{(n+2)}\xi = {}^{(n+1)}\xi + \left({}^{(n+1)}\xi - {}^{(n)}\xi\right)$$

for which the same numerical difficulties were not observed. This difference in method might have introduced a slight bias in solve time benefiting NMR over FEM, due to reduced number of Newton iterations. Thus the time difference might appear larger than with the same guessing method.

4.4 Implementation in code

The programming language used for the numerical implementations was Julia [18], in which both FEM and NMR were coded. This allowed for efficient calculations and using some Julia packages made the implementation easier. One of the main packages used was `Ferrite.jl` [19], a finite element toolbox. This package handled some of the important aspects of the program, such as interpolation between nodes, quadrature points, evaluate shape functions and its derivatives, and generating the simple meshes used. This made it possible to implement both CST and LST elements, although only CST was used in this thesis. This package was used for both of the FEM and NMR implementations.

Another important package used was `Tensors.jl` [20]. It allowed for performing actual tensor operations in Julia and not having to rewrite the equations on Voigt

form using standard matrix multiplications. This meant less chance of error in derivations, implementations, and more understandable code. This because equations could be written on a form close to their mathematical formulations.

4.5 Extracting snapshots

Performing a FEM-solution results in two fields, $\boldsymbol{u}$ and $\boldsymbol{\epsilon}^{\text{vp}}$. These can be obtained and stored for every time step, thus creating snapshots over time. To construct the reduced basis $\hat{\boldsymbol{\epsilon}}^{\text{vp}}$, the snapshots of $\boldsymbol{\epsilon}^{\text{vp}}$ were needed. One efficient, but complicated, method could be to ignore snapshots from before the material plasticised, when $\boldsymbol{\epsilon}^{\text{vp}}$ would be zeros, and only using snapshots at specified time steps. This would decrease the matrix used for POD, and thus decrease the computational cost. In the current work, all of the snapshots were used. This reduced the implementation complexity and therefore less chance of error.

If two or more training simulations were used, a method of combining them was needed. If snapshots are stored in matrices S_1, S_2, ... S_n for n training simulations, where each column corresponds to $\boldsymbol{\epsilon}^{\text{vp}}$ at a specific time step, then the combined snapshot matrix was

$$S = [S_1 \; S_2 \; \ldots \; S_n]$$

on which POD then was performed.

5

Numerical investigations

This chapter presents the numerical investigations made to test the performance of NMR described in Chapter 3. First, an illustrative example of the method applied to a simple cantilever beam is presented. Then a case with training simulations and test simulations to test how well the training simulations could predict the behaviour in a new situation.

5.1 Cantilever beam

A simple cantilever beam with vertical traction applied on the right edge is depicted in Figure 5.1. This was used to illustrate the decomposition into modes and the solution's dependency on the number of modes. The maximum traction $\boldsymbol{t}_2 = -30\boldsymbol{e}_2$ MPa was uniformly applied and increased from zero to the maximum following a sinusoidal function.

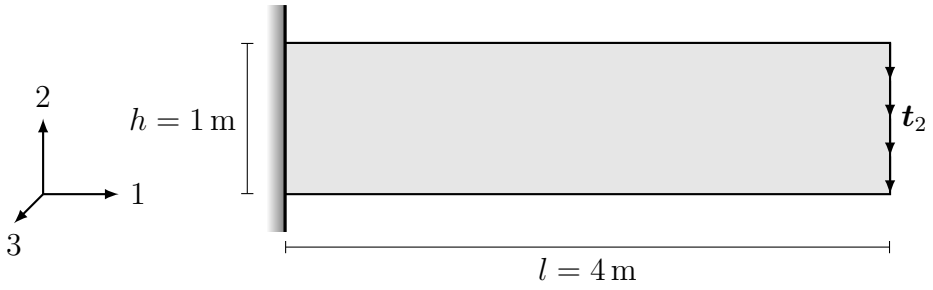


Figure 5.1: A simple cantilever beam.

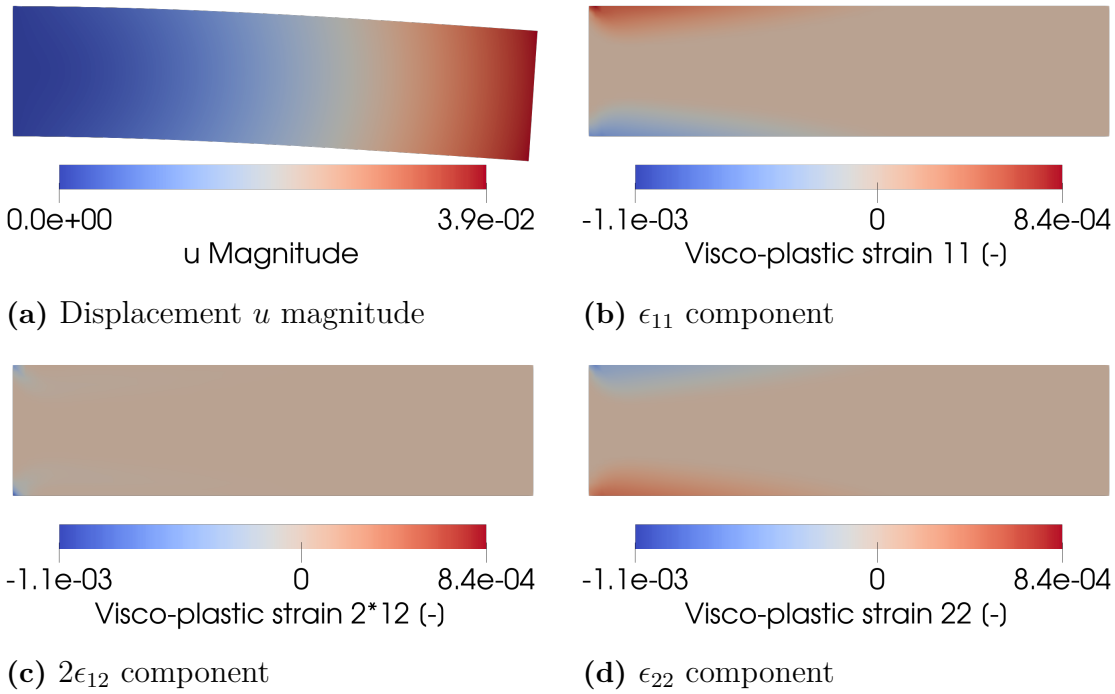
The material properties are shown in Table 5.1. The properties for G, K, and σ_y was based on structural steel [21, 22, 23]. The properties t^*, n_c and σ_c was chosen arbitrarily to introduce viscoplasticity to the material. It does therefore not necessarily reflect the true plasticity behaviour of structural steel.

Table 5.1: Material properties for the cantilever beam.

G	K	σ_y	t^*	n_c	σ_c
160 GPa	79.3 GPa	250 MPa	10 s	1	3G

5.1.1 FEM

The FEM-solution of the problem can be seen in Figure 5.2. The maximum displacement magnitude is 3.9 cm at the right edge. The viscoplastic strains are concentrated in the upper and lower left corners, because that's where the stress concentrations are for a cantilever beam. Thus more plasticity occur in those regions.



(a) Displacement u magnitude

(b) ϵ_{11} component

(c) $2\epsilon_{12}$ component

(d) ϵ_{22} component

Figure 5.2: FEM-solution at $t = T_{max}$.

5.1.2 POD

Figure 5.3 shows the first 15 normalized eigenvalues calculated during POD. They are plotted relative to the first, and the cut-off of $\beta = 10^{-6}$ resulted in six extracted modes.

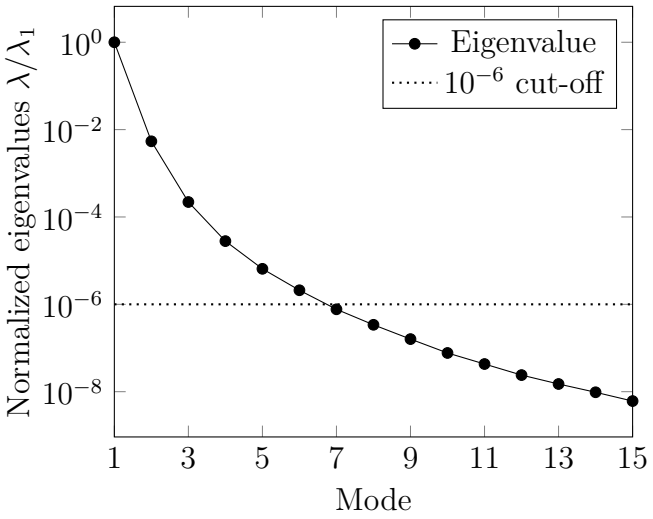
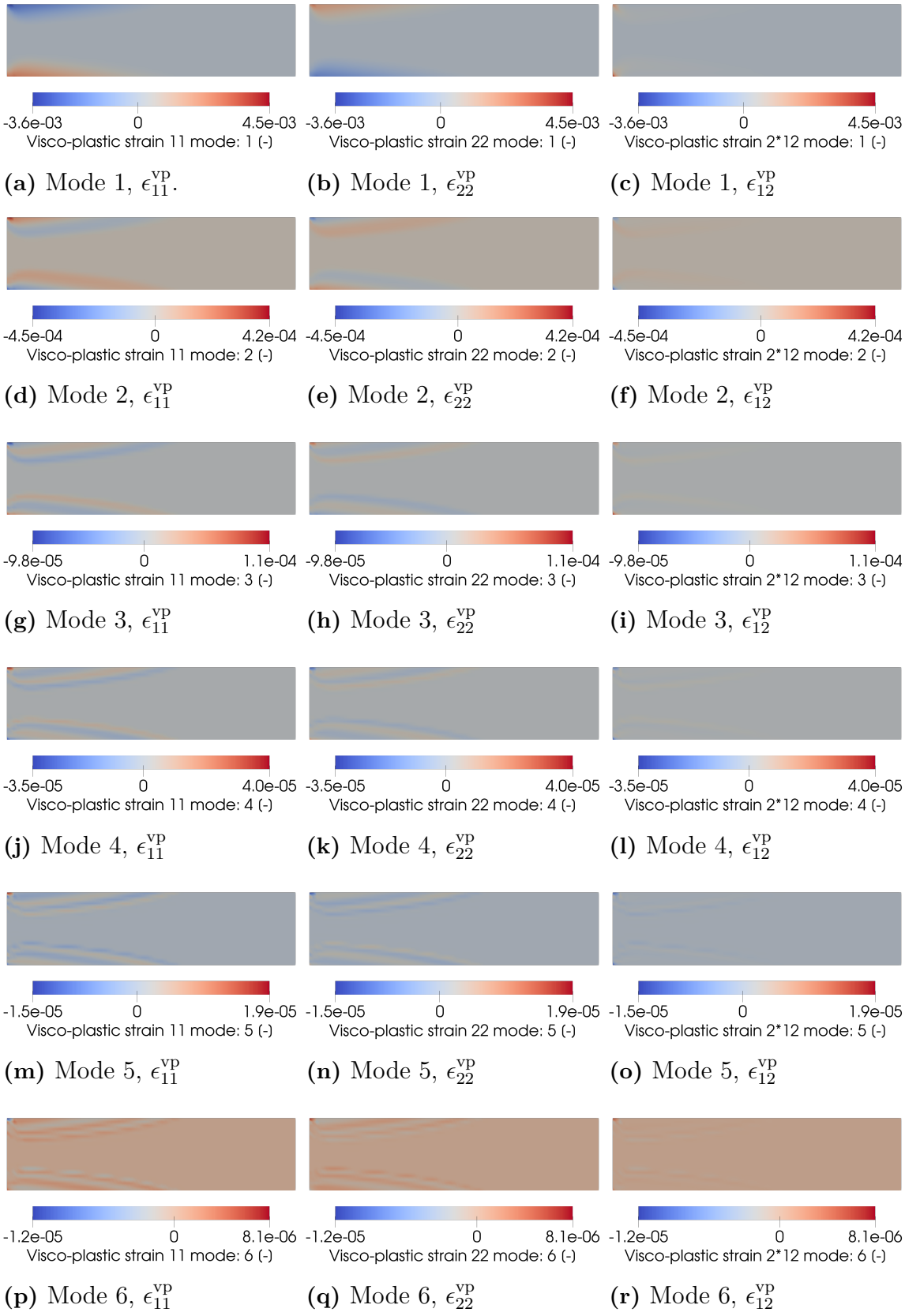


Figure 5.3: Relative eigenvalues from snapshots in POD method.

The extracted modes can be seen in Figure 5.4 for each component $\epsilon_{11}^{\text{vp}}$, $\epsilon_{22}^{\text{vp}}$ and $\epsilon_{12}^{\text{vp}}$. The components for the first mode are similar to respective component in the finite element solution in Figure 5.2. They show about the same behaviour in the distribution of the viscoplastic strains, but not the same magnitude. This is because each mode $\hat{\boldsymbol{\epsilon}}_a^{\text{vp}}$ is scaled with a mode activity function $\xi_a(t)$. It also indicates, as predicted by the eigenvalues, that the most important mode to capture the physical behaviour is the first one.

It is also clear from Figure 5.4 that the range of values for each mode approaches zero for higher modes. This is expected due to a mode scalar multiplied with itself results in it's corresponding eigenvalue, which is smaller for higher modes. Thus is it expected that most values in the higher modes are smaller than in the lower modes.

The modes also show an increase in the frequency of the features for higher modes. The first modes have one feature in each corner, the second mode two features and this pattern continues for the higher modes.



(a) Mode 1, $\epsilon_{11}^{\text{vp}}$. **(b)** Mode 1, $\epsilon_{22}^{\text{vp}}$ **(c)** Mode 1, $\epsilon_{12}^{\text{vp}}$

(d) Mode 2, $\epsilon_{11}^{\text{vp}}$ **(e)** Mode 2, $\epsilon_{22}^{\text{vp}}$ **(f)** Mode 2, $\epsilon_{12}^{\text{vp}}$

(g) Mode 3, $\epsilon_{11}^{\text{vp}}$ **(h)** Mode 3, $\epsilon_{22}^{\text{vp}}$ **(i)** Mode 3, $\epsilon_{12}^{\text{vp}}$

(j) Mode 4, $\epsilon_{11}^{\text{vp}}$ **(k)** Mode 4, $\epsilon_{22}^{\text{vp}}$ **(l)** Mode 4, $\epsilon_{12}^{\text{vp}}$

(m) Mode 5, $\epsilon_{11}^{\text{vp}}$ **(n)** Mode 5, $\epsilon_{22}^{\text{vp}}$ **(o)** Mode 5, $\epsilon_{12}^{\text{vp}}$

(p) Mode 6, $\epsilon_{11}^{\text{vp}}$ **(q)** Mode 6, $\epsilon_{22}^{\text{vp}}$ **(r)** Mode 6, $\epsilon_{12}^{\text{vp}}$

Figure 5.4: Modes for the different components.

5.1.3 NMR

The convergence of average tip displacement for NMR can be seen in Figure 5.5. It converges fast for increasing number of modes, which means that NMR gives an accurate description of the FEM-solution.

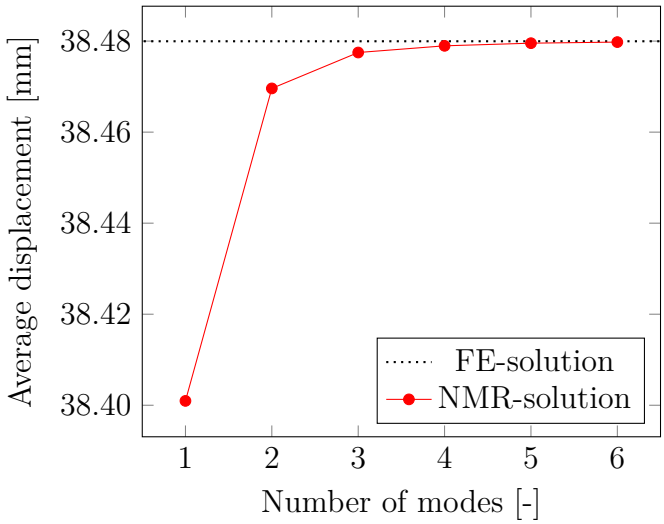
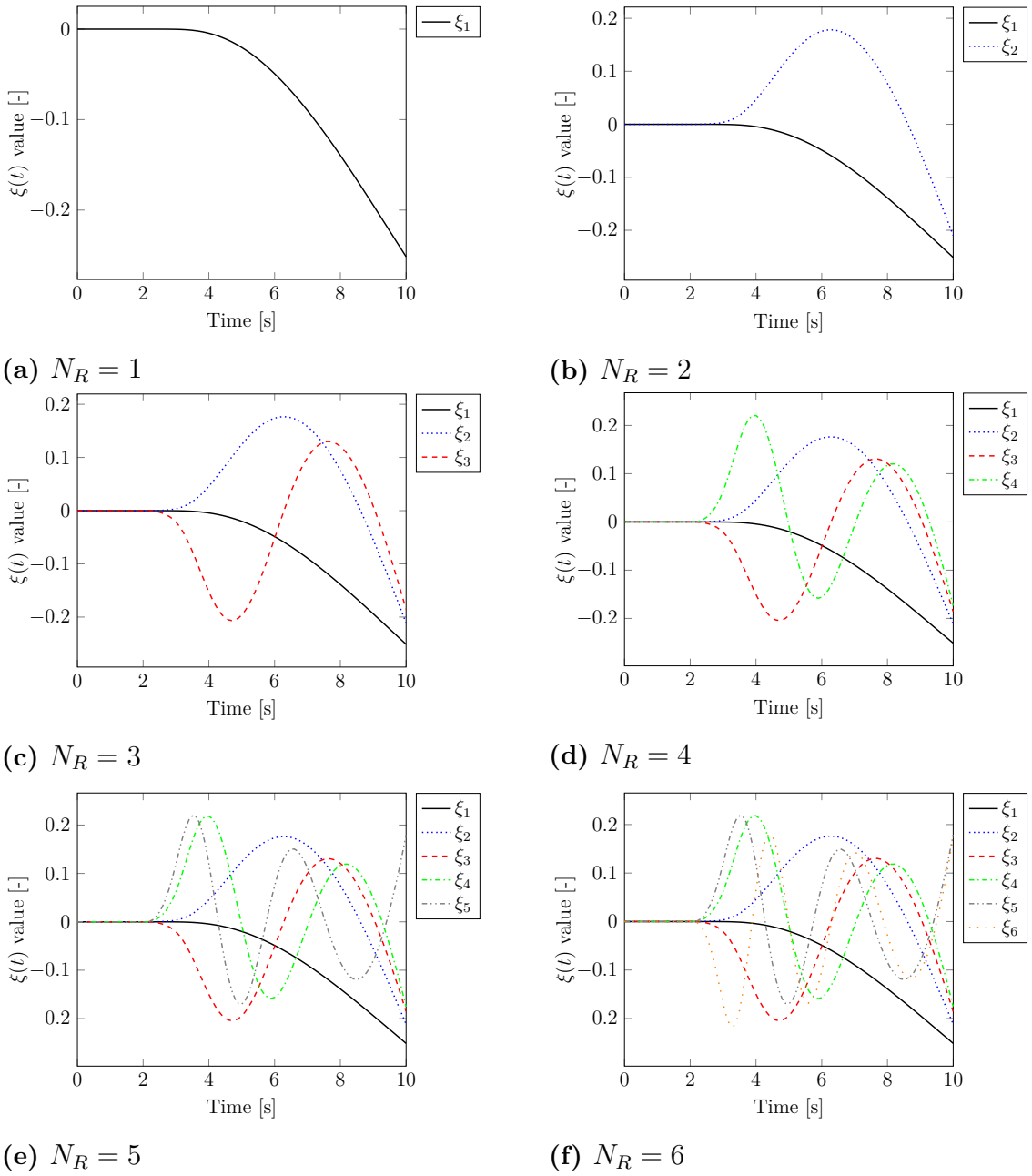


Figure 5.5: Convergence of average tip displacement for increasing number of modes.

Figure 5.6 shows the mode activity functions $\xi(t)$ for solving the cantilever beam problem for different number of modes. Modes 1 in Figure 5.4 are multiplied with $\xi_1(t)$, modes 2 with $\xi_2(t)$, and similarly for higher modes. Mode 1 is the most important and therefore follows $\xi_1(t)$ what would be expected for the plasticity, zeros in the beginning and then growing in magnitude monotonically. The rest of the modes show an oscillation over time. This is because they only add a slight change on top of the main behaviour captured by mode 1. To prohibit the summation of modes and mode activity functions to grow in magnitude for increased number of modes, the average of $\xi_n(t)$ for $n \geq 2$ must be low. These modes oscillate, ensuring that the average is low.

During approximately the first 2 s in all subfigures in Figure 5.6, all $\xi_n(t) = 0$. This is because in the beginning of the loading, the strain in the beam is in the elastic region, thus is $\xi_n(t) = 0$.



(a) $N_R = 1$ **(b)** $N_R = 2$

(c) $N_R = 3$ **(d)** $N_R = 4$

(e) $N_R = 5$ **(f)** $N_R = 6$

Figure 5.6: The mode activity functions $\xi(t)$ for different number of modes. The flat regions in the first 2 s are when the beam is in elastic bending and therefore no plasticity, thus must all $\xi_n = 0$.

5.2 Bending over edge

One test case that also was investigated is shown in Figure 5.7. A slender beam is on the left side constrained to only moving in the vertical direction. On the bottom left half is constrained to only moving in the horizontal direction. Traction was applied on the right tip. This was meant to simulate how a beam is bending over an edge.

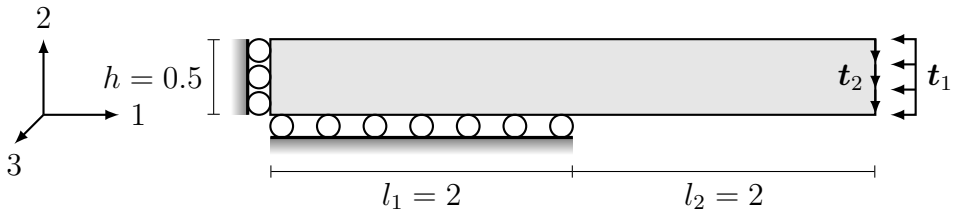


Figure 5.7: Beam hanging over an edge where the loading is applied on the outer edge.

The material properties were the same as in Section 5.1 and are shown in Table 5.2. The properties for G, K, and σ_y was based on structural steel [21, 22, 23]. The properties t^*, n_c and σ_c was chosen arbitrarily to introduce viscoplasticity to the material. It does therefore not necessarily reflect the true plasticity behaviour of structural steel.

Table 5.2: Material properties for the case bending over edge

G	K	σ_y	t^*	n_c	σ_c
160 GPa	79.3 GPa	250 MPa	10 s	1	3G

On the specified case, first two training simulations were performed to obtain snapshots. Then, for the test simulations, a new loading situation was applied. The snapshots from the the training were used to construct the reduced basis $\hat{\boldsymbol{\epsilon}}^{\text{vp}}$. This to test how well NMR performed based on simpler simulations in a new situation.

Training simulations:

1. Simulation for linear ramp to $\boldsymbol{t} = \boldsymbol{t}_2 = -40\boldsymbol{e}_2$ MPa in 2.5 s with $\Delta t = 0.05$ s
   - $\boldsymbol{t} = \boldsymbol{t}_2 \frac{t}{T_{max}}$, $t \in [0, T_{max}]$, $T_{max} = 2.5$ s
   - FEM-solution
   - POD
   - NMR to see convergence
2. Simulation for linear ramp to $\boldsymbol{t} = \boldsymbol{t}_1 = -1000\boldsymbol{e}_1$ MPa in 2.5 s with $\Delta t = 0.05$ s
   - $\boldsymbol{t} = \boldsymbol{t}_1 \frac{t}{T_{max}}$, $t \in [0, T_{max}]$, $T_{max} = 2.5$ s
   - FEM-solution
   - POD
   - NMR to see convergence

Test simulations based on training simulations:

1. Out of phase loading - Full training
   - $\boldsymbol{t}_1 = -1000\boldsymbol{e}_1$ MPa, $\boldsymbol{t}_2 = -40\boldsymbol{e}_2$ MPa
   - $\boldsymbol{t} = \boldsymbol{t}_1 \sin\left(2\pi \frac{t}{T_{max}}\right) + \boldsymbol{t}_2 \sin\left(2 \cdot 2\pi \frac{t}{T_{max}}\right)$, $t \in [0, T_{max}]$, $T_{max} = 10\,\text{s}$ with $\Delta t = 0.1\,\text{s}$
   - One cycle in 1-direction and two in 2-direction.
   - FEM-solution
   - POD on combined snapshots from training cases 1 and 2
   - NMR and check convergence for increasing N_R
2. Out of phase loading - Incomplete training
   - $\boldsymbol{t}_1 = -1000\boldsymbol{e}_1$ MPa, $\boldsymbol{t}_2 = -40\boldsymbol{e}_2$ MPa
   - $\boldsymbol{t} = \boldsymbol{t}_1 \sin\left(2\pi \frac{t}{T_{max}}\right) + \boldsymbol{t}_2 \sin\left(2 \cdot 2\pi \frac{t}{T_{max}}\right)$, $t \in [0, T_{max}]$, $T_{max} = 10\,\text{s}$ with $\Delta t = 0.1\,\text{s}$
   - Reduced basis only using snapshots from vertical training
   - FEM-solution
   - POD on snapshots from training case 1, only vertical snapshots
   - NMR and check convergence for increasing N_R
3. Larger vertical amplitude and longer time
   - $\boldsymbol{t}_1 = -1000\boldsymbol{e}_1$ MPa, $\boldsymbol{t}_2 = -80\boldsymbol{e}_2$ MPa
   - $\boldsymbol{t} = \boldsymbol{t}_1 \sin\left(2\pi \frac{t}{T_{max}}\right) + \boldsymbol{t}_2 \sin\left(2 \cdot 2\pi \frac{t}{T_{max}}\right)$, $t \in [0, T_{max}]$, $T_{max} = 100\,\text{s}$ with $\Delta t = 1.0\,\text{s}$
   - Doubled vertical amplitude and ten times the loading time
   - FEM-solution
   - POD on combined snapshots from training cases 1 and 2
   - NMR and check convergence for increasing N_R

5.2.1 Training: Vertical load

This section presents the FEM-solution from training case 1 in Section 5.2. The presented NMR-simulation aims to re-identify the training simulation based directly on the snapshots from the FEM-solution.

5.2.1.1 FEM

Convergence of tip displacements for increasing number of elements can be seen in Figure 5.8. As can be seen, the number of elements needed to converge varies between directions. With horizontal load, the 1-direction, the average tip displacement is almost constant independent of number of elements. For a vertical load, in 2-direction, it is a more distinct convergence curve. At about 19600 elements it seems to converge, but already at 6400 elements it was assumed to be sufficiently converged for use in the following simulations. Some of the meshes used can be seen in Figure 5.9.

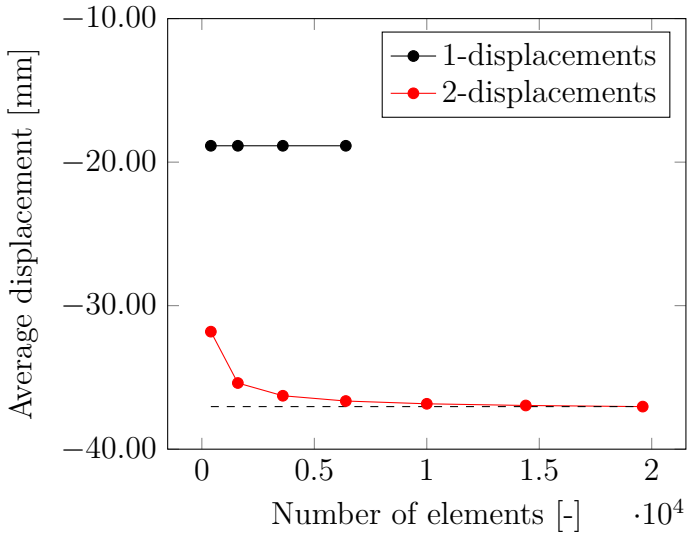
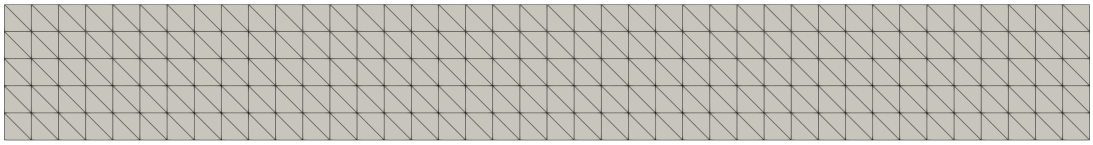
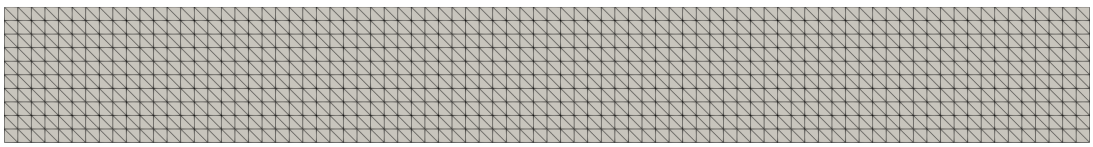


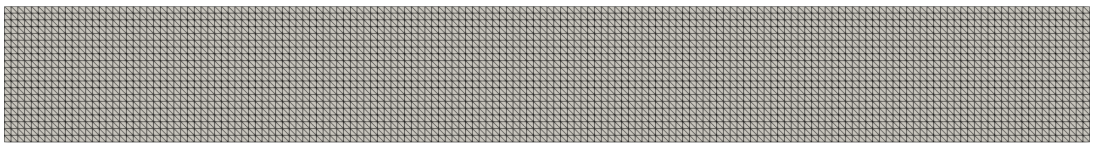
Figure 5.8: Average tip displacements for linear ramp in each direction separately for increasing number of elements in the FEM-solution using CST elements.



(a) Mesh with 400 elements.



(b) Mesh with 1600 elements.



(c) Mesh with 6400 elements.

Figure 5.9: Meshes used for refinement.

The FEM-solution of the problem can be viewed in figure 5.15 with linear loading to $\boldsymbol{t} = \boldsymbol{t}_2$. Note the stress concentration at the middle of the lower edge, which is from a singularity due to the boundary conditions.

5.2.1.2 POD

The first 20 normalized eigenvalues of the correlation matrix g, obtained from POD in the vertical training case, can be seen in figure 5.10. With the cut-off $\beta = 10^{-6}$, 7 modes were extracted.

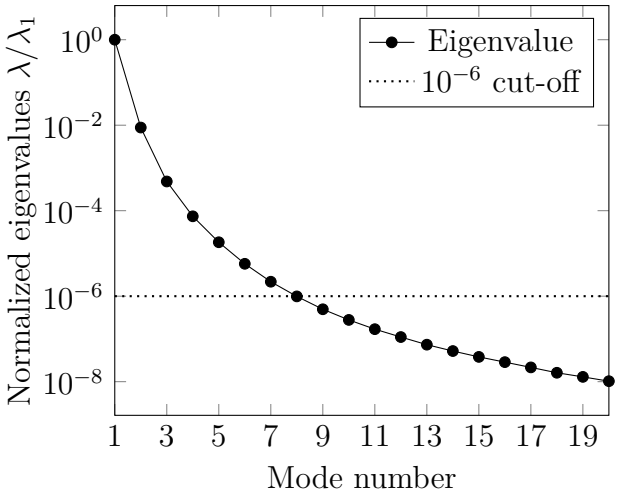
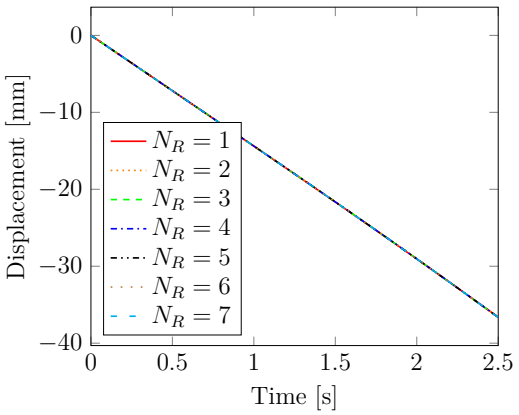
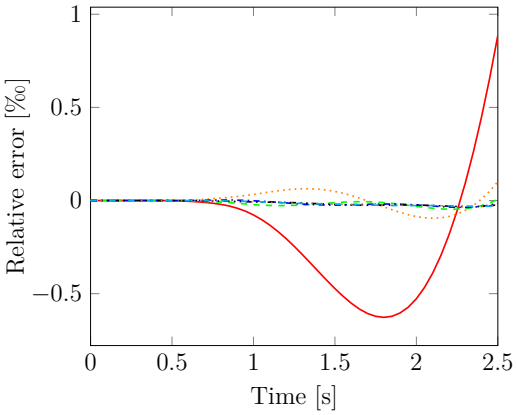


Figure 5.10: The first 20 normalized eigenvalues of the correlation matrix g, obtained from the snapshots for the vertical training using POD. 7 values above the cut-off.

5.2.1.3 NMR



(a) Vertical average tip displacements u_2.



(b) Relative displacement error between NMR and FEM in vertical displacement u_2.

Figure 5.11: NMR-solution over time for vertical training using 6400 elements. The relative difference decreases fast with increased number of modes. The higher amount of modes are so close to zero that they are hardly visible on the current axis. The relative displacement error is small, which mean that solution is very close to the FEM-solution. That is to be expected since POD was performed on snapshots from the FEM-solution.

The NMR-solution for vertical direction can be seen in Figure 5.11. The relative displacement error is very small, especially when using many modes. Therefore is the NMR-solution very close to the FEM-solution in Figure 5.15.

Figure 5.12 shows convergence of the average vertical tip displacement. It converges fast for increasing number of modes, which means that the model reduction gives an accurate description of the FEM-solution. Though, a small difference to the FEM-solution was obtained. The error for $N_R = 7$ is about $3 \cdot 10^{-3}$ %, which might be caused by using to few modes or due to numerical errors in the solving process.

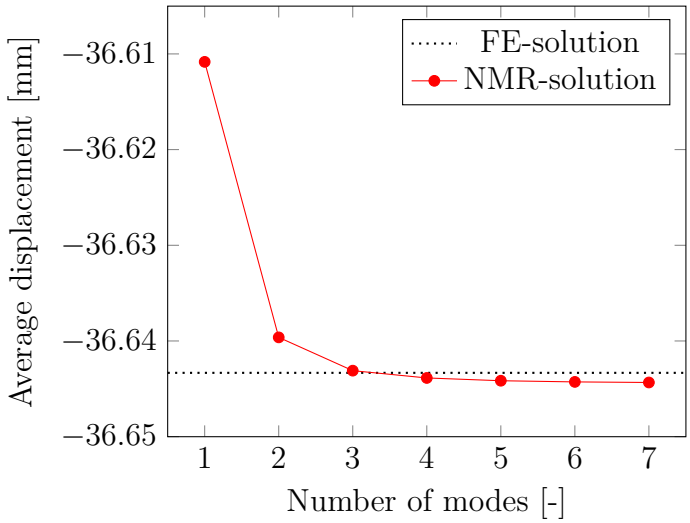
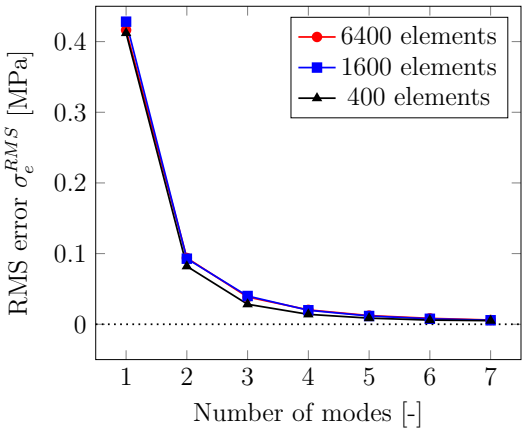
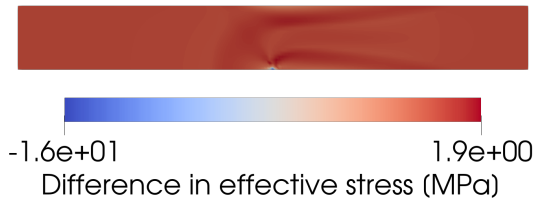


Figure 5.12: Convergence of average vertical tip displacement for increasing number of modes at $t = T_{max}$ for linear ramp with traction in vertical direction. A mesh with 6400 elements was used. Note that a small difference to the FEM-solution was obtained.

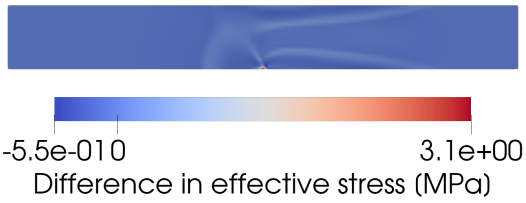
The RMS error calculated according to Equation 4.2 can be seen in Figure 5.13 for different number of elements. The error converges to zero for increased number of elements, meaning a more accurate solution. Figure 5.13 also shows the difference in effective stress, $\sigma_e^{NMR} - \sigma_e^{FEM}$. The difference is smaller for more modes, and this is the reason for the RMS error converging to zero. Note that at the stress concentration at the middle lower edge, the difference is usually the highest. In the rest of the domain, the difference in stress is more uniform, but with an oscillating pattern, possibly due to the higher modes.



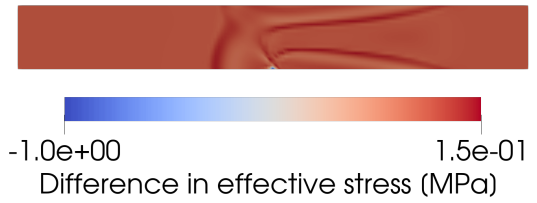
(a) RMS error σ_e^{RMS}.



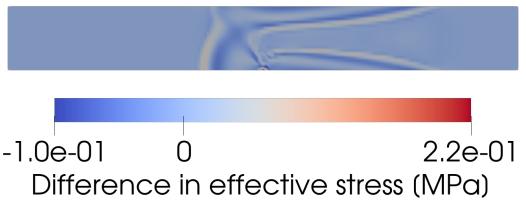
(b) Difference in effective stress for $N_R = 1$.



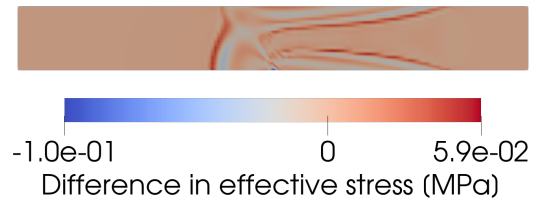
(c) Difference in effective stress for $N_R = 2$.



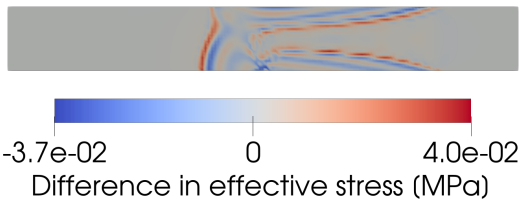
(d) Difference in effective stress for $N_R = 3$.



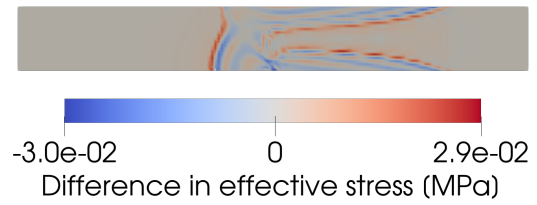
(e) Difference in effective stress for $N_R = 4$.



(f) Difference in effective stress for $N_R = 5$.



(g) Difference in effective stress for $N_R = 6$.



(h) Difference in effective stress for $N_R = 7$.

Figure 5.13: RMS error σ_e^{RMS} for vertical training case and the difference between effective stress in NMR and FEM for 6400 elements. The RMS error converges and the range in effective stress in the difference plots also gets smaller.

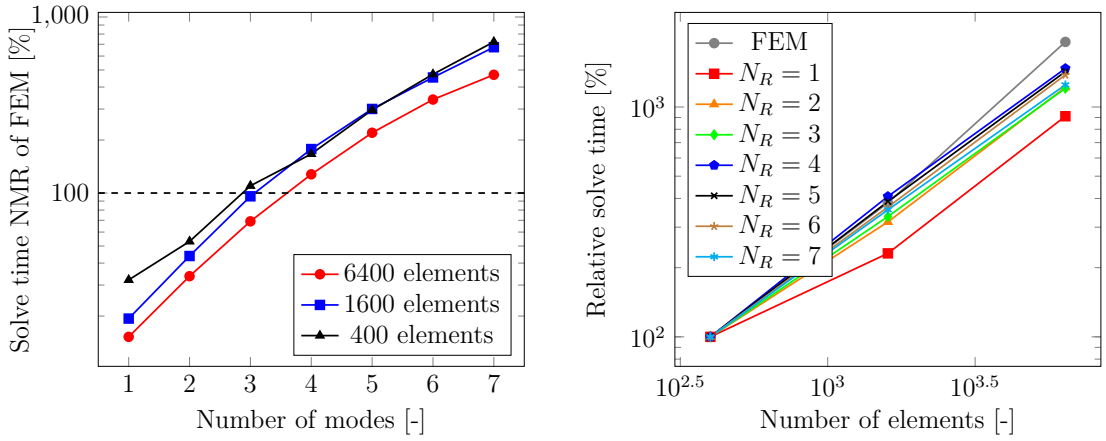
The time to solve FEM and NMR is presented in Table 5.3.

Table 5.3: Solve time in seconds for FEM and NMR for different number of elements. Only the online stage was timed for NMR.

		NMR						
Elements	FEM	$N_R = 1$	$N_R = 2$	$N_R = 3$	$N_R = 4$	$N_R = 5$	$N_R = 6$	$N_R = 7$
400	8.1	2.6	4.3	8.9	13.5	24.0	38.3	58.5
1600	31.0	6.0	13.6	29.7	55.1	93.1	140.6	208.8
6400	155.5	23.7	52.4	107.4	198.6	342.1	528.2	730.3

Figure 5.14a shows the solve time for NMR as a percentage of corresponding time for FEM for increasing number of modes. Clearly, more modes takes longer time to solve. This is to be expected because more modes means more unknowns ξ_a, and more complicated assembly of matrices. It is also obvious that an increase in number of elements also results in a bigger advantage of using NMR over FEM.

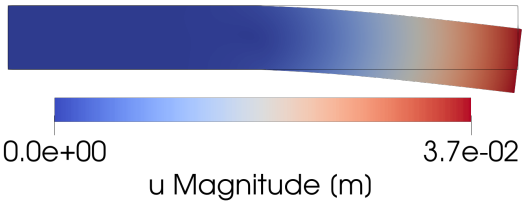
Figure 5.14b shows the solve time for FEM and NMR relative to the time for the coarsest mesh, in this case the one with 400 elements. Note that the graphs for NMR roughly cluster together, meaning the increase in relative solve time is similar regardless of the number of modes. The behaviour of FEM is also slightly different from NMR. It increases a bit faster for increased number of elements than NMR.



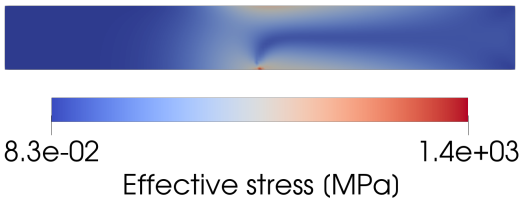
(a) Relative solve time vs. modes. Scaled with respective FEM solve time.

(b) Relative solve time vs. elements. Scaled with time for coarsest mesh.

Figure 5.14: Solve time for NMR as a percentage of corresponding time for FEM for increasing number of modes. Increasing number of elements clearly also increases the advantage of NMR.



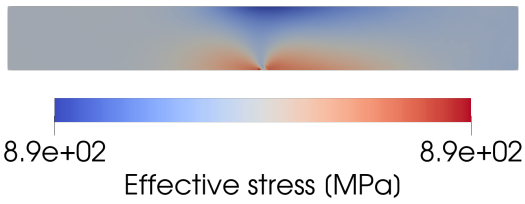
(a) Displacement magnitude at $t = T_{max}$ and outline of undeformed geometry with scale factor 5.



(b) Effective stress σ_e at $t = T_{max}$.

Figure 5.15: FEM-solution at $t = T_{max}$ for vertical training.

(a) Displacement magnitude at $t = T_{max}$ and outline of undeformed geometry with scale factor 5.



(b) Effective stress σ_e at $t = T_{max}$.

Figure 5.16: FEM-solution at $t = T_{max}$ for horizontal training. Note that the range in effective stress is almost the same, meaning the effective stress is almost uniform throughout the geometry.

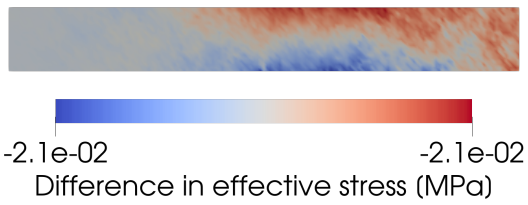


Figure 5.17: Difference in effective stress between NMR and FEM for horizontal training. Note that the range is almost the same, meaning that the difference is almost uniform around $-21\,\mathrm{kPa}$.

5.2.2 Training: Horizontal load

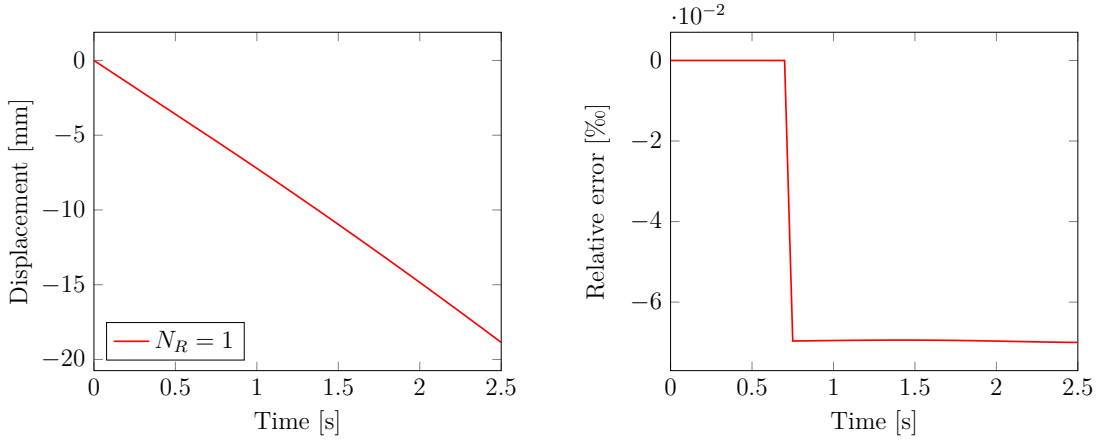
This section presents the FEM-solution from training case 2 in Section 5.2. The presented NMR-simulation aims to re-identify the training simulation based directly on the snapshots from the FEM-solution.

5.2.2.1 FEM

The FEM-soluton at $t = T_{max}$ can be seen in Figure 5.16 with the displacement magnitude and effective stress. Almost all of the displacement occurred in the horizontal direction, expected due to an axial loading. The final horizontal displacement was $-18.8599\,\text{mm}$. Note that the number of decimals given does not indicate the level of accuracy, it's only for comparing the displacements to NMR. The effective stress in Figure 5.16 is almost uniform in the domain, which is also expected due to axial loading and the left edge being simply supported, allowing expansion in the vertical direction.

5.2.2.2 NMR

The NMR-solution for horizontal displacements and relative displacement error can be seen in Figure 5.18. Only one mode was extracted and the final horizontal displacement was $-18.8612\,\text{mm}$, very close to the FEM-solution with a difference of $0.0013\,\text{mm}$. The relative displacement error was very low, at a maximum about $7 \cdot 10^{-2}$ ‰. The solution using NMR is therefore very close to to the solution using FEM in Figure 5.16. That is to be expected, since POD was directly applied on the snapshots from the FEM-solution.



(a) Vertical average tip displacements u_1.

(b) Relative displacement error between NMR and FEM in vertical displacement u_1.

Figure 5.18: NMR-solution over time for horizontal training using 6400 elements. The relative displacement error is small for the only mode. Therefore, the solution is very close to the FEM-solution in Figure 5.16. That is to be expected since POD was directly performed on snapshots from the FEM-solution.

In the horizontal training case, the only mode had a RMS error in effective stress of about 21 kPa, much smaller than for one mode in the vertical training in Figure 5.13. Figure 5.17 shows the difference in effective stress, $\sigma_e^{NMR} - \sigma_e^{FEM}$ with an almost uniform distribution around -21 kPa, which is the reason for the similar RMS value.

The solve time using FEM and NMR can be seen in Table 5.4 and the relative solve time to respective coarsest mesh can be seen in Figure 5.19. The time to solve FEM increases faster than NMR for increased number of elements. This means that the advantage of NMR increases with the number of elements.

Table 5.4: Solve time in seconds for horizontal training. Only the online stage was timed for NMR.

		NMR
Elements	FEM	$N_R = 1$
400	13.6	3.5
1600	63.4	9.0
6400	509.4	36.0

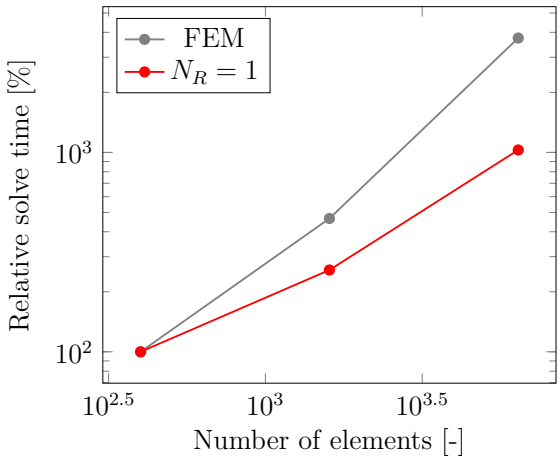


Figure 5.19: Relative solve time vs. elements scaled with time for coarsest mesh for horizontal training.

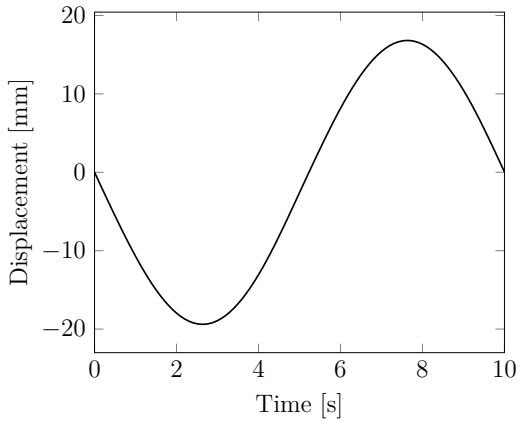
5.2.3 Test: Out of phase loading - Full training

This section presents the results for test case 1 in Section 5.2. The training for NMR is based on the two training cases, both vertical and horizontal loads.

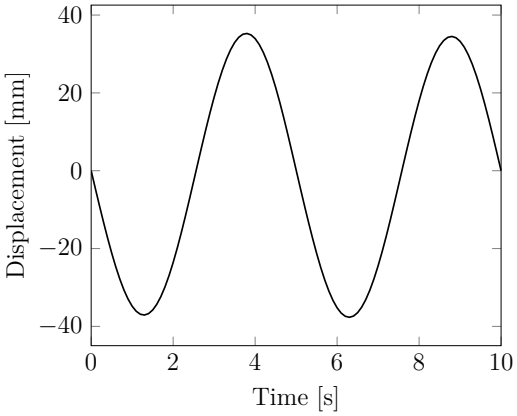
5.2.3.1 FEM

The FEM-solution of the tip displacement over time are shown in Figure 5.20. As expected from the loading, one cycle in horizontal direction and two in the vertical

direction. The amplitudes of the displacements are roughly the same, about 20 mm and 40 mm respectively. The displacement magnitude and residual effective stress at $t = T_{max}$ can also be seen in Figure 5.20. The displacements are plotted with an outline of the undeformed geometry and with a scale factor of 1000. No traction was applied in this time step, so this is the final plastic deformation. The residual effective stress is the stress left in the material after the deformation. At the middle of the lower edge, a stress concentration was formed.

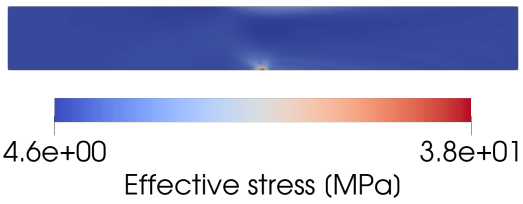


(a) Horizontal displacements u_1.



(b) Vertical displacements u_2.

(c) Displacement magnitude at $t = T_{max}$ and outline of undeformed geometry with scale factor 1000.



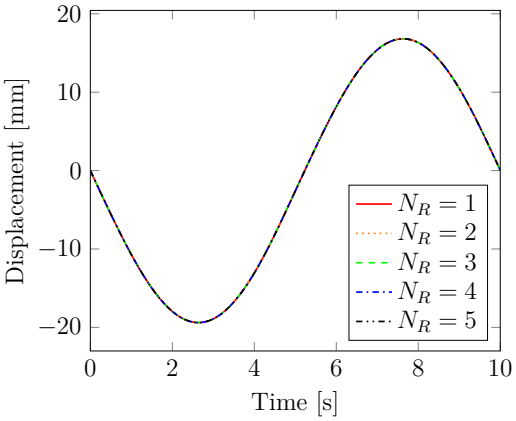
(d) Residual effective stress σ_e at $t = T_{max}$.

Figure 5.20: FEM-solution over time for multiple loads. In the Horizontal direction one cycle, and in the vertical direction two cycles. A mesh with 6400 elements was used.

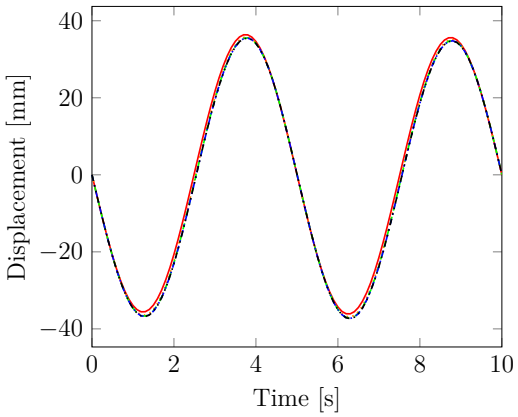
5.2.3.2 NMR

Snapshots from the two training simulations were used to construct a reduced basis for NMR using POD by putting the matrices containing the snapshots side by side in a new matrix, as described in Section 4.5. The average tip displacements vs. time can be seen in Figure 5.21 together with the relative displacement error to the FEM-solution calculated with Equation 4.1. In the horizontal direction, there were little change in solution for increased number of modes, because the first mode was the one describing horizontal movements. As was seen in the horizontal training case, only one mode was extracted. Therefore adding more modes does not change the

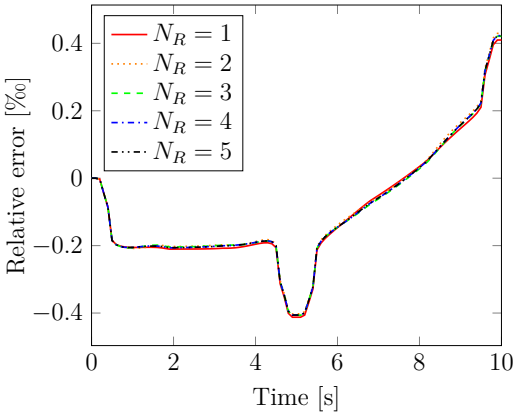
solution substantially. This probably also explains why the error is very small, only about 0.4 ‰ at most. Interestingly, at about 5 s the relative displacement error sharply dips. This is at a point in the loading when both loading directions are close to zero, which means that the displacements also are close to zero. The small displacements might then be slightly more difficult to resolve, but it's important to note that the error is still very small.



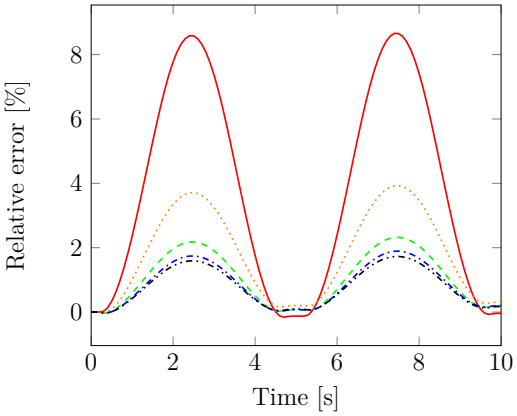
(a) Horizontal displacements u_1.



(b) Vertical displacements u_2.



(c) Relative displacement error between NMR and FEM in horizontal displacement u_1.



(d) Relative displacement error between NMR and FEM in vertical displacement u_2.

Figure 5.21: NMR-solution over time for out of phase test case with 6400 elements. Overall very similar to the FEM-solution in Figure 5.20. Note that the relative difference in the horizontal direction is in per mille, ‰, and in the vertical direction percent, %. The error is much larger in the vertical direction than in the horizontal direction.

In the vertical direction in Figure 5.21, the differences are much more noticeable. For one mode the error is at most more than 8 %, and then decreases for more modes. This is an expected behaviour, because the first mode mostly describes the horizontal displacements, and the rest of the modes mostly describe the vertical displacements. Therefore more modes should mean more accurate capture of the real

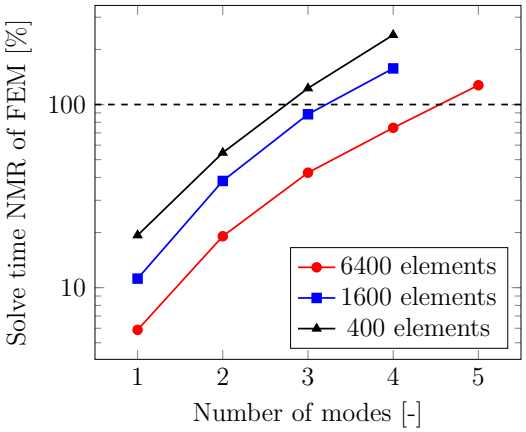
behaviour. It is also important to note that the training cases both used a linear ramp, and the current test case a cyclic sine function. This will change the loading rate, and thus might not be represented in the training snapshots.

The time to solve the problem using using FEM and NMR can be seen in Table 5.5. NMR is clearly much faster for fewer modes.

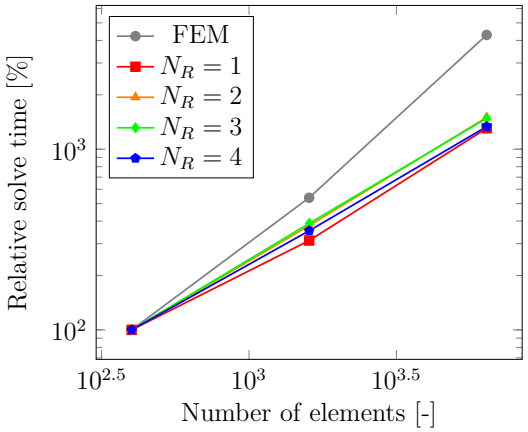
Table 5.5: Time in seconds to solve with FEM and NMR. For NMR only the online stage was timed.

		NMR				
Elements	FEM	$N_R = 1$	$N_R = 2$	$N_R = 3$	$N_R = 4$	$N_R = 5$
400	29.5	5.7	16.1	36.2	70.8	-
1600	158.9	17.8	60.8	140.6	250.3	-
6400	1265.2	74.4	241.5	537.0	944.0	1613.0

Figure 5.22a shows the solve time for the online phase of NMR relative to FEM vs. number of modes. Increasing the number of elements clearly increases the advantage of NMR. As expected, more modes take longer time to solve for, and thus the time advantage of NMR decreases. Note that the advantage fully disappear if NMR relative to FEM is above 100 %.



(a) Relative solve time vs. modes. Scaled with respective FEM solve time.

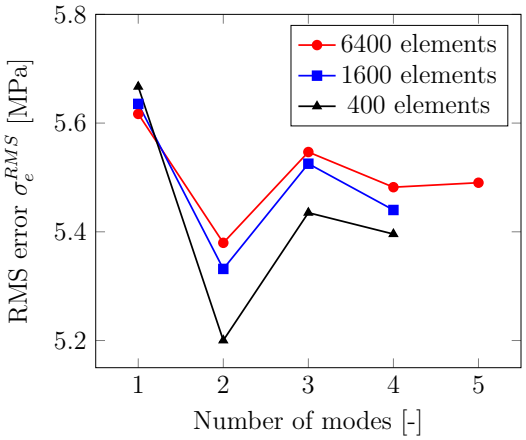


(b) Relative solve time vs. elements. Scaled with time for coarsest mesh.

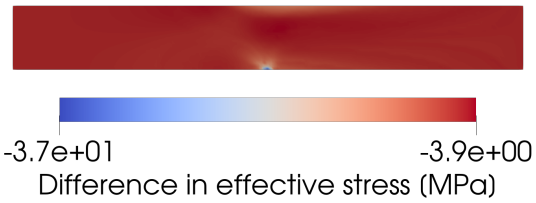
Figure 5.22: Solve time for online NMR-solution as a percentage of corresponding FEM-solution for increasing number of modes, and solve time for FEM and NMR as a percentage of solve time for respective coarsest mesh. Increasing the amount of elements clearly also increases the advantage of NMR. The solve time for FEM increases faster for increased number of elements than NMR.

The relative solve time vs. elements can be seen in Figure 5.22b. The solve time is compared to the coarsest mesh for FEM and NMR for each N_R. Increasing the number of elements increases, as expected, the solve time, but faster for FEM than for NMR. This also shows why the advantage of NMR increases for increased

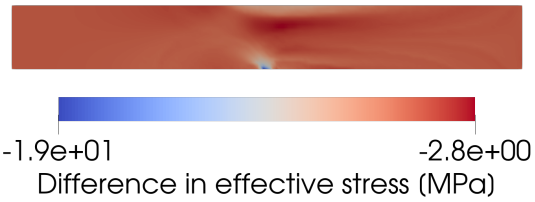
number of elements. The lines for NMR cluster around each other, and therefore is the increase in relative solve time similar, regardless of number of modes.



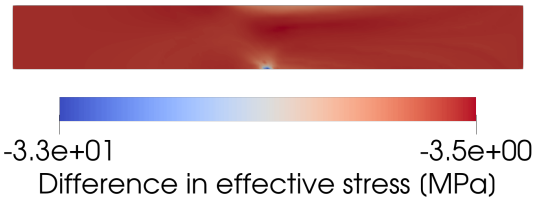
(a) RMS error σ_e^{RMS}.



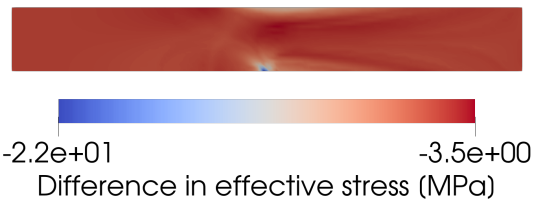
(b) Difference in effective stress for $N_R = 1$.



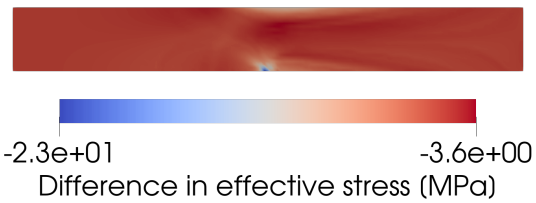
(c) Difference in effective stress for $N_R = 2$.



(d) Difference in effective stress for $N_R = 3$.



(e) Difference in effective stress for $N_R = 4$.



(f) Difference in effective stress for $N_R = 5$.

Figure 5.23: RMS error σ_e^{RMS} at $t = T_{max}$ for out of phase test case together with the difference between effective stress in NMR and FEM. The RMS error only decreases by a small amount and the differences in effective stress is similar for all values on N_R. The singularity at the bottom edge has the largest difference in effective stress.

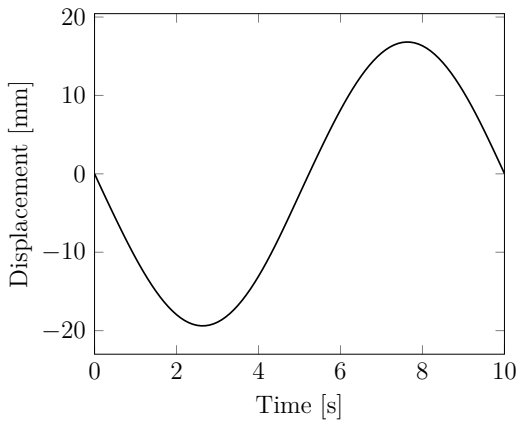
The RMS error σ_e^{RMS} of the effective stress at $t = T_{max}$ is shown in figure 5.23. As can be seen, it is much larger and only gets slightly better for increased number of modes compared to the convergence in the vertical training case in Figure 5.13. The differences in effective stress $\sigma_e^{NMR} - \sigma_e^{FEM}$ at $t = T_{max}$ can also be seen in Figure

5.23. The differences are about the same for increased number of modes. The differences are also biggest in the middle of the lower edge, at the stress concentration. This means that it is more difficult for NMR to capture the stress concentration, possibly due to it being a singularity. Most of the domain, though, is dominated by values much smaller, about $-5\,\text{MPa}$, which is similar to the RMS error value. The differences are consistently negative across all of the plots, meaning that the NMR-solution σ_e^{NMR} is always smaller than the FEM-solution σ_e^{FEM}. This contributes heavily to the RMS error.

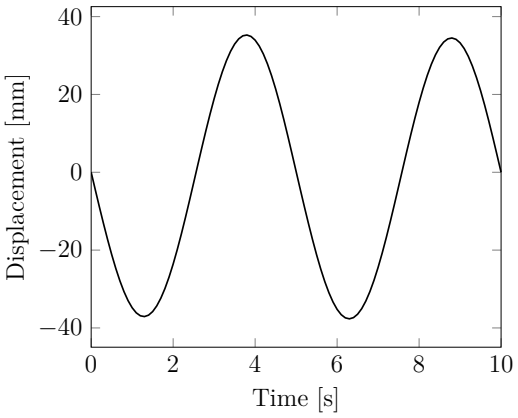
5.2.4 Test: Out of phase loading - Incomplete training

This section presents the results for test case 2 in Section 5.2. The training for NMR is only based on the first training case, vertical load.

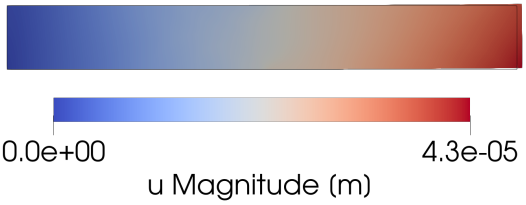
5.2.4.1 FEM



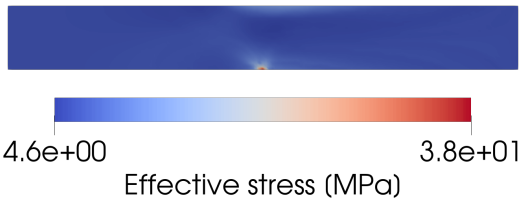
(a) Horizontal displacements u_1.



(b) Vertical displacements u_2.



(c) Displacement magnitude at $t = T_{max}$ and outline of undeformed geometry with scale factor 1000.

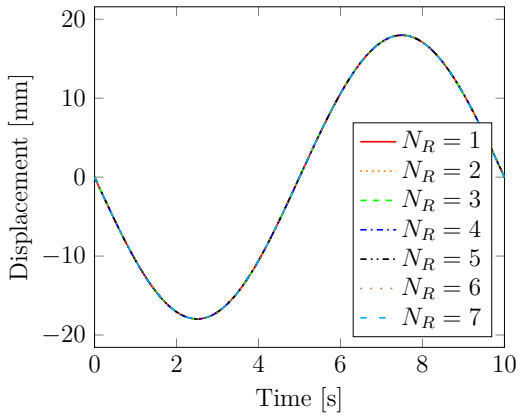


(d) Residual effective stress σ_e at $t = T_{max}$.

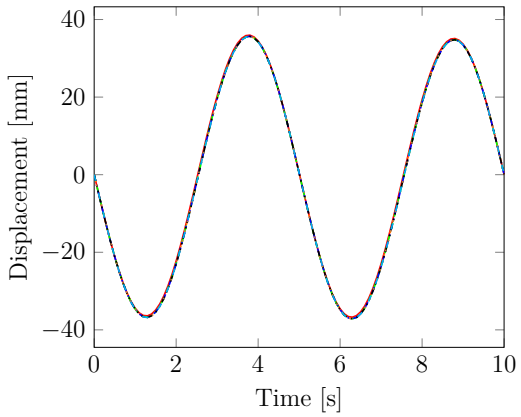
Figure 5.24: FEM-solution over time, same as in out of phase loading. In the horizontal direction one cycle, and in the vertical direction two cycles. A mesh with 6400 elements was used.

Because the only thing changed to the previous test case, out of phase loading, is which snapshots that were used, the FEM-solution will be the same in the current test case as in the previous. The solution can be seen in Figure 5.24.

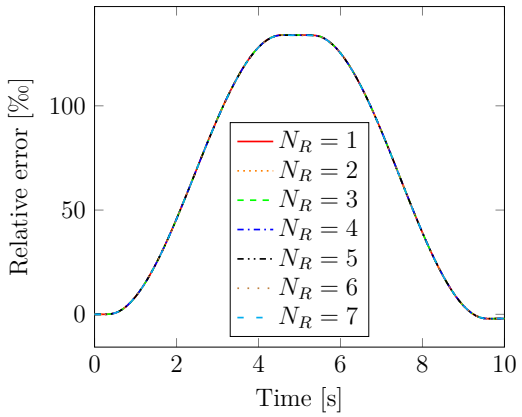
5.2.4.2 NMR



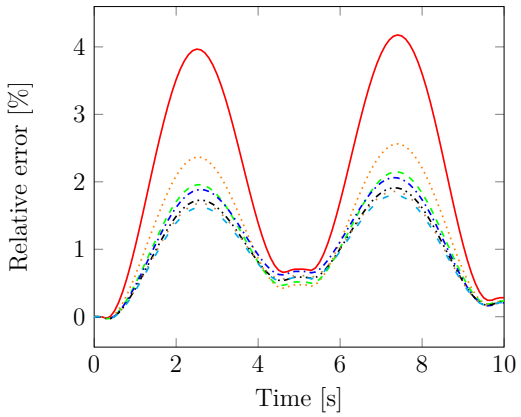
(a) Horizontal displacements u_1.



(b) Vertical displacements u_2.



(c) Relative displacement error between NMR and FEM in horizontal displacement u_1.



(d) Relative displacement error between NMR and FEM in vertical displacement u_2.

Figure 5.25: NMR-solution over time when only using vertical snapshots. The mesh had 6400 elements. A similar behaviour to the NMR-solution for out of phase loading test case in Figure 5.21, but with a larger error in the horizontal direction. Note that the relative displacement error in the horizontal direction is in per mille, ‰, and in the vertical direction percent, %.

The NMR-solution can be seen in Figure 5.25. Because only the snapshots from the vertical training case were used, the same number of modes was extracted, $N_R = 7$. This also meant that the mode from the horizontal training case was excluded, which is reflected by an increased error in the horizontal direction as compared to the out of phase loading test case in Figure 5.21. Similar to the previous test case is that all of the modes have almost the same error in the horizontal direction. In the vertical

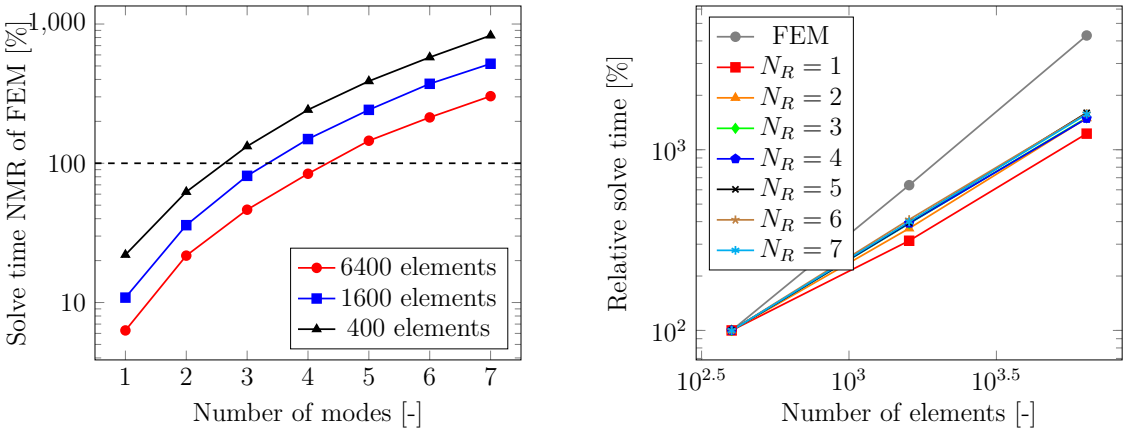
direction they are also similar, but slightly lower due to $N_R = 1$ in the previous case only contained the horizontal mode. Increasing the number of modes also decreases the error, as in the previous test case.

The solve time for only using vertical snapshots can be seen in Table 5.6. NMR is clearly faster when using few modes. Using many modes can take longer time to solve than for FEM.

Table 5.6: Time in seconds to solve with FEM and NMR. For NMR only the online stage was timed.

		NMR						
Elements	FEM	$N_R = 1$	$N_R = 2$	$N_R = 3$	$N_R = 4$	$N_R = 5$	$N_R = 6$	$N_R = 7$
400	26.8	5.9	16.7	35.5	64.9	104.2	154.6	221.8
1600	170.6	18.5	61.3	138.6	254.6	412.7	635.0	884.6
6400	1148.6	72.4	249.6	533.0	966.0	1667.5	2452.1	3482.8

Figure 5.26 shows the solve time for the online phase of NMR relative to FEM vs. number of modes. Increasing the number of elements also increases the advantage of NMR over FEM. More modes also take longer time to solve and if the solve time for NMR relative to FEM is larger than 100 %, the advantage of NMR fully disappear. Using many modes can in some cases take much longer time to solve than using FEM. This is a similar result to the previous test case in Figure 5.22.



(a) Relative solve time vs. modes. Scaled with respective FEM solve time.

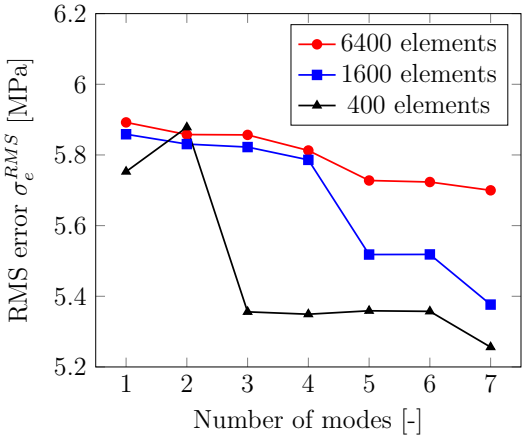
(b) Relative solve time vs. elements. Scaled with time for coarsest mesh.

Figure 5.26: Solve time for online NMR-solution when using only vertical snapshots as a percentage of corresponding FEM-solution for increasing number of modes, and solve time for FEM and NMR as a percentage of respective coarsest mesh. Increasing the amount of elements clearly also increases the advantage of NMR. These results are similar to Figure 5.22.

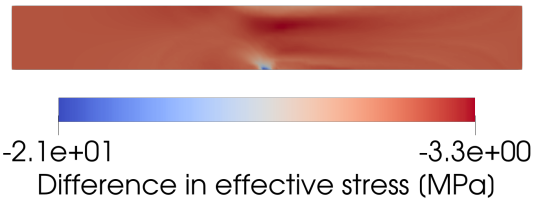
Figure 5.26 also shows the solve time for FEM and NMR as compared to respective coarsest mesh vs. number of elements. The solve time for FEM increases faster

for FEM than NMR and this is why the advantage of NMR increases for increased number of elements. Also this result is similar to the previous test case in Figure 5.22.

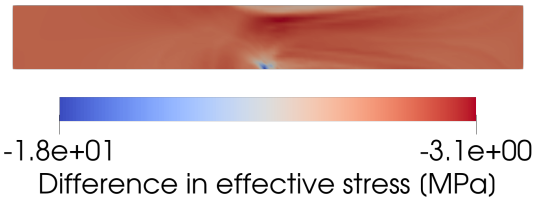
The RMS error σ_e and difference in effective stress $\sigma_e^{NMR} - \sigma_e^{FEM}$ can be seen in Figure 5.27 and behaves in a similar way to the previous test case in Figure 5.23. There might be a slightly better convergence than previously, but difficult to say anything conclusive about it. The differences in effective stress are also similar. In all but one case are they strictly negative, meaning $\sigma_e^{NMR} < \sigma_e^{FEM}$, and the biggest difference in effective stress is at the middle of the lower side, the stress concentration. This means that also for this case is it difficult for NMR to capture the behaviour.



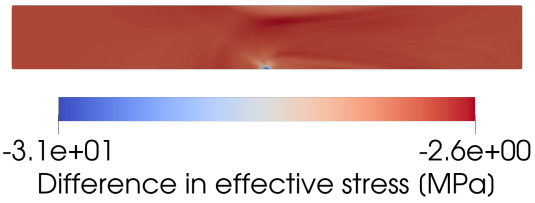
(a) RMS error σ_e^{RMS}.



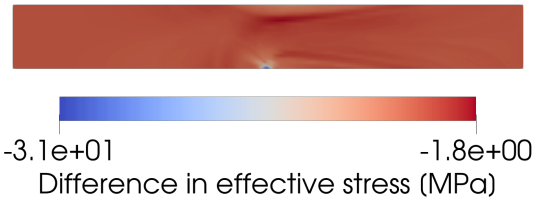
(b) Difference in effective stress for $N_R = 1$.



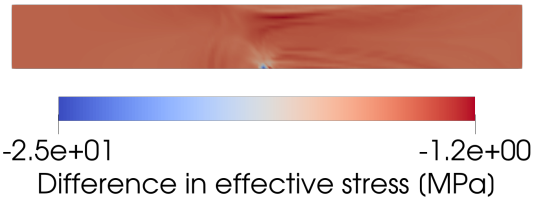
(c) Difference in effective stress for $N_R = 2$.



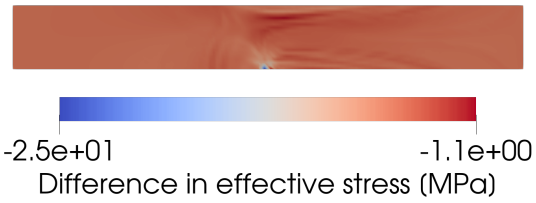
(d) Difference in effective stress for $N_R = 3$.



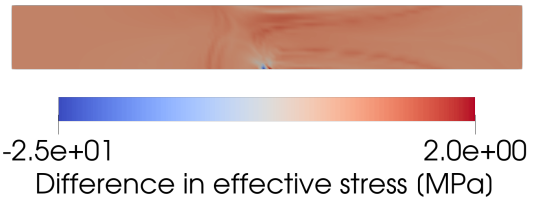
(e) Difference in effective stress for $N_R = 4$.



(f) Difference in effective stress for $N_R = 5$.



(g) Difference in effective stress for $N_R = 6$.



(h) Difference in effective stress for $N_R = 7$.

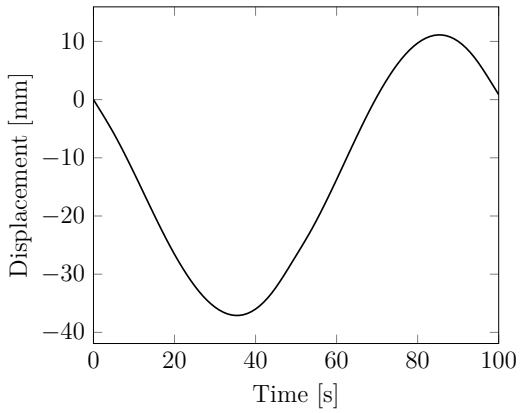
Figure 5.27: RMS error σ_e^{RMS} at $t = T_{max}$ for only vertical snapshots test case, together with with the difference between effective stress in NMR and FEM-solutions. The RMS error only decreased by a small amount and is similar to the out of phase test case, see Figure 5.23. Also the differences are similar, they are almost always negative and the biggest difference is located at the middle of the lower edge.

5.2.5 Test: Larger vertical amplitude and longer time

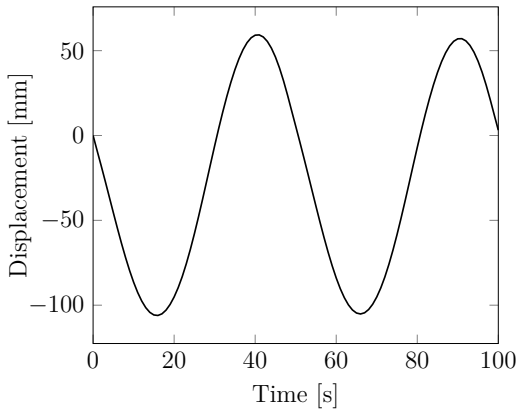
This section presents the results for test case 3 in Section 5.2. The training for NMR is based on the two training cases, both vertical and horizontal loads.

5.2.5.1 FEM

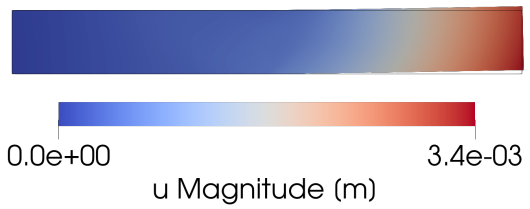
Figure 5.28 shows the horizontal and vertical displacements over time for a mesh with 6400 elements in the larger amplitude and longer time test case. Both of the amplitudes are larger as compared with the similar test case out of phase loading in Figure 5.20, out of phase loading. There is a much more pronounced difference in displacements between positive and negative directions, especially in horizontal direction. The final deformation is much larger, at most 3.4 mm as compared to 0.043 mm in Figure 5.20. Also the residual stresses are much larger, about one order of magnitude larger.



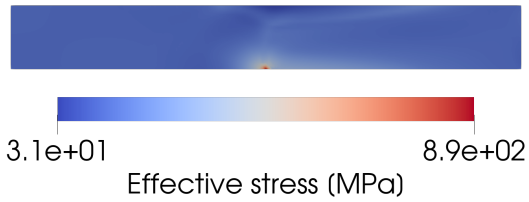
(a) Horizontal displacements u_1.



(b) Vertical displacements u_2.



(c) Displacement magnitude at $t = T_{max}$ and outline of undeformed geometry with scale factor 10.



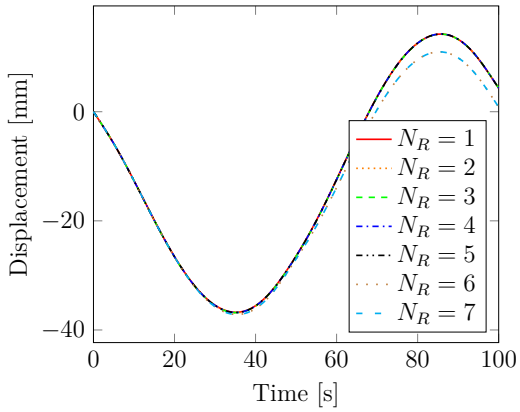
(d) Residual effective stress σ_e at $t = T_{max}$.

Figure 5.28: FEM-solution over time for larger vertical amplitude and longer loading time. In the horizontal direction one cycle, and in the vertical direction two cycles. A mesh with 6400 elements was used. Note the large difference in amplitude in positive and negative directions, especially in in the horizontal direction.

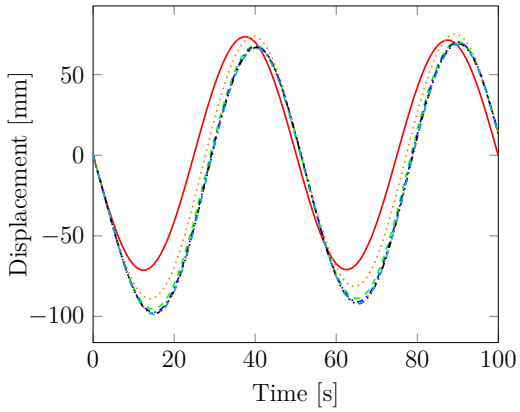
5.2.5.2 NMR

The NMR-solution for the test case larger amplitude and longer time can be seen in Figure 5.29. Because this case had double the vertical amplitude and ten times the loading time, the cut-off for POD was lowered to $\beta = 10^{-7}$ resulting in $N_R = 7$, two more modes than in the test case out of phase loading.

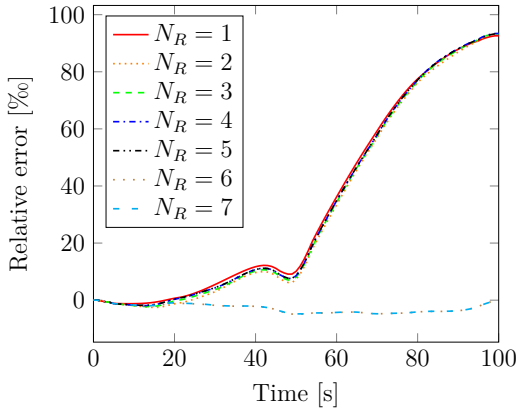
The relative displacement error in the horizontal direction in Figure 5.29 is similar to the corresponding error in Figure 5.21, but is much larger, almost 100 ‰, as compared to 0.4 ‰. However, for $N_R = 6$ and $N_R = 7$ the error is much smaller and clearly much closer to the FEM-solution, as would be expected when using more modes. Also in the vertical direction is the error larger, about 60 % and 8 % respectively for $N_R = 1$.



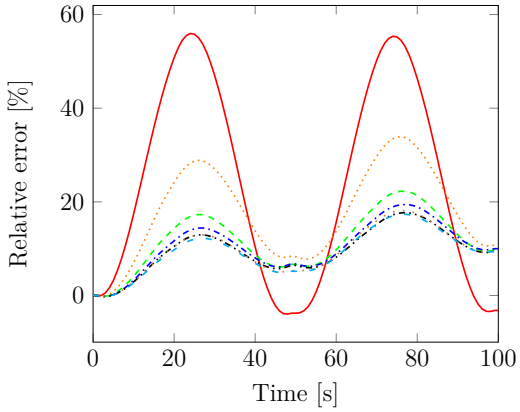
(a) Horizontal displacements u_1.



(b) Vertical displacements u_2.



(c) Relative displacement error between NMR and FEM in horizontal tip displacement u_1.



(d) Relative displacement error between NMR and FEM in vertical tip displacement u_2.

Figure 5.29: NMR-solution over time for the case with larger amplitude and longer loading time. The mesh had 6400 elements. Note that the relative difference in the horizontal direction is in per mille, ‰, and in the vertical direction percent, %.

The vertical displacements in Figure 5.29 for different number of modes N_R, are noticeably different, more than corresponding displacements in Figure 5.21.

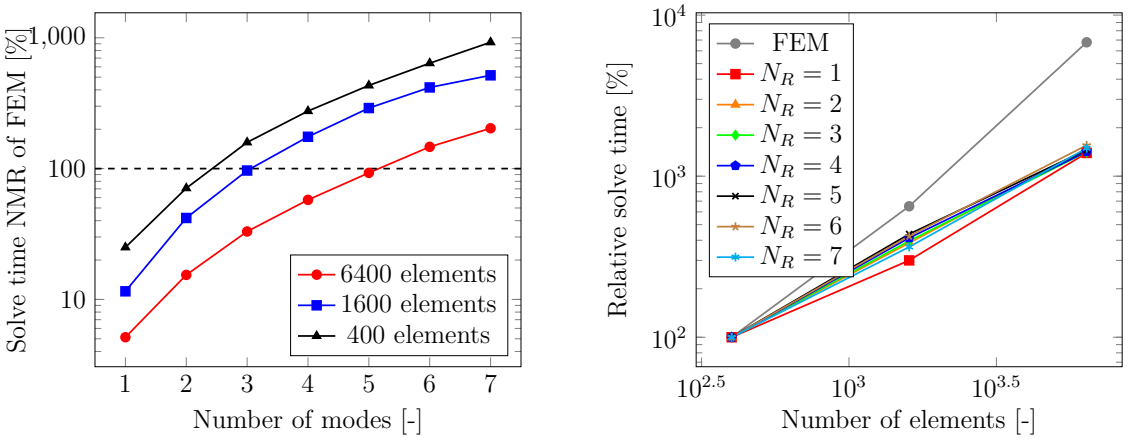
All of these much larger differences can be explained by a larger difference from the training cases. Doubled the vertical traction amplitude and ten times the loading time are big differences to the training and should therefore mean less accurate solutions. This is precisely what has been shown.

The solve time for the test case with larger vertical amplitude and longer loading time can be seen in Table 5.7. For fewer modes, NMR clearly has an advantage over FEM.

Table 5.7: Time in seconds to solve with FEM and NMR. For NMR only the online stage was timed.

		NMR						
Elements	FEM	$N_R = 1$	$N_R = 2$	$N_R = 3$	$N_R = 4$	$N_R = 5$	$N_R = 6$	$N_R = 7$
400	26.4	6.6	18.7	41.9	72.7	114.1	168.8	244.0
1600	171.7	19.8	72.0	166.0	300.7	498.4	716.7	886.9
6400	1790.1	92.0	275.6	591.7	1032.3	1660.8	2626.2	3640.5

The solve time for the online phase of NMR relative to FEM vs. number of modes can be seen in Figure 5.30. As previously established in Figures 5.22 and 5.26, increasing the number of elements also increases the advantage of NMR over FEM. The same for more modes, and if the solve time for NMR relative to FEM is larger than 100 %, the advantage of NMR fully disappear. Using many modes can in some cases take much longer time to solve than using FEM.



(a) Relative solve time vs. modes. Scaled with respective FEM solve time.

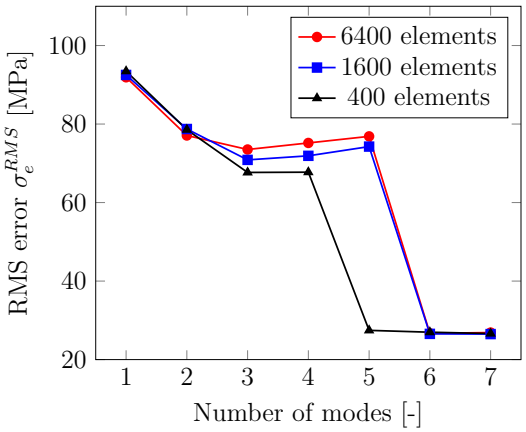
(b) Relative solve time vs. elements. Scaled with time for coarsest mesh.

Figure 5.30: Solve time for online NMR-solution for the test case larger vertical amplitude and longer loading time, as a percentage of corresponding FEM-solution for increasing number of modes. The solve time for FEM and NMR as a percentage of respective coarsest mesh vs. number of elements are also shown. Increasing the amount of elements clearly also increases the advantage of NMR. These results are similar to the previous test cases in Figures 5.22 and 5.26.

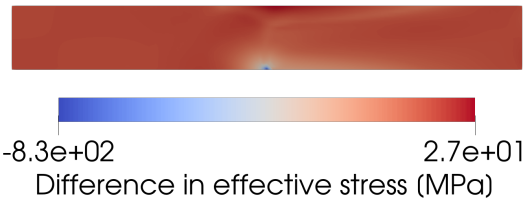
The solve time for FEM and NMR as compared to respective coarsest mesh vs. number of elements can be seen in Figure 5.30 and behaves similarly to the previous cases. The solve time for FEM increases faster for FEM than NMR and is part of the explanation why the advantage of NMR increases for increased number of elements.

The RMS error σ_e^{RMS} can be seen i Figure 5.31 together with the difference in effective stress between NMR- and FEM-solutions, $\sigma_e^{NMR} - \sigma_e^{FEM}$. These result differ from the previous test cases in Figures 5.23 and 5.27, the RMS error is much larger and decreases more. This can be because a lower cut-off was used and therefore more modes extracted, meaning a better NMR-solution. This is also a favorable result showing that NMR is generally more accurate for more modes.

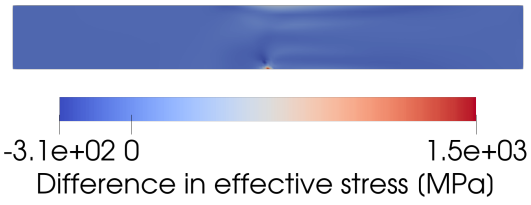
Another difference between the current test case in Figure 5.31 and the previous test cases in Figures 5.23 and 5.27, is that the difference in effective stress is both positive and negative for all N_R. Previously it was almost always negative. However, it is also important to point out that the differences can be much larger in magnitude.



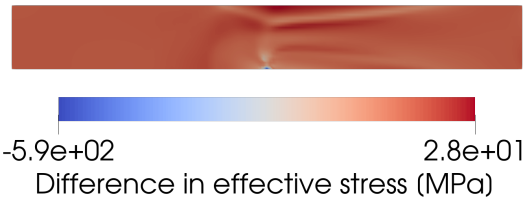
(a) RMS error σ_e^{RMS}.



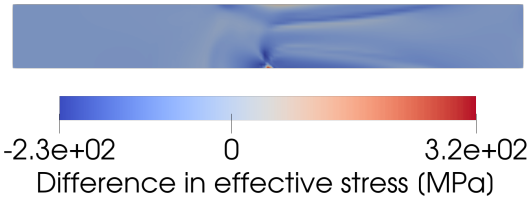
(b) Difference in effective stress $N_R = 1$.



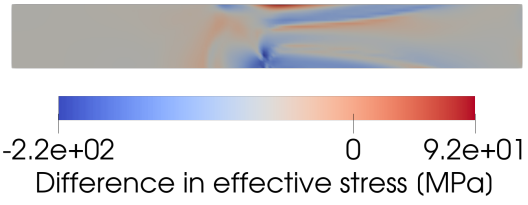
(c) Difference in effective stress $N_R = 2$.



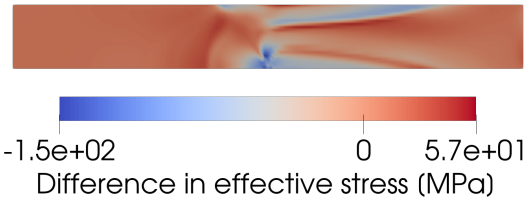
(d) Difference in effective stress $N_R = 3$.



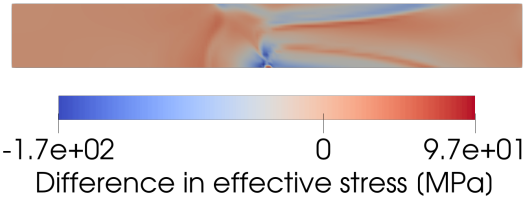
(e) Difference in effective stress $N_R = 4$.



(f) Difference in effective stress $N_R = 5$.



(g) Difference in effective stress $N_R = 6$.



(h) Difference in effective stress $N_R = 7$.

Figure 5.31: RMS error σ_e^{RMS} at $t = T_{max}$ for test case larger vertical amplitude and longer loading time, together with the difference in effective stress between NMR- and FEM-solutions. The RMS error is much larger compared to Figures 5.23 and 5.27, but decreases more thanks to more modes used. Another big difference to the previous test cases is that the differences in effective stress is both negative and positive.

6

Conclusions and outlook

6.1 Conclusions

This thesis has shown that it is possible to apply numerical model reduction to the viscoplasticity problem. The theoretical framework has been presented and implemented in the programming language Julia. The methodology has been assessed for a sequence of test cases with varying complexity of the load, richness of training simulations and discretization, both in number of POD-modes and the underlying FE-discretization.

It has been demonstrated that the computational cost can be severely reduced by using NMR. The time advantage of NMR over FEM increased for more elements in the mesh, but decreased for increased number of modes. If a too high number of modes were used, the time advantage completely disappeared, and NMR took longer time than FEM. It is here important to note that what has been demonstrated is not the best possible speedups of the calculations. The calculation off the elastic displacements $\boldsymbol{u}_0$ was included in the online stage and therefore added to the measured time. One alternative strategy would be to parameterize the loads $\boldsymbol{b}$ and $\boldsymbol{t}_{\mathrm{P}}$ and then scale according to the load amplitudes in the current test case. This would allow for precomputation of the elastic displacements and could further reduce the solve time for the online stage.

It was shown that the best accuracy of the model reduction was obtained when the targeted simulation is similar to the training used to develop the POD-basis. The extreme case, when the NMR-simulation aims to re-identify the training simulation, shows excellent accuracy. Test cases with big differences as compared to the training cases produced lower accuracy. However, the robustness of the procedure has been illustrated in terms of near monotonic error convergence, even for bad training data. That is, adding poor modes does not pollute the approximation. Still, correct training is crucial. Adding more variation among the training cases could help with more accurately capturing the test cases. Though, adding more training cases could also increase the number of modes, which would mean longer solve time and the advantage of NMR over FEM would decrease.

The RMS error in effective stress was shown to be smaller for a coarser mesh and higher for a finer mesh. In this comparison, the effective stress for each NMR-solution was compared to the effective stress for FEM using the corresponding mesh.

This does not mean that the error for NMR in the coarse mesh is lower than to the error for NMR in the fine mesh, if both are compared to FEM in the fine mesh. A finer mesh means that the FEM-approximation error is lower. Therefore, NMR based on a finer mesh should mean that the error is lower than the error for NMR based on a coarser mesh.

6.2 Future work and outlook

The finite element formulation used in this thesis used a monolithic formulation, meaning both displacements $\boldsymbol{u}$ and viscoplastic strains $\boldsymbol{\epsilon}^{\text{vp}}$ were solved for simultaneously. This method, however, is a nonstandard way of solving the viscoplasticity problem. What is usually done is to just solve for $\boldsymbol{u}$ using a tangent stiffness matrix and store the values of $\boldsymbol{\epsilon}^{\text{vp}}$ as internal variables. This could affect the solve time for FEM, and then also affect the advantage of NMR over FEM. One possible extension of this work would then be to train using the established method to obtain modes for $\boldsymbol{\epsilon}^{\text{vp}}$. Then, they would be used to solve the NMR-problem, where it would be compared to FEM using $\boldsymbol{\epsilon}^{\text{vp}}$ as internal variables.

Only a handful test cases were used in this thesis, so a natural extension of the work would be to test more cases. Since the Julia code already exists, it would be a straight forward continuation of this work.

This thesis has used CST elements but has also presented the theory necessary for LST elements. One possible continuation for future work would be to test the model reduction also for this kind of element. LST elements usually gives a better approximation of the solution with the same number of elements, as compared to CST elements.

Another possible extensions would be to continue with a full 3D-problem. This would not require much additional theoretical work, only removing the plain strain condition. The finite element formulation would require some more work, but since the Julia package `Tensors.jl` was used, it would be similar to the formulation already derived.

A time consuming part of the model reduction is to calculate for all of the quadrature points. Therefore could hyperreduction be used to reduce the number of quadrature points used to assemble the reduced system. This could then speed up the calculations even more. How this compares to FEM in terms of solve time and accuracy for the problem explored in this thesis could be a possible continuation.

Finally, having demonstrated the variability of the accuracy for different factors, e.g, training, an estimate of the approximation error would allow for error control. This is something crucial for reliable NMR-simulations, and therefore also for NMR being useful.

Bibliography

[1] Alfio Quarteroni and Gianluigi Rozza, eds. *Reduced order methods for modeling and computational reduction*. MS&A - Modeling, simulation and applications 9. Cham: Springer, 2014. 332 pp. ISBN: 978-3-319-02089-1 978-3-319-02090-7.

[2] David Amsallem et al. "Design optimization using hyper-reduced-order models". In: *Structural and Multidisciplinary Optimization* 51.4 (Apr. 2015), pp. 919–940. ISSN: 1615-147X, 1615-1488. DOI: 10.1007/s00158-014-1183-y. URL: http://link.springer.com/10.1007/s00158-014-1183-y (visited on 10/06/2021).

[3] R. Jänicke et al. "Numerical identification of a viscoelastic substitute model for heterogeneous poroelastic media by a reduced order homogenization approach". In: *Computer Methods in Applied Mechanics and Engineering* 298 (1st Jan. 2016), pp. 108–120. ISSN: 0045-7825. DOI: 10.1016/j.cma.2015.09.024. URL: https://www.sciencedirect.com/science/article/pii/S0045782515003187 (visited on 20/05/2021).

[4] Fredrik Ekre, Fredrik Larsson and Kenneth Runesson. "On error controlled numerical model reduction in FE2-analysis of transient heat flow". In: *International Journal for Numerical Methods in Engineering* 119.1 (2019). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.6041, pp. 38–73. ISSN: 1097-0207. DOI: https://doi.org/10.1002/nme.6041. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6041 (visited on 05/06/2021).

[5] Thomas Abrahamsson. *Structural Dynamics and Linear Systems Compute, Test, Calibrate and Validate*. 2019.

[6] S. Roussette, J. C. Michel and P. Suquet. "Nonuniform transformation field analysis of elastic–viscoplastic composites". In: *Composites Science and Technology*. Mechanical Response of Fibre Reinforced Composites 69.1 (1st Jan. 2009), pp. 22–27. ISSN: 0266-3538. DOI: 10.1016/j.compscitech.2007.10.032. URL: https://www.sciencedirect.com/science/article/pii/S0266353807004356 (visited on 20/05/2021).

[7] George J. Dvorak and Yakov Benveniste. "On transformation strains and uniform fields in multiphase elastic media". In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 437.1900 (8th May

1992), pp. 291–310. ISSN: 0962-8444, 2053-9177. DOI: 10.1098/rspa.1992.0062. URL: https://royalsocietypublishing.org/doi/10.1098/rspa.1992.0062 (visited on 02/06/2021).

[8] J. C. Michel and P. Suquet. "Nonuniform transformation field analysis". In: *International Journal of Solids and Structures*. Special issue in Honor of George J. Dvorak 40.25 (1st Dec. 2003), pp. 6937–6955. ISSN: 0020-7683. DOI: 10.1016/S0020-7683(03)00346-9. URL: https://www.sciencedirect.com/science/article/pii/S0020768303003469 (visited on 30/05/2021).

[9] Felix Fritzen and Thomas Böhlke. "Reduced basis homogenization of viscoelastic composites". In: *Composites Science and Technology* 76 (4th Mar. 2013), pp. 84–91. ISSN: 0266-3538. DOI: 10.1016/j.compscitech.2012.12.012. URL: https://www.sciencedirect.com/science/article/pii/S0266353812004198 (visited on 30/05/2021).

[10] Niels Saabye Ottosen and Matti Ristinmaa. *The mechanics of constitutive modeling*. OCLC: ocm60740765. Amsterdam ; London: Elsevier, 2005. 745 pp. ISBN: 978-0-08-044606-6.

[11] J. Lemaître, ed. *Handbook of materials behavior models*. San Diego, CA: Academic Press, 2001. 1231 pp. ISBN: 978-0-12-443341-0 978-0-12-443342-7 978-0-12-443343-4 978-0-12-443344-1.

[12] Francisco Chinesta, Roland Keunings and Adrien Leygue. *The proper generalized decomposition for advanced numerical simulations*. New York: Springer, 2013. ISBN: 978-3-319-02864-4.

[13] D. Ryckelynck. "A priori hyperreduction method: an adaptive approach". In: *Journal of Computational Physics* 202.1 (Jan. 2005), pp. 346–366. ISSN: 00219991. DOI: 10.1016/j.jcp.2004.07.015. URL: https://linkinghub.elsevier.com/retrieve/pii/S002199910400289X (visited on 18/06/2021).

[14] Claes Johnson. *Numerical solution of partial differential equations by the finite element method*. Dover ed. Dover books on mathematics. OCLC: ocn227923865. Mineola, N.Y: Dover Publications, 2009. 278 pp. ISBN: 978-0-486-46900-3.

[15] Stig Larsson and Vidar Thomée. *Partial differential equations with numerical methods*. First softcover printing. Texts in applied mathematics 45. OCLC: 298551827. Berlin Heidelberg: Springer, 2009. 260 pp. ISBN: 978-3-540-88705-8.

[16] Niels Saabye Ottosen and Hans Petersson. *Introduction to the finite element method*. New York: Prentice Hall, 1992. 410 pp. ISBN: 978-0-13-473877-2.

[17] Robert Davis Cook, ed. *Concepts and applications of finite element analysis*. 4th ed. New York, NY: Wiley, 2001. 719 pp. ISBN: 978-0-471-35605-9.

[18] Jeff Bezanson et al. “Julia: A Fresh Approach to Numerical Computing”. In: *SIAM Review* 59.1 (1st Jan. 2017). Publisher: Society for Industrial and Applied Mathematics, pp. 65–98. ISSN: 0036-1445. DOI: 10.1137/141000671. URL: https://epubs.siam.org/doi/10.1137/141000671 (visited on 17/05/2021).

[19] *Ferrite-FEM/Ferrite.jl*. original-date: 2015-09-02T07:53:47Z. 10th May 2021. URL: https://github.com/Ferrite-FEM/Ferrite.jl (visited on 17/05/2021).

[20] Kristoffer Carlsson and Fredrik Ekre. “Tensors.jl — Tensor Computations in Julia”. In: *Journal of Open Research Software* 7 (21st Mar. 2019), p. 7. ISSN: 2049-9647. DOI: 10.5334/jors.182. URL: http://openresearchsoftware.metajnl.com/articles/10.5334/jors.182/ (visited on 17/05/2021).

[21] *Modulus of Rigidity*. URL: https://www.engineeringtoolbox.com/modulus-rigidity-d_946.html (visited on 13/05/2021).

[22] *Metals and Alloys - Bulk Modulus*. URL: https://www.engineeringtoolbox.com/bulk-modulus-metals-d_1351.html (visited on 13/05/2021).

[23] *Young’s Modulus - Tensile and Yield Strength for common Materials*. URL: https://www.engineeringtoolbox.com/young-modulus-d_417.html (visited on 13/05/2021).

A

Appendix 1

A.1 Differential of Perzyna formulation

In this section the Perzyna formulation is first presented, see equation down below. $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}}$ is then derived for the Perzyna formulation and this is called the general derivation. After that $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}}$ is derived for Perzyna with the von Mises yield criterion. After that $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}}$ is derived for Perzyna, von Mises yield criterion and Norton model.

$$\dot{\boldsymbol{\epsilon}}^{\mathrm{vp}} = \lambda \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} = \frac{\eta[\Phi]}{t^*} \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} = \lambda \boldsymbol{f} = \boldsymbol{g}(\boldsymbol{\sigma})$$

A.1.1 General

To derive $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}}$ the differential $\mathrm{d}\boldsymbol{g}$ is expanded

$$\begin{aligned}
\mathrm{d}\boldsymbol{g} &= \mathrm{d}\left(\frac{\eta[\Phi]}{t^*}\right) \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} + \frac{\eta[\Phi]}{t^*} \mathrm{d}\left(\frac{\partial \Phi}{\partial \boldsymbol{\sigma}}\right) \\
&= \frac{1}{t^*} \frac{\partial \eta[\Phi]}{\partial \Phi} \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} : \mathrm{d}\boldsymbol{\sigma} \otimes \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} + \frac{\eta[\Phi]}{t^*} \frac{\partial^2 \Phi}{\partial \boldsymbol{\sigma} \otimes \partial \boldsymbol{\sigma}} : \mathrm{d}\boldsymbol{\sigma} \\
&= \left(\frac{1}{t^*} \frac{\partial \eta[\Phi]}{\partial \Phi} \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} \otimes \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} + \frac{\eta[\Phi]}{t^*} \frac{\partial^2 \Phi}{\partial \boldsymbol{\sigma} \otimes \partial \boldsymbol{\sigma}}\right) : \mathrm{d}\boldsymbol{\sigma}
\end{aligned}$$

This result and using the chain rule leads to

$$\boxed{\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} = \frac{1}{t^*} \frac{\partial \eta[\Phi]}{\partial \Phi} \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} \otimes \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} + \frac{\eta[\Phi]}{t^*} \frac{\partial^2 \Phi}{\partial \boldsymbol{\sigma} \otimes \partial \boldsymbol{\sigma}}} \tag{A.1}$$

A.1.2 Von Mises yield criterion

The Von Mises yield criterion is

$$\Phi = \sigma_e - \sigma_y, \quad \text{with} \quad \sigma_e = \sqrt{\frac{3}{2}} |\boldsymbol{\sigma}_{\mathrm{dev}}|$$

Differentiation of $\frac{\partial \Phi}{\partial \boldsymbol{\sigma}}$ to later insert in Equation A.1

$$\frac{\partial \Phi}{\partial \boldsymbol{\sigma}} = \underbrace{\frac{\partial \Phi}{\partial \sigma_e}}_{=1} \underbrace{\frac{\partial \sigma_e}{\partial \boldsymbol{\sigma}_{\mathrm{dev}}}}_{=\frac{3}{2}\frac{\boldsymbol{\sigma}_{\mathrm{dev}}}{\sigma_e}} : \underbrace{\frac{\partial \boldsymbol{\sigma}_{\mathrm{dev}}}{\partial \boldsymbol{\sigma}}}_{=\mathbf{I}^{\mathrm{sym}}_{\mathrm{dev}}} = \frac{3}{2}\frac{\boldsymbol{\sigma}_{\mathrm{dev}}}{\sigma_e} : \mathbf{I}^{\mathrm{sym}}_{\mathrm{dev}} = \frac{3}{2}\frac{\boldsymbol{\sigma}_{\mathrm{dev}}}{\sigma_e}$$

To derive $\frac{\partial^2 \Phi}{\partial \boldsymbol{\sigma} \otimes \partial \boldsymbol{\sigma}}$ the differential $\mathrm{d}\left(\frac{\partial \Phi}{\partial \boldsymbol{\sigma}}\right)$ is expanded

$$\begin{aligned}
\mathrm{d}\left(\frac{\partial \Phi}{\partial \boldsymbol{\sigma}}\right) = \mathrm{d}\left(\frac{3}{2}\frac{\boldsymbol{\sigma}_{\mathrm{dev}}}{\sigma_e}\right) &= \frac{3}{2\sigma_e}\,\mathrm{d}\boldsymbol{\sigma}_{\mathrm{dev}} + \frac{3}{2}\boldsymbol{\sigma}_{\mathrm{dev}}\,\mathrm{d}\left(\frac{1}{\sigma_e}\right) \\
&= \frac{3}{2\sigma_e}\,\mathrm{d}\boldsymbol{\sigma}_{\mathrm{dev}} - \frac{3}{2\sigma_e^2}\boldsymbol{\sigma}_{\mathrm{dev}}\,\mathrm{d}\sigma_e \\
&= \frac{3}{2\sigma_e}\frac{\partial \boldsymbol{\sigma}_{\mathrm{dev}}}{\partial \boldsymbol{\sigma}} : \mathrm{d}\boldsymbol{\sigma} - \frac{3}{2\sigma_e^2}\boldsymbol{\sigma}_{\mathrm{dev}} \otimes \frac{\partial \sigma_e}{\partial \boldsymbol{\sigma}_{\mathrm{dev}}} : \frac{\partial \boldsymbol{\sigma}_{\mathrm{dev}}}{\partial \boldsymbol{\sigma}} : \mathrm{d}\boldsymbol{\sigma} \\
&= \frac{3}{2\sigma_e}\mathbf{I}^{\mathrm{sym}}_{\mathrm{dev}} : \mathrm{d}\boldsymbol{\sigma} - \frac{3}{2\sigma_e^2}\boldsymbol{\sigma}_{\mathrm{dev}} \otimes \frac{3}{2}\frac{\boldsymbol{\sigma}_{\mathrm{dev}}}{\sigma_e} : \mathbf{I}^{\mathrm{sym}}_{\mathrm{dev}} : \mathrm{d}\boldsymbol{\sigma} \\
&= \left(\frac{3}{2\sigma_e}\mathbf{I}^{\mathrm{sym}}_{\mathrm{dev}} - \frac{9}{4\sigma_e^3}\boldsymbol{\sigma}_{\mathrm{dev}} \otimes \boldsymbol{\sigma}_{\mathrm{dev}}\right) : \mathrm{d}\boldsymbol{\sigma}
\end{aligned}$$

This result and using the chain rule leads to

$$\frac{\partial^2 \Phi}{\partial \boldsymbol{\sigma} \otimes \partial \boldsymbol{\sigma}} = \frac{3}{2\sigma_e}\mathbf{I}^{\mathrm{sym}}_{\mathrm{dev}} - \frac{9}{4\sigma_e^3}\boldsymbol{\sigma}_{\mathrm{dev}} \otimes \boldsymbol{\sigma}_{\mathrm{dev}}$$

Inserting the result from $\frac{\partial \Phi}{\partial \boldsymbol{\sigma}}$ and $\frac{\partial^2 \Phi}{\partial \boldsymbol{\sigma} \otimes \partial \boldsymbol{\sigma}}$ for von Mises yield criterion into Equation A.1

$$\begin{aligned}
\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} &= \frac{1}{t^*}\frac{\partial \eta[\Phi]}{\partial \Phi}\frac{3}{2}\frac{\boldsymbol{\sigma}_{\mathrm{dev}}}{\sigma_e} \otimes \frac{3}{2}\frac{\boldsymbol{\sigma}_{\mathrm{dev}}}{\sigma_e} + \frac{\eta[\Phi]}{t^*}\left(\frac{3}{2\sigma_e}\mathbf{I}^{\mathrm{sym}}_{\mathrm{dev}} - \frac{9}{4\sigma_e^3}\boldsymbol{\sigma}_{\mathrm{dev}} \otimes \boldsymbol{\sigma}_{\mathrm{dev}}\right) \\
&= \frac{3\eta[\Phi]}{2\sigma_e t^*}\mathbf{I}^{\mathrm{sym}}_{\mathrm{dev}} + \frac{9}{4t^*\sigma_e}\left(\frac{\partial \eta[\Phi]}{\partial \Phi}\sigma_e - \eta[\Phi]\right)\boldsymbol{\sigma}_{\mathrm{dev}} \otimes \boldsymbol{\sigma}_{\mathrm{dev}}
\end{aligned}$$

This result in that $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}}$ is then

$$\boxed{\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} = \frac{3\eta[\Phi]}{2\sigma_e t^*}\mathbf{I}^{\mathrm{sym}}_{\mathrm{dev}} + \frac{9}{4t^*\sigma_e}\left(\frac{\partial \eta[\Phi]}{\partial \Phi}\sigma_e - \eta[\Phi]\right)\boldsymbol{\sigma}_{\mathrm{dev}} \otimes \boldsymbol{\sigma}_{\mathrm{dev}}} \tag{A.2}$$

A.1.3 Norton model

The Norton model for the overstress function $\eta[\Phi]$ is

$$\eta[\Phi] = \left(\frac{\langle \Phi \rangle}{\sigma_c}\right)^{n_c} = \left(\frac{\langle \sigma_e - \sigma_y \rangle}{\sigma_c}\right)^{n_c} \geq 0$$

Then the derivative $\frac{\partial \eta[\Phi]}{\partial \Phi]}$ is

$$\frac{\partial \eta[\Phi]}{\partial \Phi} = n_c \frac{\langle \sigma_e - \sigma_y \rangle^{n_c - 1}}{\sigma_c^{n_c}}$$

Insert $\frac{\partial \eta[\Phi]}{\partial \Phi]}$ into Equation A.2

$$\begin{aligned}\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} &= \frac{3}{2\sigma_e t^*}\left(\frac{\langle \sigma_e - \sigma_y \rangle}{\sigma_c}\right)^{n_c} \mathbf{I}_{\mathrm{dev}}^{\mathrm{sym}} + \frac{9}{4t^*\sigma_e}\left(n_c \frac{\langle \sigma_e - \sigma_y \rangle^{n_c-1}}{\sigma_c^{n_c}}\sigma_e - \left(\frac{\langle \sigma_e - \sigma_y \rangle}{\sigma_c}\right)^{n_c}\right)\boldsymbol{\sigma}_{\mathrm{dev}} \otimes \boldsymbol{\sigma}_{\mathrm{dev}} \\ &= \frac{3(\sigma_e - \sigma_y)^{n_c}}{2t^*\sigma_e \sigma_c^{n_c}}\left[\mathbf{I}_{\mathrm{dev}}^{\mathrm{sym}} + \frac{3}{2\sigma_e^2}\left(\frac{n_c \sigma_e}{\sigma_e - \sigma_y} - 1\right)\boldsymbol{\sigma}_{\mathrm{dev}} \otimes \boldsymbol{\sigma}_{\mathrm{dev}}\right]\end{aligned}$$

This result in $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}}$ is

$$\boxed{\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} = \begin{cases} 0 & \sigma_e \leq \sigma_y \\ \dfrac{3(\sigma_e - \sigma_y)^{n_c}}{2t^*\sigma_e \sigma_c^{n_c}}\left[\mathbf{I}_{\mathrm{dev}}^{\mathrm{sym}} + \dfrac{3}{2\sigma_e^2}\left(\dfrac{n_c \sigma_e}{\sigma_e - \sigma_y} - 1\right)\boldsymbol{\sigma}_{\mathrm{dev}} \otimes \boldsymbol{\sigma}_{\mathrm{dev}}\right] & \sigma_e > \sigma_y \end{cases}} \tag{A.3}$$

A.2 Linearisation of weak residual

In this section the Gateaux derivative of residuals of the mixed weak form is derived.

The Gateaux derivatives of the residual R^{u} in the direction $\mathrm{d}\boldsymbol{u}$ is

$$\begin{aligned}R_u^{\mathrm{u}'}(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{u}, \mathrm{d}\boldsymbol{u}) &= a^u(\boldsymbol{u}; \delta\boldsymbol{u}, \mathrm{d}\boldsymbol{u})) - 0 - 0 \\ &= \int_\Omega \boldsymbol{\epsilon}[\delta\boldsymbol{u}] : \mathbf{E} : \left(\frac{\partial}{\partial \gamma}(\boldsymbol{\epsilon}[\boldsymbol{u} + \gamma \mathrm{d}\boldsymbol{u}])\right)\Big|_{\gamma=0} \mathrm{d}\Omega\end{aligned}$$

where the derivative with respect to γ is

$$\begin{aligned}\frac{\partial}{\partial \gamma}(\boldsymbol{\epsilon}[\boldsymbol{u} + \gamma \mathrm{d}\boldsymbol{u}]) &= \frac{\partial}{\partial \gamma}\left(\frac{1}{2}\Big(((\boldsymbol{u} + \gamma \mathrm{d}\boldsymbol{u}) \otimes \boldsymbol{\nabla}) + ((\boldsymbol{u} + \gamma \mathrm{d}\boldsymbol{u}) \otimes \boldsymbol{\nabla})^{\mathrm{T}}\Big)\right) \\ &= \frac{1}{2}\left(\left(\frac{\partial(\boldsymbol{u} + \gamma \mathrm{d}\boldsymbol{u})}{\partial \gamma} \otimes \boldsymbol{\nabla}\right) + \left(\frac{\partial(\boldsymbol{u} + \gamma \mathrm{d}\boldsymbol{u})}{\partial \gamma} \otimes \boldsymbol{\nabla}\right)^{\mathrm{T}}\right) \\ &= \frac{1}{2}\Big((\mathrm{d}\boldsymbol{u} \otimes \boldsymbol{\nabla}) + (\mathrm{d}\boldsymbol{u} \otimes \boldsymbol{\nabla})^{\mathrm{T}}\Big) \\ &= \boldsymbol{\epsilon}[\mathrm{d}\boldsymbol{u}]\end{aligned}$$

which leads to

$$\begin{aligned}R_u^{\mathrm{u}'}(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{u}, \mathrm{d}\boldsymbol{u}) &= \int_\Omega \boldsymbol{\epsilon}[\delta\boldsymbol{u}] : \mathbf{E} : \boldsymbol{\epsilon}[\mathrm{d}\boldsymbol{u}]\, \mathrm{d}\Omega \\ &= a^u(\delta\boldsymbol{u}, \mathrm{d}\boldsymbol{u}).\end{aligned}$$

The Gateaux derivative of residual R^{u} in the direction $\mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}$ is

$$\begin{aligned}
R^{\mathrm{u}'}_{\boldsymbol{\epsilon}^{\mathrm{vp}}}(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{u}, \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}) &= 0 - c(\boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{u}, \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}) - 0 \\
&= -\int_\Omega \boldsymbol{\epsilon}[\delta\boldsymbol{u}] : \mathbf{E} : \left(\frac{\partial(\boldsymbol{\epsilon}^{\mathrm{vp}} + \gamma \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}})}{\partial\gamma} \right)\Big|_{\gamma=0} \mathrm{d}\Omega \\
&= -\int_\Omega \boldsymbol{\epsilon}[\delta\boldsymbol{u}] : \mathbf{E} : \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}} \, \mathrm{d}\Omega \\
&= -c(\delta\boldsymbol{u}, \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}})
\end{aligned}$$

The Gateaux derivative of residual $R^{\boldsymbol{\epsilon}^{\mathrm{vp}}}$ in the direction $\mathrm{d}\boldsymbol{u}$ is

$$\begin{aligned}
R^{\boldsymbol{\epsilon}^{\mathrm{vp}\prime}}_{u}(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}}, \mathrm{d}\boldsymbol{u}) &= \frac{\partial}{\partial\gamma} R^{\boldsymbol{\epsilon}^{\mathrm{vp}}}(\boldsymbol{u} + \gamma\mathrm{d}\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}})\Big|_{\gamma=0} \\
&= 0 - 0 - \Delta t \frac{\partial}{\partial\gamma} \, b(\boldsymbol{u} + \gamma\mathrm{d}\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}})\Big|_{\gamma=0} \\
&= -\Delta t \frac{\partial}{\partial\gamma} \, b(\boldsymbol{u} + \gamma\mathrm{d}\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}})\Big|_{\gamma=0} \\
&= -\Delta t \frac{\partial}{\partial\gamma} \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \boldsymbol{g} \, \mathrm{d}\Omega \Big|_{\gamma=0} \\
&= -\Delta t \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} : \frac{\partial \boldsymbol{\sigma}(\boldsymbol{\epsilon}[\boldsymbol{u} + \gamma\mathrm{d}\boldsymbol{u}] - \boldsymbol{\epsilon}^{\mathrm{vp}})}{\partial\gamma}\Big|_{\gamma=0} \mathrm{d}\Omega
\end{aligned}$$

where $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}}$ is derived in Appendix A.1 for the used visco-plastic model. The derivative of $\boldsymbol{\sigma}$ is

$$\begin{aligned}
\frac{\partial \boldsymbol{\sigma}(\boldsymbol{\epsilon}[\boldsymbol{u} + \gamma\mathrm{d}\boldsymbol{u}] - \boldsymbol{\epsilon}^{\mathrm{vp}})}{\partial\gamma}\Big|_{\gamma=0} &= \frac{\partial}{\partial\gamma}(\mathbf{E} : (\boldsymbol{\epsilon}[\boldsymbol{u} + \gamma\mathrm{d}\boldsymbol{u}] - \boldsymbol{\epsilon}^{\mathrm{vp}}))\Big|_{\gamma=0} \\
&= \mathbf{E} : \boldsymbol{\epsilon}[\mathrm{d}\boldsymbol{u}].
\end{aligned}$$

This means that the Gateaux derivative in $\mathrm{d}\boldsymbol{u}$ direction is

$$R^{\boldsymbol{\epsilon}^{\mathrm{vp}\prime}}_{u}(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}}, \mathrm{d}\boldsymbol{u}) = -\Delta t \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} : \mathbf{E} : \boldsymbol{\epsilon}[\mathrm{d}\boldsymbol{u}] \, \mathrm{d}\Omega$$

The Gateaux derivative of residual $R^{\boldsymbol{\epsilon}^{\mathrm{vp}}}$ in the direction $\mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}$

$$\begin{aligned}
R^{\boldsymbol{\epsilon}^{\mathrm{vp}\prime}}_{\boldsymbol{\epsilon}^{\mathrm{vp}}}(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}}, \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}) &= \frac{\partial}{\partial\gamma} R^{\boldsymbol{\epsilon}^{\mathrm{vp}}}(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}} + \gamma\mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}})\Big|_{\gamma=0} \\
&= \frac{\partial}{\partial\gamma} \left(a^{\mathrm{vp}}(\boldsymbol{\epsilon}^{\mathrm{vp}} + \gamma\mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}}) - a^{\mathrm{vp}}\left({}^{(n)}\boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}} \right) - \Delta t \, b(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}} + \gamma\mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}}) \right)\Big|_{\gamma=0} \\
&= \frac{\partial}{\partial\gamma} a^{\mathrm{vp}}(\boldsymbol{\epsilon}^{\mathrm{vp}} + \gamma\mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}})\Big|_{\gamma=0} - 0 - \Delta t \, \frac{\partial}{\partial\gamma} b(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}} + \gamma\mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}})\Big|_{\gamma=0}
\end{aligned}$$

where it is obvious that

$$\begin{aligned}
\frac{\partial}{\partial\gamma} a^{\mathrm{vp}}(\boldsymbol{\epsilon}^{\mathrm{vp}} + \gamma\mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}})\Big|_{\gamma=0} &= \frac{\partial}{\partial\gamma} a^{\mathrm{vp}}(\boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}})\Big|_{\gamma=0} + \frac{\partial}{\partial\gamma} \gamma a^{\mathrm{vp}}(\mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}})\Big|_{\gamma=0} \\
&= a^{\mathrm{vp}}(\mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}}) \\
&= \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}} \, \mathrm{d}\Omega
\end{aligned}$$

and for b it is similar as in the $\mathrm{d}\boldsymbol{u}$ direction

$$\begin{aligned}\frac{\partial}{\partial\gamma} b(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}} + \gamma \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}})\Big|_{\gamma=0} &= \frac{\partial}{\partial\gamma} \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \boldsymbol{g}\, \mathrm{d}\Omega \Big|_{\gamma=0} \\ &= \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} : \frac{\partial \boldsymbol{\sigma}(\boldsymbol{\epsilon}[\boldsymbol{u}] - \boldsymbol{\epsilon}^{\mathrm{vp}} - \gamma \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}})}{\partial\gamma}\Big|_{\gamma=0} \mathrm{d}\Omega \\ &= -\int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} : \mathbf{E} : \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}\, \mathrm{d}\Omega\,.\end{aligned}$$

where $\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}}$ is derived in Appendix A.1 for the used visco-plastic model.

The Gateaux derivative in $\mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}$ direction is thus

$$R^{\epsilon^{\mathrm{vp}\prime}}_{\boldsymbol{\epsilon}^{\mathrm{vp}}}(\boldsymbol{u}, \boldsymbol{\epsilon}^{\mathrm{vp}}; \delta\boldsymbol{\epsilon}^{\mathrm{vp}}, \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}) = \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}\, \mathrm{d}\Omega + \Delta t \int_\Omega \delta\boldsymbol{\epsilon}^{\mathrm{vp}} : \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\sigma}} : \mathbf{E} : \mathrm{d}\boldsymbol{\epsilon}^{\mathrm{vp}}\, \mathrm{d}\Omega$$

DEPARTMENT OF INDUSTRIAL AND MATERIALS SCIENCE
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY