CHALMERS

Phase field modeling of damage using proper generalized decomposition

Master's thesis in Applied Mechanics

FREDRIK EKRE

Department of Applied Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2016

MASTER'S THESIS IN APPLIED MECHANICS

Phase field modeling of damage using proper generalized decomposition

FREDRIK EKRE

Department of Applied Mechanics
Material and Computational Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2016

Phase field modeling of damage using proper generalized decomposition
FREDRIK EKRE

Master's thesis 2016:37
ISSN 1652-8557
Department of Applied Mechanics
Material and Computational Mechanics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Chalmers Reproservice
Göteborg, Sweden 2016

Phase field modeling of damage using proper generalized decomposition
Master's thesis in Applied Mechanics
FREDRIK EKRE
Department of Applied Mechanics
Material and Computational Mechanics
Chalmers University of Technology

Abstract

The aim of this thesis is to investigate the model order reduction technique *proper generalized decomposition* (PGD). In particular, phase field modeling of damage will be considered. The general idea of PGD is to approximate the unknown field as a finite sum of modes. These modes are typically constructed in a multiplicative fashion based on functions of lower dimensionality, compared to the original function. The main advantage, from this decomposition, is that the modes can be discretized using much fewer degrees of freedom compared to a standard grid based method, such as the finite element method (FEM). It is, therefore, of interest to investigate this approach as a mean to reduce the computational cost.

There are a number of difficulties arising from a decomposition like this. Since the approximation is non-linear in the unknowns, the resulting equations will be non-linear (even if the governing differential equation is linear). The most common method to solve the non-linear PGD problem is to use a fix point numerical scheme, which proved to be best suited in this study as well. Boundary conditions and material parameters are some other difficulties that are discussed in the PGD setting.

In this thesis the PGD method is applied to a two field structural problem involving displacements and a scalar phase field, in order to simulate damage evolution. The numerical examples studied shows good agreement with a corresponding FEM solution (used as a reference solution), especially for the elasticity problem. In contrast, for the coupled problem with both displacements and damage as unknowns convergence issues were encountered and calls for further study.

The PGD method proved to be a useful tool to some extent; however, due to the additional modeling complexities introduces by the method and convergence issues encountered, its suitability for modeling progressive failure is currently uncertain.

Keywords: Proper generalized decomposition (PGD), Model reduction, Phase field damage

Preface

This thesis is the final piece of my Master of Science education in Applied Mechanics. The work was performed during the spring of 2016, during my fifth and last year of the basic education. Fortunately I have been given the opportunity to continue my journey in the academic jungle as a PhD student at Chalmers for another five years, which I am very excited about!

I want to thank my supervisor Jim Brouzoulis who always have been helpful and willing to discuss any problem I have encountered. I also want to thank Ronja for her love, support and encouraging words.

Fredrik Ekre, Gothenburg, 2016.

Contents

1 Introduction

In the field of computational fracture mechanics (CFM) there are a number of different approaches to simulate fracture and crack growth in materials. One such method is phase field modeling, where a scalar field, $d = d(\boldsymbol{x})$, is introduced to describe the state of the material. Some advantages with this method is that crack branching is automatically accounted for and it is not necessary to track the crack front. There is also no need to introduce a displacement discontinuity to describe the crack. Instead the phase field models a diffuse crack, smeared out over a small length scale, l. This means that the damage of the material is described as a diffuse transition, from fully intact to fully damaged, within this length scale. When the length scale approaches zero the smeared crack will result in a discrete crack. The length scale needs to be resolved when solving the equations and, when modeling discrete cracks and localized damage which requires a very fine mesh[1] the result is a large system of equations. It is therefore of interest to investigate alternative solution methods with better computational efficiency.

One potential way to increase computational efficiency is to use *proper generalized decomposition* (PGD). PGD is a solution method which decomposes the unknown function(s) as a sum of products of simpler *component* functions. For instance, an arbitrary scalar function $f = f(\boldsymbol{x})$ can be decomposed as

$$f(\boldsymbol{x}) = \sum_{i=1}^{M} F_x^i(x) \cdot F_y^i(y) \cdot F_z^i(z) \tag{1.1}$$

where M is the number of modes in the approximation and F_x^i, F_y^i, F_z^i are the three component functions for mode i. Since the components of the decomposed function are only dependent on one variable they are one dimensional. In a finite element setting, this means that they can be discretized using a one dimensional mesh for each function. By using such a decomposition the number of unknowns scales linearly with the number of dimensions, instead of exponentially like in grid based approximations. In Figure 1.1 a domain Ω is discretized with a conventional grid (used for e.g. finite elements) and a PGD grid. The conventional grid is three dimensional, while the PGD grid is constructed of three one dimensional grids. Note that the original unknown was only one function and by using the PGD approximation in Eq. (1.1) there are instead $3M$ (3 components $\times$ M modes) unknown functions. These unknown functions are simpler in the sense that they are defined over reduced domains.

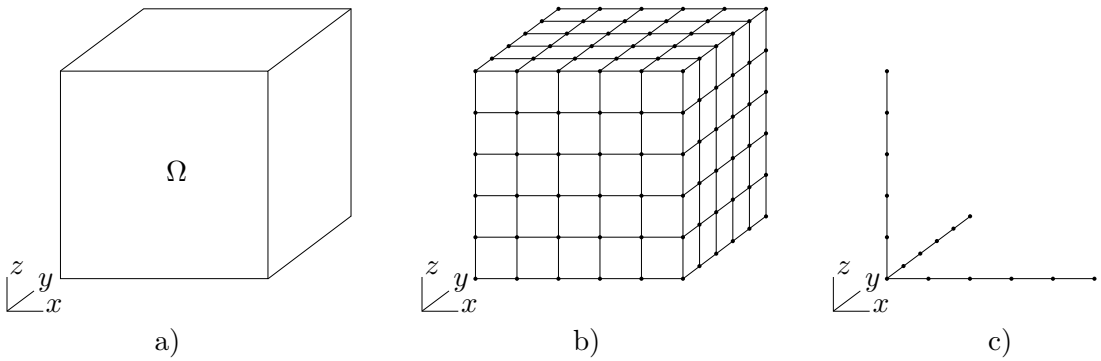


Figure 1.1: *Illustration of the different types of discretization; a) computational domain, b) grid based discretization of* Ω *with* 6^3 *nodes (*5^3 *3D elements) and c) PGD discretization of* Ω *with* 3×6 *nodes (*3×5 *1D elements).*

[1] Assuming the finite element method is used to solve the system, which is a common approach.

The PGD framework is relatively new and still under development and there are a number of inherent difficulties related to such a decomposition. However, the method has successfully been applied to a number of different problems. In [8] the method is described in detail, using the Poisson equation as an illustrative example providing accurate results compared to analytical solutions. In [14] PGD is used to analyze composite beams to resolve the transverse direction in a computationally efficient way. In [5] PGD is applied to shell and plate geometries by separating the in-plane coordinates (x, y) from the out of plane coordinate (z). Such a decomposition results in a problem with two dimensional complexity, without neglecting the dependence of the out of plane direction. Another interesting aspect of the PGD is to include the variation of some parameters as unknowns in the original problem. The variation of parameters was for example studied in [5], where the angle of selected plies in a composite were used as unknowns, and introduced as extra coordinates of the problem. The effect is that the solution to the problem is obtained simultaneously for many different angles. In [11] this property is used to optimize material parameters for a one dimensional heat transfer problem.

Since phase field modeling of fracture generally leads to very fine meshes, PGD can be a way to reduce the system to a smaller one. The purpose of this thesis is to investigate the possibility to combine PGD with phase field modeling of damage. In the following chapters the theory behind phase field modeling will be introduced, and how the PGD approximation of the unknown fields are set up. Following this, a number of example cases are investigated and in the end the results are discussed and some suggestions of further work is presented.

2 Phase field modeling of damage

To simulate damage and fracture in the material a phase field, d, is introduced as a scalar damage field with $d(\boldsymbol{x}) \in [0,1]$. This parameter describes the degradation of the stress in the material; $d = 0$ for the undamaged state and $d = 1$ for the fully damaged state. The introduction of the scalar field will lead to a coupled, two-field, problem with the displacement field, $\boldsymbol{u}(\boldsymbol{x},t)$, and the damage field, $d(\boldsymbol{x},t)$, as unknowns. There are different formulations of phase field damage, see for example [1]. Many are based on the minimization of an energy functional, for which the elastic energy and crack surface are minimized in the domain. This idea, with corresponding functional, was introduced in [9] for discrete cracks. A regularization of the same energy functional was introduced in [7] where the crack is described as a field parameter, which smoothly varies between the damaged and undamaged state, over a length scale l.

The driving force of the damage field is an energy field, $\mathcal{H}$, which are dependent on the displacements from the equations of equilibrium. In return, the stress in the equations of equilibrium are degraded based on the current damage field. Most formulations of phase field damage are similar, and the main differences between the models are the definitions for the energy field and stress degradation. One basic formulation is to simply degrade the full stress tensor, and use the total stored elastic energy as driving force, i.e.

$$\begin{cases} \boldsymbol{\sigma} = (1-d)^2 \, \dfrac{\partial \, \Psi}{\partial \boldsymbol{\varepsilon}} = (1-d)^2 \, \mathbf{E} : \boldsymbol{\varepsilon} \\ \mathcal{H} = \Psi(\boldsymbol{u}) \end{cases} \tag{2.1}$$

where $\Psi = \frac{1}{2}\boldsymbol{\varepsilon} : \mathbf{E} : \boldsymbol{\varepsilon}$ is the stored elastic energy and $\mathbf{E}$ the fourth order elasticity tensor. There are two main problems with this formulation; *a)* the process is reversible such that a crack can heal *b)* a crack will develop even if the material is loaded in compression. To stop cracks from healing, the driving energy $\mathcal{H}$ can be replaced by a history field of the energy, defined for example as

$$\mathcal{H} = \mathcal{H}(t) = \max_{\tau \in [0,t]} \Psi(\tau) \tag{2.2}$$

where t is the current time. With this definition the driving force cannot decrease, and thus a developed crack will not heal. The formulation in Eq. (2.1) together with the history field in Eq. (2.2) is called an isotropic formulation since it does not distinguish between compression and tension in the material and that all stress components are degraded in the same way. In [13] another formulation is proposed, which accounts for the loading type by splitting the elastic energy into a positive and negative part. With this formulation, only the tensile stress is degraded as follows

$$\begin{cases} \boldsymbol{\sigma} = (1-d)^2 \, \dfrac{\partial \, \Psi^+}{\partial \boldsymbol{\varepsilon}} + \dfrac{\partial \, \Psi^-}{\partial \boldsymbol{\varepsilon}} = (1-d)^2 \, \boldsymbol{\sigma}^+ + \boldsymbol{\sigma}^- \\ \mathcal{H} = \displaystyle\max_{\tau \in [0,t]} \Psi^+(\tau) \end{cases} \tag{2.3}$$

where the positive and negative part of the energy is defined using a spectral decomposition of the strain tensor

$$\Psi^{\pm} = \frac{1}{2}\boldsymbol{\varepsilon}^{\pm} : \mathbf{E} : \boldsymbol{\varepsilon}^{\pm} \qquad \boldsymbol{\varepsilon}^{\pm} = \sum_{i=1}^{3} < \varepsilon^i >_{\pm} \boldsymbol{n}^i \otimes \boldsymbol{n}^i \tag{2.4}$$

where $< \varepsilon^i >_{\pm}$ is the positive/negative part of the eigenvalues of the strain tensor, and $\boldsymbol{n}^i$ are the eigendirections of the strain. By adopting this model damage will be prevented from developing due to compressive loading. A similar decomposition of the energy Ψ is proposed in [4] where the decomposition of the energy is based on the volumetric and deviatoric part of the strain tensor.

In this thesis the simpler, isotropic, formulation is used, where the stress is defined according to Eq. (2.1) together with the history field in Eq. (2.2). The governing equations which are used to obtain the displacement and damage fields are presented in more detail below.

2.1 Weak form of equations of equilibrium

For the displacement field, $\boldsymbol{u}$, the equilibrium equations are solved. For a domain Ω with Neumann boundary Γ_{N} and Dirichlet boundary Γ_{D} the strong form of equilibrium can be stated as

$$-\boldsymbol{\sigma}(\boldsymbol{u}) \cdot \boldsymbol{\nabla} = \boldsymbol{f} \quad \text{in } \Omega \tag{2.5}$$

$$\boldsymbol{t} = \hat{\boldsymbol{t}} \quad \text{on } \Gamma_{\mathrm{N}} \tag{2.6}$$

$$\boldsymbol{u} = \hat{\boldsymbol{u}} \quad \text{on } \Gamma_{\mathrm{D}} \tag{2.7}$$

where $\boldsymbol{\sigma}$ is the stress tensor, $\boldsymbol{f}$ the body force and $\hat{\boldsymbol{t}}$ and $\hat{\boldsymbol{u}}$ are prescribed values for the traction and displacement on respective boundary, Γ_{N} and Γ_{D}. The solution to the equations of equilibrium is found from the corresponding weak form, i.e. find $\boldsymbol{u} \in \mathbb{U}_{\boldsymbol{u}}$ such that

$$\int_\Omega [\boldsymbol{v}_{\boldsymbol{u}} \otimes \boldsymbol{\nabla}] : \boldsymbol{\sigma} \, \mathrm{d}\Omega = \int_\Omega \boldsymbol{v}_{\boldsymbol{u}} \cdot \boldsymbol{f} \, \mathrm{d}\Omega + \int_{\Gamma_{\mathrm{N}}} \boldsymbol{v}_{\boldsymbol{u}} \cdot \hat{\boldsymbol{t}} \, \mathrm{d}\Gamma \quad \forall v_{\boldsymbol{u}} \in \mathbb{V}_{\boldsymbol{u}} \tag{2.8}$$

$$\mathbb{U}_{\boldsymbol{u}} = \left\{ \boldsymbol{v} \mid \boldsymbol{v} \in \mathbb{H}^1(\Omega), \boldsymbol{v} = \hat{\boldsymbol{u}} \text{ on } \Gamma_{\mathrm{D}} \right\} \tag{2.9}$$

$$\mathbb{V}_{\boldsymbol{u}} = \left\{ \boldsymbol{v} \mid \boldsymbol{v} \in \mathbb{H}^1(\Omega), \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma_{\mathrm{D}} \right\} \tag{2.10}$$

where $\boldsymbol{v}_{\boldsymbol{u}}$ is a test function and $\mathbb{H}^1(\Omega)$ is the Sobolov space[1] with functions whose gradients are square integrable in the domain Ω. For brevity, the weak form of the equilibrium equation will be written as

$$a_{\boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{v}_{\boldsymbol{u}}) = \ell_{\boldsymbol{u}}(\boldsymbol{v}_{\boldsymbol{u}}) \quad \forall \boldsymbol{v}_{\boldsymbol{u}} \in \mathbb{V}_{\boldsymbol{u}} \tag{2.11}$$

where the functionals $a_{\boldsymbol{u}}(\bullet, \bullet)$ and $\ell_{\boldsymbol{u}}(\bullet)$ are introduced for the internal and external forces respectively, defined as

$$a_{\boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{v}_{\boldsymbol{u}}) \stackrel{\text{def}}{=} \int_\Omega [\boldsymbol{v}_{\boldsymbol{u}} \otimes \boldsymbol{\nabla}] : \boldsymbol{\sigma} \, \mathrm{d}\Omega \tag{2.12}$$

$$\ell_{\boldsymbol{u}}(\boldsymbol{v}_{\boldsymbol{u}}) \stackrel{\text{def}}{=} \int_\Omega \boldsymbol{v}_{\boldsymbol{u}} \cdot \boldsymbol{f} \, \mathrm{d}\Omega + \int_{\Gamma_{\mathrm{N}}} \boldsymbol{v}_{\boldsymbol{u}} \cdot \hat{\boldsymbol{t}} \, \mathrm{d}\Gamma \tag{2.13}$$

The equations of equilibrium is coupled to the damage field via the stress tensor. The effective stress in the material, $\boldsymbol{\sigma}$, is expressed as

$$\boldsymbol{\sigma} = (1-d)^2 \, \hat{\boldsymbol{\sigma}} = (1-d)^2 \, \mathbf{E} : \boldsymbol{\varepsilon} \tag{2.14}$$

where d is the damage field and $\hat{\boldsymbol{\sigma}}$ the linear elastic stress tensor, with $\mathbf{E}$ being the fourth order elasticity tensor and $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}(\boldsymbol{u})$ the engineering strain tensor. Here it is clear that the damage field degrades the elastic stress in the material; when $d = 1$ the material is fully damaged and the resulting effective stress will be zero.

[1] $\mathbb{H}^1 = \left\{ \boldsymbol{v} \quad \mid \int_\Omega |\boldsymbol{v}|^2 + |\boldsymbol{v} \otimes \boldsymbol{\nabla}|^2 \, \mathrm{d}\Omega < \infty \right\}$

2.2 Weak form of damage evolution

The equations used to model phase field damage originates from the minimization of the fracture area, cf. [13]. In this thesis the governing equations for the damage field, d, is stated as

$$-l^2\Delta d + d = \frac{2l}{G_c}(1-d)\mathcal{H} \quad \text{in } \Omega \tag{2.15}$$

$$\boldsymbol{\nabla} d \cdot \boldsymbol{n} = 0 \quad \text{on } \Gamma \tag{2.16}$$

$$d = \hat{d} = 1 \quad \text{in } \Lambda \tag{2.17}$$

where l [m] is a length scale, G_c $[\frac{\text{J}}{\text{m}^2}]$ the fracture energy, $\mathcal{H}$ $[\frac{\text{J}}{\text{m}^3}]$ the energy history field, Γ the outer boundary of Ω and Λ is a (possible) initial crack surface within Ω. The damage field, d, is solved from the weak form corresponding to Eq. (2.15), i.e. find $d \in \mathbb{U}_d$ such that

$$\int_\Omega l^2 \ \boldsymbol{\nabla} v_d \cdot \boldsymbol{\nabla} d + v_d d \,\mathrm{d}\Omega = \int_\Omega v_d \frac{2l}{G_c}(1-d)\mathcal{H} \,\mathrm{d}\Omega \quad \forall v_d \in \mathbb{V}_d \tag{2.18}$$

$$\mathbb{U}_d = \left\{ v \mid v \in \mathbb{H}^1(\Omega), v = \hat{d} \text{ on } \Lambda \right\} \tag{2.19}$$

$$\mathbb{V}_d = \left\{ v \mid v \in \mathbb{H}^1(\Omega), v = 0 \text{ on } \Lambda \right\} \tag{2.20}$$

where v_d is a scalar test function. Here it is assumed that the gradient of the damage field in the normal direction of the boundary Γ is zero, which cancels the boundary integral from the partial integration[2]. In Eq. (2.18) it is clear that the energy field $\mathcal{H}$ will be the driving force for damage evolution, and since $\mathcal{H}$ is defined in such a way that it will not decrease, the damage will not decrease either. For convenience Eq. (2.18) is rewritten with all terms involving the unknown d gathered on the left hand side

$$\int_\Omega l^2 \ \boldsymbol{\nabla} v_d \cdot \boldsymbol{\nabla} d + \big(1 + \frac{2l}{G_c}\mathcal{H}\big) v_d d \,\mathrm{d}\Omega = \int_\Omega v_d \frac{2l}{G_c}\mathcal{H} \,\mathrm{d}\Omega \quad \forall v_d \in \mathbb{V}_d \tag{2.21}$$

$$\mathbb{U}_d = \left\{ v \mid v \in \mathbb{H}^1(\Omega), v = \hat{d} \text{ on } \Lambda \right\} \tag{2.22}$$

$$\mathbb{V}_d = \left\{ v \mid v \in \mathbb{H}^1(\Omega), v = 0 \text{ on } \Lambda \right\} \tag{2.23}$$

The internal boundary, Λ, describing an initial crack within the domain may be modeled by prescribing the damage parameter to a fully damaged state on Λ i.e. $d = \hat{d} = 1$. An initial crack may also be achieved by prescribing the history field, $\mathcal{H}$, along Λ, as suggested in [6]. Thus, the weak form in Eq. (2.21) can be reformulated as

$$\int_\Omega l^2 \ \boldsymbol{\nabla} v_d \cdot \boldsymbol{\nabla} d + \big(1 + \frac{2l}{G_c}\mathcal{H}\big) v_d d \,\mathrm{d}\Omega = \int_\Omega v_d \frac{2l}{G_c}\mathcal{H} \,\mathrm{d}\Omega \quad \forall v_d \in \mathbb{V}_d \tag{2.24}$$

$$\mathcal{H} = \hat{\mathcal{H}} \text{ on } \Lambda \tag{2.25}$$

$$\mathbb{U}_d = \mathbb{V}_d = \left\{ v \mid v \in \mathbb{H}^1(\Omega) \right\} \tag{2.26}$$

For brevity the weak form of the damage evolution is written as

$$a_d(d, v_d) = \ell_d(v_d) \quad \forall v_d \in \mathbb{V}_d \tag{2.27}$$

with the two functionals $a_d(d, v_d)$ and $\ell_d(v_d)$ defined as

[2] Cf. the traction term in the weak form of equilibrium, Eq. (2.8)

$$a_d(d, v_d) \stackrel{\text{def}}{=} \int_\Omega l^2 \, \boldsymbol{\nabla} v_d \cdot \boldsymbol{\nabla} d + \big(1 + \frac{2l}{G_c}\mathcal{H}\big) v_d d \, \mathrm{d}\Omega \tag{2.28}$$

$$\ell_d(v_d) \stackrel{\text{def}}{=} \int_\Omega v_d \frac{2l}{G_c}\mathcal{H} \, \mathrm{d}\Omega \tag{2.29}$$

2.3 Solution method

The coupled problem can be summarized as; find $\boldsymbol{u} \in \mathbb{U}_{\boldsymbol{u}}$ and $d \in \mathbb{U}_d$ such that

$$a_{\boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{v}_{\boldsymbol{u}}) = \ell_{\boldsymbol{u}}(\boldsymbol{v}_{\boldsymbol{u}}) \quad \forall \boldsymbol{v}_{\boldsymbol{u}} \in \mathbb{V}_{\boldsymbol{u}} \tag{2.30}$$

$$a_d(d, v_d) = \ell_d(v_d) \quad \forall v_d \in \mathbb{V}_d \tag{2.31}$$

where the functionals are defined in the previous sections. This is a set of coupled non-linear equations which needs to be solved to obtain the displacement and damage fields. One solution method, which has proven to be more stable and robust compared to a monolithic solution, is presented in [13, 1, 4]. The method is called *staggered* solution scheme, since the two fields are solved individually, considering the other field as fix. For a given time step, the damage field is first solved, keeping the displacement field fixed. Next, the displacement field is solved, keeping the damage field fixed. The procedure is described by the four steps in Box 2.1, where steps 2-4 needs to be iterated for each time step, to reach a stationary point, since the two equations are coupled but solved in this decoupled fashion. Before the damage field starts to develop, typically only one iteration is needed to reach a stationary point. When the damage evolution starts, more iterations are needed, depending on the choice of damage model and history field, as shown in [1].

1. For a time step $n+1$ the old fields are known; ${}^n\boldsymbol{u}$, ${}^n d$ and ${}^n\mathcal{H}$. Update prescribed loads for the new time step $n+1$; ${}^{n+1}\hat{\boldsymbol{u}}$, ${}^{n+1}\hat{\boldsymbol{t}}$ and ${}^{n+1}\boldsymbol{f}$
2. Solve ${}^{n+1}d$ from Eq. (2.31) with known ${}^n\mathcal{H}$ where ${}^n\mathcal{H} = {}^n\mathcal{H}({}^n\boldsymbol{u})$
3. Solve ${}^{n+1}\boldsymbol{u}$ from Eq. (2.30) with the newly computed ${}^{n+1}d$
4. Update history field: ${}^{n+1}\mathcal{H} = {}^{n+1}\mathcal{H}({}^{n+1}\boldsymbol{u})$

Box 2.1: *Steps in the staggered solution scheme for solving the coupled system of equations with displacement and damage field as unknown.*

3 Proper Generalized Decomposition

Proper generalized decomposition (PGD) is a numerical model reduction method that has risen in popularity lately. Numerical model reduction methods are often appealing to increase computational efficiency, and especially for computationally heavy problems. The main features of the PGD method is outlined here, but a more extensive overview can be found in [8], which also discusses the relation between PGD and other numerical model reduction techniques. First a general description of PGD is given, and later also in a more specific context, related to solving the equations of equilibrium together with a damage field as presented in Section 2.1 and Section 2.2.

There are different ways to apply a PGD discretization scheme. The general idea is to approximate an unknown function defined in $\mathbb{R}^N$ as a finite sum of modes. Each mode is approximated as a product of *component functions* which, typically, are defined in a lower dimension e.g. $\mathbb{R}^M$ where $M < N$. This property is one of the main advantages of the method, and removes the so-called *curse of dimensionality*. As an example, the unknown scalar field f defined in $\mathbb{R}^N$ can be decomposed, in the most general form of PGD, as a sum of M modes

$$f(x_1, x_2, \ldots, x_N) \approx \sum_{i=1}^{M} F^i_{x_1}(x_1) \cdot F^i_{x_2}(x_2) \cdot \ldots \cdot F^i_{x_N}(x_N) \tag{3.1}$$

where, in this case, the component functions, $F^i_{x_k}(x_k)$, are functions of only one dimension, i.e. they are defined in $\mathbb{R}^1$. This means that the original problem, which is defined in $\mathbb{R}^N$ has been decomposed into N smaller problems defined in $\mathbb{R}^1$. As mentioned above, the number of modes and the component functions are unknown a priori, which results in $N \times M$ unknown component functions to approximate the original function. These functions are simpler in the sense that they are defined in a lower dimension than the original function. The unknowns are obtained by inserting the PGD approximation into the governing differential equations, see Section 3.1.

The decomposition of the unknown function can be performed in many different ways, depending on the application and governing equations. A problem defined in space and time, i.e. $f = f(\boldsymbol{x}, t)$, can, quite naturally, be decomposed as

$$f(\boldsymbol{x}, t) = \sum_{i=1}^{M} F^i_{\boldsymbol{x}}(\boldsymbol{x}) \cdot F^i_t(t) \tag{3.2}$$

where the spatial coordinates, $\boldsymbol{x}$, are separated from the time coordinate, t. Of course, the spatial domain could be decomposed as well, which results in four component functions per mode as

$$f(\boldsymbol{x}, t) = \sum_{i=1}^{M} F^i_x(x) \cdot F^i_y(y) \cdot F^i_z(z) \cdot F^i_t(t) \tag{3.3}$$

For plate and shell geometries it is common to keep the in-plane coordinates together (x, y) and separate the out of plane coordinate (z), see for instance [5, 12] for this type of decomposition.

For a vector valued function, the decomposition looks the same, but with vector valued component functions. For example, the vector valued function $\boldsymbol{f}(\boldsymbol{x}) = (f(\boldsymbol{x}),\ g(\boldsymbol{x}),\ h(\boldsymbol{x}))$ may be decomposed as

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{i=1}^{M} \boldsymbol{F}^i_x(x) \circ \boldsymbol{F}^i_y(y) \circ \boldsymbol{F}^i_z(z) \tag{3.4}$$

where $\circ$ denotes the Hadamard product for element wise multiplication, i.e.

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{i=1}^{M} \left(\begin{bmatrix} F_x^i(x) \\ G_x^i(x) \\ H_x^i(x) \end{bmatrix} \circ \begin{bmatrix} F_y^i(y) \\ G_y^i(y) \\ H_y^i(y) \end{bmatrix} \circ \begin{bmatrix} F_z^i(z) \\ G_z^i(z) \\ H_z^i(z) \end{bmatrix} \right) = \sum_{i=1}^{M} \begin{pmatrix} F_x^i(x) \cdot F_y^i(y) \cdot F_z^i(z) \\ G_x^i(x) \cdot G_y^i(y) \cdot G_z^i(z) \\ H_x^i(x) \cdot H_y^i(y) \cdot H_z^i(z) \end{pmatrix} \tag{3.5}$$

Henceforth, only the subscript is kept for brevity (on the component functions) to indicate its dependence, i.e. $F_x^i = F_x^i(x)$. We also introduce the notation $\tilde{\bullet}^i$ to denote mode i of the PGD approximation and the notation $\bullet^n$ to denote the sum of all modes up to n. With these notations, the approximation in Eq. (3.4) can be written in a more compact form as

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{i=1}^{M} \underbrace{\boldsymbol{F}_x^i \circ \boldsymbol{F}_y^i \circ \boldsymbol{F}_z^i}_{\stackrel{\text{def}}{=} \tilde{\boldsymbol{f}}^i} = \sum_{i=1}^{M} \tilde{\boldsymbol{f}}^i \stackrel{\text{def}}{=} \boldsymbol{f}^M \tag{3.6}$$

3.1 Enrichment process

The unknown component functions are computed, one mode at a time, in a so called *enrichment* process. In this section the steps to obtain the modes are described for a general case, using an arbitrary weak form (Eq. (3.7)) with a PGD approximated solution, f; a similar description can be found in [15].

We start out from a general weak form; find $f \in \mathbb{U}$ such that

$$A(f, v) = L(v) \quad \forall v \in \mathbb{V} \tag{3.7}$$

where $A(\bullet, \bullet)$ and $L(\bullet)$ are functionals originating from the governing differential equation, f is the unknown (scalar) solution to the problem, v is an appropriate test function and $\mathbb{U}$ and $\mathbb{V}$ corresponding function spaces. We assume a separable domain, Ω, see Figure 3.1 which can be separated as $\Omega = \Omega_x \times \Omega_y \times \Omega_z$. We also define the boundaries of the sub-domains as: Γ_x, Γ_y and Γ_z.

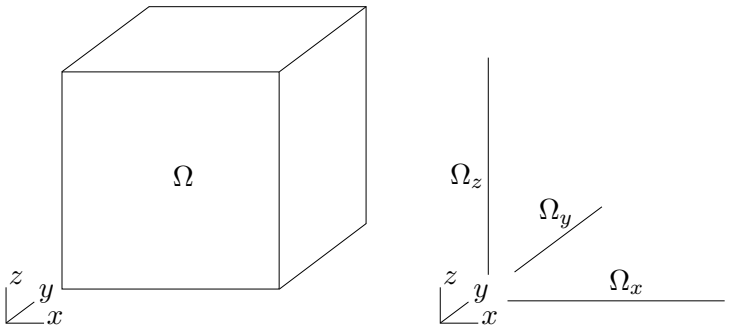


Figure 3.1: *Computational domain, Ω, to the left and the corresponding separated domain $\Omega = \Omega_x \times \Omega_y \times \Omega_z$, to the right.*

From the start, at enrichment step 1, no modes are known. After the first mode have been calculated the approximation can be enriched with a second mode and so on. This means that at enrichment step n there are $n-1$ modes which are already computed, and an n:th mode are sought

to enhance the solution. At this enrichment step the approximation thus reads

$$f^n(\boldsymbol{x}) = \sum_{i=1}^{n-1} \left(F_x^i \cdot F_y^i \cdot F_z^i\right) + F_x^n \cdot F_y^n \cdot F_z^n = f^{n-1} + \tilde{f}^n \tag{3.8}$$

where the three components F_x^n, F_y^n and F_z^n are the three unknown functions which will enhance the solution. They are obtained by solving the weak form (Eq. (3.7)).

For the weak form a test function v is also needed. The simplest, and most common, choice is to use the variation of the unknown function itself, i.e.

$$v = v^n = \delta(F_x^n \cdot F_y^n \cdot F_z^n) = \underbrace{\delta F_x^n \cdot F_y^n \cdot F_z^n}_{v_x} + \underbrace{F_x^n \cdot \delta F_y^n \cdot F_z^n}_{v_y} + \underbrace{F_x^n \cdot F_y^n \cdot \delta F_z^n}_{v_z} \tag{3.9}$$

Inserting this test function in Eq. (3.7) gives three equations (assuming $A(\bullet, \bullet)$ is bi-linear) from which the new mode can be calculated, i.e. find $\tilde{f}^n = F_x^n \cdot F_y^n \cdot F_z^n \in \mathbb{U}$ such that

$$A(\tilde{f}^n, v_x) = -A(f^{n-1}, v_x) + L(v_x) \qquad \forall v_x \in \mathbb{V}_x \tag{3.10}$$

$$A(\tilde{f}^n, v_y) = -A(f^{n-1}, v_y) + L(v_y) \qquad \forall v_y \in \mathbb{V}_y \tag{3.11}$$

$$A(\tilde{f}^n, v_z) = -A(f^{n-1}, v_z) + L(v_z) \qquad \forall v_z \in \mathbb{V}_z \tag{3.12}$$

where the function spaces are defined as

$$\mathbb{V}_\xi = \left\{v \mid \text{suff. reg., } v = 0 \text{ on } \Gamma_{\mathrm{D}\xi}\right\} \qquad \xi = x, y, z \tag{3.13}$$

Note that Eqs. (3.10)-(3.12) are non-linear, since a multiplicative decomposition is used. This is clearly one drawback associated with the PGD, and a suitable numerical method is needed to solve for the new modes, see Section 3.2.

The approximation is enriched with new modes until a chosen convergence criterion is fulfilled. Several criteria are mentioned in [8] to evaluate the influence of the new mode. The first criterion compares the contribution of the new mode, $\tilde{f}^n$, to the total solution, f^n

$$\epsilon(n) = \frac{\|\tilde{f}^n\|}{\|f^n\|}, \quad \epsilon(n) < \bar{\epsilon} \tag{3.14}$$

where $\| \bullet \|$ is a, for the problem, suitable norm and $\bar{\epsilon}$ a sufficiently small number. Another criterion from [8] compares the new mode only to the first one, assuming the first mode has the largest impact on the total solution

$$\epsilon(n) = \frac{\|\tilde{f}^n\|}{\|f^1\|}, \quad \epsilon(n) < \bar{\epsilon} \tag{3.15}$$

In [2] a criteria based on the amplitude of the modes is used. They also use a slightly different approach as they use the normalized modes as a basis, and re-calculate the corresponding amplitudes after each new mode/base is found. When the amplitude for the new mode is below a threshold the enhancement stops. Other criteria could of course also be used, and for instance based on some quantity of interest depending on the problem at hand.

3.2 Alternating direction strategy

To solve the non-linear equations, in Eqs. (3.10)-(3.12), a suitable numerical scheme is needed to obtain the components for the new mode. Newton's method has been used in some cases, see for

instance [2], where it successfully has been applied for a time and space dependent temperature problem. The most common method, however, is a so called *alternating directions strategy*[1]. This method has been shown to be more stable and more robust, see e.g. [3, 8], and is therefore used for almost all PGD applications. In this thesis the alternating directions strategy is also used, since some tests with Newton's method did not give desired results. The alternating directions strategy is an iterative process to solve for the unknown component functions. Using this method, all but one component of the unknown mode is considered to be known. This results in a linearization of the approximation, and the equations in Eqs. (3.10)-(3.12) will become linear. The steps to solve for the unknown are presented in Box 3.1. The steps will be iterated until the three modes have reached a stationary point, and Eqs. (3.10)-(3.12) are fulfilled. When the components of mode n are computed the next enrichment step can start, where mode $n+1$ is sought to enhance the solution.

Note that the linearization described above does not only linearize the first argument of $A(\bullet, \bullet)$, but also the second (the test function). For e.g. Eq. (3.16) the test function, $v_x = \delta F_x^n \cdot F_y^n \cdot F_z^n$, is linear, since F_y^n and F_z^n are considered known.

– Given F_y^n and F_z^n; update F_x^n such that

$$A(\tilde{f}^n, v_x) = -A(f^{n-1}, v_x) + L(v_x) \qquad \forall v_x \in \mathbb{V}_x \tag{3.16}$$

– Given F_x^n and F_z^n; update F_y^n such that

$$A(\tilde{f}^n, v_y) = -A(f^{n-1}, v_y) + L(v_y) \qquad \forall v_y \in \mathbb{V}_y \tag{3.17}$$

– Given F_x^n and F_y^n; update F_z^n such that

$$A(\tilde{f}^n, v_z) = -A(f^{n-1}, v_z) + L(v_z) \qquad \forall v_z \in \mathbb{V}_z \tag{3.18}$$

Box 3.1: *Steps of the alternating directions strategy to iterate for the components of the new mode for the example problem in Eq.* (3.7).

3.3 Aspects of integration and application of boundary conditions using PGD

Another advantage, apart from the reduced number of unknowns, with the PGD is that integrals involved in the weak form can be evaluated on smaller domains. To be able to take advantage of this property, the domain needs to be separable. To separate the spatial variables x, y and z the domain Ω needs to be the Cartesian product of the subspaces Ω_x, Ω_y and Ω_z, i.e. $\Omega = \Omega_x \times \Omega_y \times \Omega_z$ as illustrated in Figure 3.1. If this is the case the integrals can be evaluated in the three one-dimensional sub-domains, instead of the original three dimensional domain. Since the simpler integrals are much cheaper to evaluate, this is a property that can save computational time. For a non-separable domain the integrals can either be evaluated in the original domain, or by embedding the domain in a larger, separable, domain, $\bar{\Omega}$, such that $\Omega \subset \bar{\Omega}$. This procedure is illustrated for a non-separable, two dimensional, domain Ω in Figure 3.2. This method was introduced in [10] and shows promising results for some examples.

To be able to evaluate the integrals in the separated domains it is not enough that the unknown function and the original domain is separated. Other parameters involved in the weak form, such as

[1] Also called *fix-point* method.

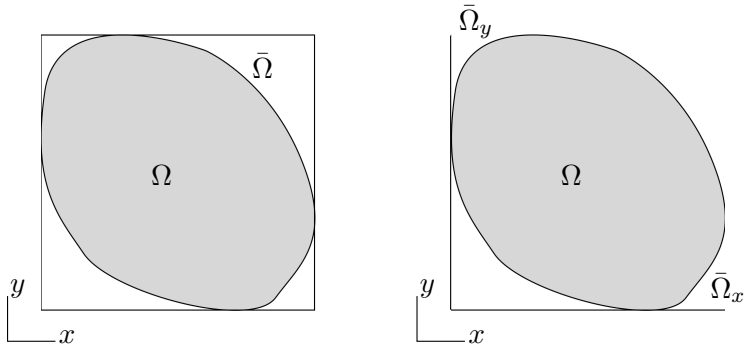


Figure 3.2: *Illustration of a non separable domain Ω embedded in the separable domain $\bar{\Omega}$ with the sub-domains $\bar{\Omega}_x$ and $\bar{\Omega}_y$ such that $\Omega \subset \bar{\Omega} = \bar{\Omega}_x \times \bar{\Omega}_y$.*

material parameters, must also be decomposed in the same way (or be independent of the decomposed domains). For instance, for an elasticity problem, the elastic stiffness tensor, $\mathbf{E} = \mathbf{E}(x, y, z)$, can be approximated with PGD as

$$\mathbf{E}(x, y, z) = \sum_{i=1}^{M} \mathbf{E}_x^i(x) \circ \mathbf{E}_y^i(y) \circ \mathbf{E}_z^i(z) \tag{3.19}$$

which can be done in a pre-processing step; however, if the stiffness changes with time this decomposition needs to be performed in each time step. With everything separated, the integrals can be evaluated in the smaller domain, which can save a lot of computational time.

Another difficulty with using PGD compared to FEM is applying boundary conditions. Neumann boundary conditions can normally be included as usual in the boundary integral of the weak form, and homogeneous Dirichlet boundary conditions can be dealt with by enforcing them on all of the modes. Non-homogeneous Dirichlet boundary conditions, however, can not be enforced on all modes since the approximation are constructed as the sum of all modes. To deal with this type of boundary condition, a first mode can be constructed as described in [10, 8]. The mode is constructed in such a way that it fulfills all Dirichlet boundary conditions (both homogeneous and non-homogeneous). The following modes are prescribed to zero on this part of the boundary, which can be expressed as

$$\begin{aligned} \tilde{f}^1 &= \hat{f} \quad \text{on } \Gamma_{\mathrm{D}} \\ \tilde{f}^i &= 0 \quad \text{on } \Gamma_{\mathrm{D}} \qquad i = 2, 3, \ldots, M \end{aligned} \tag{3.20}$$

where $\hat{f}$ is the prescribed value of the function f on Γ_{D}. For simple cases, such a first mode can easily be set up, but for complicated cases the first mode needs to be computed a priori, to fulfill the Dirichlet boundary conditions.

4 PGD formulation of governing equations

In this section the PGD method is applied to the governing equations presented in Section 2.1 and Section 2.2. For the equations of equilibrium the displacement field is decomposed in the spatial dimensions, and for the damage evolution the damage field is decomposed in the spatial dimensions, as well. The procedure is much the same for both problems, the main difference being the governing differential equation. The problem is first approximated using continuous modes, and then these modes are discretized using one-dimensional finite elements. The solution procedure to obtain the discrete modes are also presented.

4.1 PGD formulation of the equations of equilibrium

To solve the weak form of the equations of equilibrium, the unknown displacement field, $\boldsymbol{u}(\boldsymbol{x}) = \begin{bmatrix} u(\boldsymbol{x}) & v(\boldsymbol{x}) & w(\boldsymbol{x}) \end{bmatrix}^{\mathrm{T}}$, is approximated using PGD. This field is decomposed into each spatial dimension and approximated as the sum of M modes as

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_{i=1}^{M} \boldsymbol{U}_x^i(x) \circ \boldsymbol{U}_y^i(y) \circ \boldsymbol{U}_z^i(z) = \sum_{i=1}^{M} \tilde{\boldsymbol{u}}^i = \boldsymbol{u}^M \tag{4.1}$$

with the notation introduced earlier, cf. Section 3. Since the displacement field $\boldsymbol{u}$ is vector valued, we obtain

$$\boldsymbol{U}_\xi^i(\xi) = \begin{bmatrix} U_\xi^i(\xi) \\ V_\xi^i(\xi) \\ W_\xi^i(\xi) \end{bmatrix}, \qquad \xi = x, y, z \tag{4.2}$$

for each comonent ξ. Using the Hadamard product between the components, the approximation of the displacement field can be written explicitly as

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_{i=1}^{M} \left(\begin{bmatrix} U_x^i \\ V_x^i \\ W_x^i \end{bmatrix} \circ \begin{bmatrix} U_y^i \\ V_y^i \\ W_y^i \end{bmatrix} \circ \begin{bmatrix} U_z^i \\ V_z^i \\ W_z^i \end{bmatrix} \right) = \sum_{i=1}^{M} \begin{bmatrix} U_x^i \cdot U_y^i \cdot U_z^i \\ V_x^i \cdot V_y^i \cdot V_z^i \\ W_x^i \cdot W_y^i \cdot W_z^i \end{bmatrix} = \sum_{i=1}^{M} \underbrace{\begin{bmatrix} \tilde{u}^i \\ \tilde{v}^i \\ \tilde{w}^i \end{bmatrix}}_{\tilde{\boldsymbol{u}}^i} \tag{4.3}$$

Each of the modes are now approximated as a product of three component functions, which are functions of only one spatial variable; x, y or z. The fact that the components are one dimensional makes it possible to use a one dimensional discretization for each component, and thus fewer degrees of freedom are needed for the total approximation.

For the weak form of equilibrium, Eq. (2.8), the strain is needed. The strain is defined as the symmetric part of the gradient of the displacement field $\boldsymbol{u}$

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}(\boldsymbol{u}) = [\boldsymbol{u} \otimes \boldsymbol{\nabla}]^{\mathrm{sym}} = \frac{1}{2}[(\boldsymbol{u} \otimes \boldsymbol{\nabla}) + (\boldsymbol{\nabla} \otimes \boldsymbol{u})] \tag{4.4}$$

and by inserting the approximation from Eq. (4.1) the strain of the PGD approximated displacement

field is obtained

$$\boldsymbol{\varepsilon}^M(\boldsymbol{u}^M) = [\boldsymbol{u}^M \otimes \boldsymbol{\nabla}]^{\text{sym}} = \sum_{i=1}^{M} [\tilde{\boldsymbol{u}}^i \otimes \boldsymbol{\nabla}]^{\text{sym}} =$$

$$= \sum_{i=1}^{M} \begin{bmatrix} \partial_x \boldsymbol{U}_x^i \circ \boldsymbol{U}_y^i \circ \boldsymbol{U}_z^i & \boldsymbol{U}_x^i \circ \partial_y \boldsymbol{U}_y^i \circ \boldsymbol{U}_z^i & \boldsymbol{U}_x^i \circ \boldsymbol{U}_y^i \circ \partial_z \boldsymbol{U}_z^i \end{bmatrix}^{\text{sym}} = \tag{4.5}$$

$$= \sum_{i=1}^{M} \Big[\begin{bmatrix} \partial_x \boldsymbol{U}_x^i & \boldsymbol{U}_x^i & \boldsymbol{U}_x^i \end{bmatrix} \circ \begin{bmatrix} \boldsymbol{U}_y^i & \partial_y \boldsymbol{U}_y^i & \boldsymbol{U}_y^i \end{bmatrix} \circ \begin{bmatrix} \boldsymbol{U}_z^i & \boldsymbol{U}_z^i & \partial_z \boldsymbol{U}_z^i \end{bmatrix} \Big]^{\text{sym}}$$

Note that, due to the decomposition of the displacement field $\boldsymbol{u}$, the strain also is decomposed in each direction. The strain can explicitly be written as

$$\boldsymbol{\varepsilon}^M(\boldsymbol{u}^M) = \sum_{i=1}^{M} \begin{pmatrix} \partial_x U_x^i \cdot U_y^i \cdot U_z^i & \frac{1}{2}(U_x^i \cdot \partial_y U_y^i \cdot U_z^i + \partial_x V_x^i \cdot V_y^i \cdot V_z^i) & \frac{1}{2}(U_x^i \cdot U_y^i \cdot \partial_z U_z^i + \partial_x W_x^i \cdot W_y^i \cdot W_z^i) \\ & V_x^i \cdot \partial_y V_y^i \cdot V_z^i & \frac{1}{2}(V_x^i \cdot V_y^i \cdot \partial_z V_z^i + W_x^i \cdot \partial_y W_y^i \cdot W_z^i) \\ \text{Sym.} & & W_x^i \cdot W_y^i \cdot \partial_z W_z^i \end{pmatrix} \tag{4.6}$$

Following the solution method described in Section 3.1 and Section 3.2 it is convenient to split the strain in two parts; one known part corresponding to all modes up to $n-1$, and one unknown part for mode, n

$$\boldsymbol{\varepsilon}^n(\boldsymbol{u}^n) = \underbrace{\sum_{i=1}^{n-1} [\boldsymbol{u}^i \otimes \boldsymbol{\nabla}]^{\text{sym}}}_{\text{known}} + \underbrace{[\tilde{\boldsymbol{u}}^n \otimes \boldsymbol{\nabla}]^{\text{sym}}}_{\text{unknown}} = \boldsymbol{\varepsilon}^{n-1}(\boldsymbol{u}^{n-1}) + \boldsymbol{\varepsilon}^n(\tilde{\boldsymbol{u}}^n) \tag{4.7}$$

For the weak form, a test function must be chosen. As described in Section 3.1 it is taken as the variation of the unknown, n:th, mode

$$\boldsymbol{v}_{\boldsymbol{u}} = \delta(\boldsymbol{U}_x^n \circ \boldsymbol{U}_y^n \circ \boldsymbol{U}_z^n) = \underbrace{\delta \boldsymbol{U}_x^n \circ \boldsymbol{U}_y^n \circ \boldsymbol{U}_z^n}_{\boldsymbol{v}_{\boldsymbol{u}x}} + \underbrace{\boldsymbol{U}_x^n \circ \delta \boldsymbol{U}_y^n \circ \boldsymbol{U}_z^n}_{\boldsymbol{v}_{\boldsymbol{u}y}} + \underbrace{\boldsymbol{U}_x^n \circ \boldsymbol{U}_y^n \circ \delta \boldsymbol{U}_z^n}_{\boldsymbol{v}_{\boldsymbol{u}z}} = \boldsymbol{v}_{\boldsymbol{u}x} + \boldsymbol{v}_{\boldsymbol{u}y} + \boldsymbol{v}_{\boldsymbol{u}z} \tag{4.8}$$

The gradient of the test function is also needed for the weak form

$$\boldsymbol{v}_{\boldsymbol{u}} \otimes \boldsymbol{\nabla} = \boldsymbol{v}_{\boldsymbol{u}x} \otimes \boldsymbol{\nabla} + \boldsymbol{v}_{\boldsymbol{u}y} \otimes \boldsymbol{\nabla} + \boldsymbol{v}_{\boldsymbol{u}z} \otimes \boldsymbol{\nabla} \tag{4.9}$$

The weak form can now be formulated in a PGD setting, with the n:th mode, $\tilde{\boldsymbol{u}}^n$, as unknown

$$\underbrace{\int_\Omega [\boldsymbol{v}_{\boldsymbol{u}} \otimes \boldsymbol{\nabla}] : \tilde{\boldsymbol{\sigma}}^n \, \mathrm{d}\Omega}_{a_{\boldsymbol{u}}(\tilde{\boldsymbol{u}}^n, \boldsymbol{v}_{\boldsymbol{u}})} = -\underbrace{\int_\Omega [\boldsymbol{v}_{\boldsymbol{u}} \otimes \boldsymbol{\nabla}] : \boldsymbol{\sigma}^{n-1} \, \mathrm{d}\Omega}_{a_{\boldsymbol{u}}(\boldsymbol{u}^{n-1}, \boldsymbol{v}_{\boldsymbol{u}})} + \underbrace{\int_\Omega \boldsymbol{v}_{\boldsymbol{u}} \cdot \boldsymbol{f} \, \mathrm{d}\Omega + \int_{\Gamma_{\text{N}}} \boldsymbol{v}_{\boldsymbol{u}} \cdot \hat{\boldsymbol{t}} \, \mathrm{d}\Gamma}_{\ell_{\boldsymbol{u}}(\boldsymbol{v}_{\boldsymbol{u}})} \qquad \forall \boldsymbol{v}_{\boldsymbol{u}} \in \mathbb{V} \tag{4.10}$$

where $\tilde{\boldsymbol{\sigma}}^n = \tilde{\boldsymbol{\sigma}}^n(\tilde{\boldsymbol{u}}^n)$ is the stress due to the strain from mode n and $\boldsymbol{\sigma}^{n-1} = \boldsymbol{\sigma}^{n-1}(\boldsymbol{u}^{n-1})$ the stress due to the strain of all known modes. By inserting the expression for the test function from Eq. (4.8) and Eq. (4.9) three equations are obtained, from which the new mode can be solved, i.e.; find $\tilde{\boldsymbol{u}}^n = \boldsymbol{U}_x^n \circ \boldsymbol{U}_y^n \circ \boldsymbol{U}_z^n \in \mathbb{U}_{\boldsymbol{u}}$ such that

$$a_{\boldsymbol{u}}(\tilde{\boldsymbol{u}}^n, \boldsymbol{v}_{\boldsymbol{u}x}) = -a_{\boldsymbol{u}}(\boldsymbol{u}^{n-1}, \boldsymbol{v}_{\boldsymbol{u}x}) + \ell_{\boldsymbol{u}}(\boldsymbol{v}_{\boldsymbol{u}x}) \qquad \forall \boldsymbol{v}_{\boldsymbol{u}x} \in \mathbb{V}_{\boldsymbol{u}x} \tag{4.11}$$

$$a_{\boldsymbol{u}}(\tilde{\boldsymbol{u}}^n, \boldsymbol{v}_{\boldsymbol{u}y}) = -a_{\boldsymbol{u}}(\boldsymbol{u}^{n-1}, \boldsymbol{v}_{\boldsymbol{u}y}) + \ell_{\boldsymbol{u}}(\boldsymbol{v}_{\boldsymbol{u}y}) \qquad \forall \boldsymbol{v}_{\boldsymbol{u}y} \in \mathbb{V}_{\boldsymbol{u}y} \tag{4.12}$$

$$a_{\boldsymbol{u}}(\tilde{\boldsymbol{u}}^n, \boldsymbol{v}_{\boldsymbol{u}z}) = -a_{\boldsymbol{u}}(\boldsymbol{u}^{n-1}, \boldsymbol{v}_{\boldsymbol{u}z}) + \ell_{\boldsymbol{u}}(\boldsymbol{v}_{\boldsymbol{u}z}) \qquad \forall \boldsymbol{v}_{\boldsymbol{u}z} \in \mathbb{V}_{\boldsymbol{u}z} \tag{4.13}$$

where the function spaces are defined as

$$\mathbb{V}_{\boldsymbol{u}\xi} = \left\{ \boldsymbol{v} \mid \boldsymbol{v} \in \mathbb{H}^1(\Omega), \boldsymbol{v} = \boldsymbol{0} \text{ on } \Gamma_{\mathrm{D}\xi} \right\} \qquad \xi = x, y, z \tag{4.14}$$

The alternating directions strategy, described in Section 3.2, is used to calculate the components of the new mode. The resulting steps to find the new mode, $\tilde{\boldsymbol{u}}^n$, are presented in Box 4.1.

– Given $\boldsymbol{U}_y^n$ and $\boldsymbol{U}_z^n$; find $\boldsymbol{U}_x^n$ such that

$$a_{\boldsymbol{u}}(\tilde{\boldsymbol{u}}^n, \boldsymbol{v}_{\boldsymbol{u}x}) = -a_{\boldsymbol{u}}(\boldsymbol{u}^{n-1}, \boldsymbol{v}_{\boldsymbol{u}x}) + \ell_{\boldsymbol{u}}(\boldsymbol{v}_{\boldsymbol{u}x}) \qquad \forall \boldsymbol{v}_{\boldsymbol{u}x} \in \mathbb{V}_{\boldsymbol{u}x} \tag{4.15}$$

– Given $\boldsymbol{U}_x^n$ and $\boldsymbol{U}_z^n$; find $\boldsymbol{U}_y^n$ such that

$$a_{\boldsymbol{u}}(\tilde{\boldsymbol{u}}^n, \boldsymbol{v}_{\boldsymbol{u}y}) = -a_{\boldsymbol{u}}(\boldsymbol{u}^{n-1}, \boldsymbol{v}_{\boldsymbol{u}y}) + \ell_{\boldsymbol{u}}(\boldsymbol{v}_{\boldsymbol{u}y}) \qquad \forall \boldsymbol{v}_{\boldsymbol{u}y} \in \mathbb{V}_{\boldsymbol{u}y} \tag{4.16}$$

– Given $\boldsymbol{U}_x^n$ and $\boldsymbol{U}_y^n$; find $\boldsymbol{U}_z^n$ such that

$$a_{\boldsymbol{u}}(\tilde{\boldsymbol{u}}^n, \boldsymbol{v}_{\boldsymbol{u}z}) = -a_{\boldsymbol{u}}(\boldsymbol{u}^{n-1}, \boldsymbol{v}_{\boldsymbol{u}z}) + \ell_{\boldsymbol{u}}(\boldsymbol{v}_{\boldsymbol{u}z}) \qquad \forall \boldsymbol{v}_{\boldsymbol{u}z} \in \mathbb{V}_{\boldsymbol{u}z} \tag{4.17}$$

Box 4.1: *Steps of the alternating directions strategy to iterate for the components of the new mode of the displacement field.*

4.1.1 FEM approximation of the modes

Each component function of the PGD approximation is discretized using one dimensional finite elements, in a standard fashion, as a linear combination of shape functions and nodal values as

$$\boldsymbol{U}_\xi^i(\xi) \approx \boldsymbol{U}_{\xi,h}^i(\xi) = \sum_{k=1}^{N_\xi^{\mathrm{DOFS}}} \boldsymbol{N}_{\xi k}(\xi) \cdot \{\underline{a}_\xi^i\}_k, \qquad \xi = x, y, z \tag{4.18}$$

where $\boldsymbol{N}_{\xi k}$ is the tensorial shape function corresponding to degree of freedom k in the solution vector $\underline{a}_\xi^i$ and N_ξ^{DOFS} is the number of degrees of freedom used for the discretization. The h index, indicating the finite element discretization, is henceforth dropped since only the finite element discretized modes are considered. We also introduce a short hand notation for the sum over the degrees of freedom and write the discretized modes as

$$\boldsymbol{U}_\xi^i = \sum_k \boldsymbol{N}_{\xi k} \cdot \{\underline{a}_\xi^i\}_k, \qquad \xi = x, y, z \tag{4.19}$$

Inserting Eq. (4.19) into the expression for mode i of the displacement in Eq. (4.1) gives

$$\tilde{\boldsymbol{u}}^i(\boldsymbol{x}) = \Big(\sum_k \boldsymbol{N}_{xk} \cdot \{\underline{a}_x^i\}_k \Big) \circ \Big(\sum_l \boldsymbol{N}_{yl} \cdot \{\underline{a}_y^i\}_l \Big) \circ \Big(\sum_m \boldsymbol{N}_{zm} \cdot \{\underline{a}_z^i\}_m \Big) \tag{4.20}$$

and thus the total displacement field, with M modes, is approximated as

$$\boldsymbol{u}^M = \sum_{i=1}^{M} \Big(\sum_k \boldsymbol{N}_{xk} \cdot \{\underline{a}_x^i\}_k \Big) \circ \Big(\sum_l \boldsymbol{N}_{yl} \cdot \{\underline{a}_y^i\}_l \Big) \circ \Big(\sum_m \boldsymbol{N}_{zm} \cdot \{\underline{a}_z^i\}_m \Big) \tag{4.21}$$

The next step is to express the strain in terms of the FE approximated displacement field in Eq. (4.21). To write the strain on separated form, as in Eq. (4.5), we define a second order tensor, with derivatives of the shape functions as

$$\boldsymbol{B}_{xk} = \begin{pmatrix} \partial_x \boldsymbol{N}_{xk} & \boldsymbol{N}_{xk} & \boldsymbol{N}_{xk} \end{pmatrix} \tag{4.22}$$

$$\boldsymbol{B}_{yl} = \begin{pmatrix} \boldsymbol{N}_{yl} & \partial_y \boldsymbol{N}_{yl} & \boldsymbol{N}_{yl} \end{pmatrix} \tag{4.23}$$

$$\boldsymbol{B}_{zm} = \begin{pmatrix} \boldsymbol{N}_{zm} & \boldsymbol{N}_{zm} & \partial_z \boldsymbol{N}_{zm} \end{pmatrix} \tag{4.24}$$

such that the strain tensor of the full solution from Eq. (4.5) can be written as

$$\begin{aligned} \boldsymbol{\varepsilon}^M(\boldsymbol{u}^M) = [\boldsymbol{u}^M \otimes \boldsymbol{\nabla}]^{\mathrm{sym}} = \\ = \sum_{i=1}^{M} \left[\Big(\sum_k \boldsymbol{B}_{xk} \cdot \{\underline{a}_x^i\}_k \Big) \circ \Big(\sum_l \boldsymbol{B}_{yl} \cdot \{\underline{a}_y^i\}_l \Big) \circ \Big(\sum_m \boldsymbol{B}_{zm} \cdot \{\underline{a}_z^i\}_m \Big) \right]^{\mathrm{sym}} \end{aligned} \tag{4.25}$$

As stated in the previous section, it is convenient to split the strain in two parts, as in Eq. (4.7). This gives

$$\begin{aligned} \boldsymbol{\varepsilon}^n(\boldsymbol{u}^n) = [\boldsymbol{u}^{n-1} \otimes \boldsymbol{\nabla}]^{\mathrm{sym}} + [\tilde{\boldsymbol{u}}^n \otimes \boldsymbol{\nabla}]^{\mathrm{sym}} = \\ = \sum_{i=1}^{n-1} \left[\Big(\sum_k \boldsymbol{B}_{xk} \cdot \{\underline{a}_x^i\}_k \Big) \circ \Big(\sum_l \boldsymbol{B}_{yl} \cdot \{\underline{a}_y^i\}_l \Big) \circ \Big(\sum_m \boldsymbol{B}_{zm} \cdot \{\underline{a}_z^i\}_m \Big) \right]^{\mathrm{sym}} + \\ + \left[\Big(\sum_k \boldsymbol{B}_{xk} \cdot \{\underline{a}_x^n\}_k \Big) \circ \Big(\sum_l \boldsymbol{B}_{yl} \cdot \{\underline{a}_y^n\}_l \Big) \circ \Big(\sum_m \boldsymbol{B}_{zm} \cdot \{\underline{a}_z^n\}_m \Big) \right]^{\mathrm{sym}} \end{aligned} \tag{4.26}$$

The test function are discretized in the same way as the displacement field, cf. Eq. (4.20), such that

$$\begin{aligned} \boldsymbol{v}_{\boldsymbol{u}} = \underbrace{\Big(\sum_k \boldsymbol{N}_{xk} \cdot \{\delta \underline{a}_x^n\}_k \Big) \circ \Big(\sum_l \boldsymbol{N}_{yl} \cdot \{\underline{a}_y^n\}_l \Big) \circ \Big(\sum_m \boldsymbol{N}_{zm} \cdot \{\underline{a}_z^n\}_m \Big)}_{\boldsymbol{v}_{\boldsymbol{u}x}} + \\ + \underbrace{\Big(\sum_k \boldsymbol{N}_{xk} \cdot \{\underline{a}_x^n\}_k \Big) \circ \Big(\sum_l \boldsymbol{N}_{yl} \cdot \{\delta \underline{a}_y^n\}_l \Big) \circ \Big(\sum_m \boldsymbol{N}_{zm} \cdot \{\underline{a}_z^n\}_m \Big)}_{\boldsymbol{v}_{\boldsymbol{u}y}} + \\ + \underbrace{\Big(\sum_k \boldsymbol{N}_{xk} \cdot \{\underline{a}_x^n\}_k \Big) \circ \Big(\sum_l \boldsymbol{N}_{yl} \cdot \{\underline{a}_y^n\}_l \Big) \circ \Big(\sum_m \boldsymbol{N}_{zm} \cdot \{\delta \underline{a}_z^n\}_m \Big)}_{\boldsymbol{v}_{\boldsymbol{u}z}} = \\ = \boldsymbol{v}_{\boldsymbol{u}x} + \boldsymbol{v}_{\boldsymbol{u}y} + \boldsymbol{v}_{\boldsymbol{u}z} \end{aligned} \tag{4.27}$$

and the gradient of the test function

$$\boldsymbol{v}_{\boldsymbol{u}} \otimes \boldsymbol{\nabla} = \underbrace{\Big(\sum_k \boldsymbol{B}_{xk} \cdot \{\delta \underline{a}_x^n\}_k\Big) \circ \Big(\sum_l \boldsymbol{B}_{yl} \cdot \{\underline{a}_y^n\}_l\Big) \circ \Big(\sum_m \boldsymbol{B}_{zm} \cdot \{\underline{a}_z^n\}_m\Big) +}_{\boldsymbol{v}_{\boldsymbol{u}x} \otimes \boldsymbol{\nabla}}$$
$$+ \underbrace{\Big(\sum_k \boldsymbol{B}_{xk} \cdot \{\underline{a}_x^n\}_k\Big) \circ \Big(\sum_l \boldsymbol{B}_{yl} \cdot \{\delta \underline{a}_y^n\}_l\Big) \circ \Big(\sum_m \boldsymbol{N}_{zm} \cdot \{\underline{a}_z^n\}_m\Big) +}_{\boldsymbol{v}_{\boldsymbol{u}y} \otimes \boldsymbol{\nabla}}$$
$$+ \underbrace{\Big(\sum_k \boldsymbol{B}_{xk} \cdot \{\underline{a}_x^n\}_k\Big) \circ \Big(\sum_l \boldsymbol{B}_{yl} \cdot \{\underline{a}_y^n\}_l\Big) \circ \Big(\sum_m \boldsymbol{B}_{zm} \cdot \{\delta \underline{a}_z^n\}_m\Big)}_{\boldsymbol{v}_{\boldsymbol{u}z} \otimes \boldsymbol{\nabla}}$$
$$= \boldsymbol{v}_{\boldsymbol{u}x} \otimes \boldsymbol{\nabla} + \boldsymbol{v}_{\boldsymbol{u}y} \otimes \boldsymbol{\nabla} + \boldsymbol{v}_{\boldsymbol{u}z} \otimes \boldsymbol{\nabla} \quad (4.28)$$

The FE discretized expressions are inserted in the weak forms of Eqs. (4.15)-(4.17) and the discrete modes can be solved for.

4.1.2 Solution procedure for the discrete displacement field

To solve for the discrete displacement field, the equations in Box 4.1 are used. They are solved consecutively until convergence for the new mode is reached. In this section the first step is presented in more detail, using the FE discretized field from Section 4.1.1. Again, the first step reads

– Given $\boldsymbol{U}_y^n$ and $\boldsymbol{U}_z^n$; find $\boldsymbol{U}_x^n$ such that

$$a_{\boldsymbol{u}}(\tilde{\boldsymbol{u}}^n, \boldsymbol{v}_{\boldsymbol{u}x}) = -a_{\boldsymbol{u}}(\boldsymbol{u}^{n-1}, \boldsymbol{v}_{\boldsymbol{u}x}) + \ell_{\boldsymbol{u}}(\boldsymbol{v}_{\boldsymbol{u}x}) \qquad \forall \boldsymbol{v}_{\boldsymbol{u}x} \in \mathbb{V}_{\boldsymbol{u}x} \quad (4.29)$$

where we recall the expression for the unknown mode, $\tilde{\boldsymbol{u}}^n$, and the test function, $\boldsymbol{v}_{\boldsymbol{u}x}$ as

$$\tilde{\boldsymbol{u}}^n = \Big(\sum_k \boldsymbol{N}_{xk} \cdot \{\underline{a}_x^n\}_k\Big) \circ \Big(\sum_l \boldsymbol{N}_{yl} \cdot \{\underline{a}_y^n\}_l\Big) \circ \Big(\sum_m \boldsymbol{N}_{zm} \cdot \{\underline{a}_z^n\}_m\Big) \quad (4.30)$$

$$\boldsymbol{v}_{\boldsymbol{u}x} = \Big(\sum_k \boldsymbol{N}_{xk} \cdot \{\delta \underline{a}_x^n\}_k\Big) \circ \Big(\sum_l \boldsymbol{N}_{yl} \cdot \{\underline{a}_y^n\}_l\Big) \circ \Big(\sum_m \boldsymbol{N}_{zm} \cdot \{\underline{a}_z^n\}_m\Big) \quad (4.31)$$

Since, at this step, $\boldsymbol{U}_y^n$ and $\boldsymbol{U}_z^n$ are fixed, the only unknown is $\boldsymbol{U}_x^n$, and for the FE discretized case this means that only $\underline{a}_x^n$ is unknown. Since Eq. (4.29) should hold for all possible $\boldsymbol{v}_{\boldsymbol{u}x}$, or more specifically for all $\delta \underline{a}_x^n$ we can reformulate it as

$$\underline{K}_{\boldsymbol{u}x} \cdot \underline{a}_x^n = \underline{f}_{\boldsymbol{u}x} \quad (4.32)$$

where the components of the stiffness matrix $\underline{K}_{\boldsymbol{u}x}$ and force vector $\underline{f}_{\boldsymbol{u}x}$ are given by

$$\big\{\underline{K}_{\boldsymbol{u}x}\big\}_{kp} = a_{\boldsymbol{u}}(\boldsymbol{v}_{\boldsymbol{u}xp}, \boldsymbol{v}_{\boldsymbol{u}xk}) \quad (4.33)$$

$$\big\{\underline{f}_{\boldsymbol{u}x}\big\}_k = -a_{\boldsymbol{u}}(\boldsymbol{u}^{n-1}, \boldsymbol{v}_{\boldsymbol{u}xk}) + \ell_{\boldsymbol{u}}(\boldsymbol{v}_{\boldsymbol{u}xk}) \quad (4.34)$$

where the k:th and p:th test functions are defined as

$$\boldsymbol{v}_{\boldsymbol{u}xk} = \boldsymbol{N}_{xk} \circ \Big(\sum_l \boldsymbol{N}_{yl} \cdot \{\underline{a}_y^n\}_l\Big) \circ \Big(\sum_m \boldsymbol{N}_{zm} \cdot \{\underline{a}_z^n\}_m\Big) \quad (4.35)$$

$$\boldsymbol{v}_{\boldsymbol{u}xp} = \boldsymbol{N}_{xp} \circ \Big(\sum_l \boldsymbol{N}_{yl} \cdot \{\underline{a}_y^n\}_l\Big) \circ \Big(\sum_m \boldsymbol{N}_{zm} \cdot \{\underline{a}_z^n\}_m\Big) \quad (4.36)$$

From Eq. (4.32) the solution vector $\underline{a}_x^n$ can be solved, corresponding to the $\boldsymbol{U}_x^n$ mode, by inverting the stiffness matrix, i.e.

$$\underline{a}_x^n = \left(\underline{K}_{\boldsymbol{u}x}\right)^{-1} \cdot \underline{f}_{\boldsymbol{u}x} \tag{4.37}$$

With this, the first step of Box 4.1 is completed, and next the solution vector for the $\boldsymbol{U}_y^n$ mode is solved etc. The steps are repeated until a stationary point is reached, and the new mode, $\tilde{\boldsymbol{u}}^n$, is solved.

A note on the integration

As the described in Section 3.3, using PGD allows for the integrals to be evaluated in the decomposed domains. For the case of equilibrium, with constant elastic tangent stiffness matrix, the integrals to calculate the stiffness matrix, and force vector, in Eq. (4.32) can be evaluated along the three one dimensional domains. For example, component kp of the stiffness matrix $\underline{K}_{\boldsymbol{u}x}$ can be written as

$$\begin{aligned}
&\left\{\underline{K}_{\boldsymbol{u}x}\right\}_{kp} = a_{\boldsymbol{u}}(\boldsymbol{v}_{\boldsymbol{u}xp}, \boldsymbol{v}_{\boldsymbol{u}xk}) = \int_\Omega [\boldsymbol{v}_{\boldsymbol{u}xp} \otimes \boldsymbol{\nabla}] : \mathsf{E} : [\boldsymbol{v}_{\boldsymbol{u}xp} \otimes \boldsymbol{\nabla}]^{\text{sym}} \,\mathrm{d}\Omega = \{\mathsf{E} \text{ symmetric}\} = \\
&= \int_\Omega \Bigg[\boldsymbol{B}_{xk} \circ \underbrace{\Big(\sum_l \boldsymbol{B}_{yl} \cdot \{\underline{a}_y^n\}_l\Big)}_{\boldsymbol{U}_{y,y}^n} \circ \underbrace{\Big(\sum_m \boldsymbol{B}_{zm} \cdot \{\underline{a}_z^n\}_m\Big)}_{\boldsymbol{U}_{z,z}^n}\Bigg] : \mathsf{E} : \Bigg[\boldsymbol{B}_{xp} \circ \underbrace{\Big(\sum_q \boldsymbol{B}_{yq} \cdot \{\underline{a}_z^n\}_q\Big)}_{\boldsymbol{U}_{y,y}^n} \circ \underbrace{\Big(\sum_r \boldsymbol{B}_{zr} \cdot \{\underline{a}_z^n\}_r\Big)}_{\boldsymbol{U}_{z,z}^n}\Bigg]^{\text{sym}} \mathrm{d}\Omega = \\
&= \int_\Omega \left[\boldsymbol{B}_{xk} \circ \boldsymbol{U}_{y,y}^n \circ \boldsymbol{U}_{z,z}^n\right] : \mathsf{E} : \left[\boldsymbol{B}_{xp} \circ \boldsymbol{U}_{y,y}^n \circ \boldsymbol{U}_{z,z}^n\right] \mathrm{d}\Omega = \int_\Omega \boldsymbol{B}_{xk} : \left[\boldsymbol{U}_{y,y}^n \boxdot \left[\boldsymbol{U}_{z,z}^n \boxdot \mathsf{E} \boxdot \boldsymbol{U}_{z,z}^n\right] \boxdot \boldsymbol{U}_{y,y}^n\right] : \boldsymbol{B}_{xp} \,\mathrm{d}\Omega = \\
&= \int_x \boldsymbol{B}_{xk} : \underbrace{\int_y \boldsymbol{U}_{y,y}^n \boxdot \underbrace{\int_z \boldsymbol{U}_{z,z}^n \boxdot \mathsf{E} \boxdot \boldsymbol{U}_{z,z}^n \,\mathrm{d}z}_{\mathsf{E}_z} \boxdot \boldsymbol{U}_{y,y}^n \,\mathrm{d}y}_{\mathsf{E}_{zy}} : \boldsymbol{B}_{xp} \,\mathrm{d}x = \int_x \boldsymbol{B}_{xk} : \mathsf{E}_{zy} : \boldsymbol{B}_{xp} \,\mathrm{d}x
\end{aligned} \tag{4.38}$$

where we have introduced the symbol $\boxdot$ to denote a Hadamard like product for a second and fourth order tensor[1]. Now the components of the stiffness matrix can be computed by evaluating a one dimensional integral, since the integrals for E_z and E_{zy} can be calculated in a pre-process. This gives a substantial increase in performance, simply because the one dimensional integrals are much cheaper to evaluate, compared to a three dimensional integral.

[1]Element wise multiplication for the two closest indices; multiplication between a second order tensor $\boldsymbol{A}$ and a fourth order tensor C gives another fourth order tensor D, $\boldsymbol{A} \boxdot \mathsf{C} = \mathsf{D}$ where the components of D are defined as $D_{ijkl} = A_{ij}C_{ijkl}$, no sum for i, j. Such that $(\boldsymbol{A} \circ \boldsymbol{B}) : \mathsf{C} = \boldsymbol{A} : (\boldsymbol{B} \boxdot \mathsf{C})$ and $\mathsf{C} : (\boldsymbol{A} \circ \boldsymbol{B}) = (\mathsf{C} \boxdot \boldsymbol{A}) : \boldsymbol{B}$

4.2 PGD formulation of damage evolution

The PGD formulation of the damage field is similar to the equilibrium equations in Section 4.1, the only difference being the governing equation and that the damage is a scalar field. For completeness, the PGD formulation and FE discretization of the damage evolution is presented here.

The PGD damage field, decomposed in the spatial domains x, y and z, with M modes is written as

$$d(\boldsymbol{x}) = \sum_{i=1}^{M} D_x^i \cdot D_y^i \cdot D_z^i = \sum_{i=1}^{M} \tilde{d}^i = d^M \tag{4.39}$$

where D_x^i, D_y^i and D_z^i are scalar functions, in contrast to the displacement field in Eq. (4.1), which has vector valued component functions. The next step is to set up the gradient of the damage field, which is used in the weak form (Eq. (2.31)). The gradient of the PGD approximated damage field becomes

$$\boldsymbol{\nabla} d^M = \boldsymbol{\nabla}\big(\sum_{i=1}^{M} D_x^i \cdot D_y^i \cdot D_z^i\big) = \sum_{i=1}^{M} \begin{pmatrix} \partial_x D_x^i \cdot D_y^i \cdot D_z^i \\ D_x^i \cdot \partial_y D_y^i \cdot D_z^i \\ D_x^i \cdot D_y^i \cdot \partial_z D_z^i \end{pmatrix} = \sum_{i=1}^{M} \begin{pmatrix} \partial_x D_x^i \\ D_x^i \\ D_x^i \end{pmatrix} \circ \begin{pmatrix} D_y^i \\ \partial_y D_y^i \\ D_y^i \end{pmatrix} \circ \begin{pmatrix} D_z^i \\ D_z^i \\ \partial_z D_z^i \end{pmatrix} \tag{4.40}$$

Similarly as for the displacement field it is, for the solution procedure, convenient to split the field itself and its gradient into one known and one unknown part, cf. Eq. (4.7). For the damage field this gives

$$d^n = d^{n-1} \; + \; \tilde{d}^n \tag{4.41}$$

$$\boldsymbol{\nabla} d^n = \boldsymbol{\nabla} d^{n-1} \; + \; \boldsymbol{\nabla} \tilde{d}^n \tag{4.42}$$

A test function is needed for the weak form, Eq. (2.31), and as for the equilibrium equations it is simply taken as the variation of the field itself. For the PGD damage field this gives

$$v_d = \delta(D_x^n \cdot D_y^n \cdot D_z^n) = \underbrace{\delta D_x^n \cdot D_y^n \cdot D_z^n}_{v_{dx}} + \underbrace{D_x^n \cdot \delta D_y^n \cdot D_z^n}_{v_{dy}} + \underbrace{D_x^n \cdot D_y^n \cdot \delta D_z^n}_{v_{dz}} = v_{dx} + v_{dy} + v_{dz} \tag{4.43}$$

$$\boldsymbol{\nabla} v_d = \boldsymbol{\nabla} v_{dx} + \boldsymbol{\nabla} v_{dy} + \boldsymbol{\nabla} v_{dz} \tag{4.44}$$

The PGD approximated field, Eq. (4.39) can now be inserted in the weak form from Eq. (2.31), which gives

$$\begin{aligned} &\underbrace{\int_\Omega l^2 \; \boldsymbol{\nabla} v_d \cdot \boldsymbol{\nabla} \tilde{d}^n + \Big(1 + \frac{2l}{G_c}\mathcal{H}\Big) v_d \tilde{d}^n \, \mathrm{d}\Omega}_{a_d(\tilde{d}^n, v_d)} = \\ &\qquad = \underbrace{-\int_\Omega l^2 \; \boldsymbol{\nabla} v_d \cdot \boldsymbol{\nabla} d^{n-1} + \Big(1 + \frac{2l}{G_c}\mathcal{H}\Big) v_d d^{n-1} \, \mathrm{d}\Omega}_{a_d(d^{n-1}, v_d)} + \underbrace{\int_\Omega v_d \frac{2l}{G_c}\mathcal{H} \, \mathrm{d}\Omega}_{\ell_d(v_d)} \quad \forall v_d \in \mathbb{V}_d \end{aligned} \tag{4.45}$$

By inserting the expressions for the test function, Eq. (4.43), and its gradient, Eq. (4.44) into Eq. (4.45) three equations are obtained. From these equations the unknown mode can be found, i.e.; find $\tilde{d}^n \in \mathbb{U}_d$ such that

$$a_d(\tilde{d}^n, v_{dx}) = a_d(d^{n-1}, v_{dx}) + \ell_d(v_{dx}) \quad \forall v_{dx} \in \mathbb{V}_{dx} \tag{4.46}$$
$$a_d(\tilde{d}^n, v_{dy}) = a_d(d^{n-1}, v_{dy}) + \ell_d(v_{dy}) \quad \forall v_{dy} \in \mathbb{V}_{dy} \tag{4.47}$$
$$a_d(\tilde{d}^n, v_{dz}) = a_d(d^{n-1}, v_{dz}) + \ell_d(v_{dz}) \quad \forall v_{dz} \in \mathbb{V}_{dz} \tag{4.48}$$

with the function spaces defined as

$$\mathbb{V}_{d\xi} = \left\{ v \mid v \in \mathbb{H}^1(\Omega), v = 0 \text{ on } \Gamma_{\mathrm{D}\xi} \right\} \qquad \xi = x, y, z \tag{4.49}$$

The alternating directions strategy is used to solve Eqs. (4.46)-(4.48) to obtain the new mode for the damage field. In Box 4.2 the resulting steps to calculate the mode is presented, cf. Box 4.1 for the equations of equilibrium.

– Given D_y^n and D_z^n; find D_x^n such that

$$a_d(\tilde{d}^n, v_{dx}) = -a_d(d^{n-1}, v_{dx}) + \ell_d(v_{dx}) \qquad \forall v_{dx} \in \mathbb{V}_{dx} \tag{4.50}$$

– Given D_x^n and D_z^n; find D_y^n such that

$$a_d(\tilde{d}^n, v_{dy}) = -a_d(d^{n-1}, v_{dy}) + \ell_d(v_{dy}) \qquad \forall v_{dy} \in \mathbb{V}_{dy} \tag{4.51}$$

– Given D_x^n and D_y^n; find D_z^n such that

$$a_d(\tilde{d}^n, v_{dz}) = -a_d(d^{n-1}, v_{dz}) + \ell_d(v_{dz}) \qquad \forall v_{dz} \in \mathbb{V}_{dz} \tag{4.52}$$

Box 4.2: *Steps of the alternating directions strategy to iterate for the components of the new mode of the damage field.*

4.2.1 FEM approximation of the damage field

The component functions of the damage field is approximated by one dimensional finite elements in the same way as for the displacement field. Each component is thus approximated as

$$D_\xi^i(\xi) \approx D_{\xi,h}^i(\xi) = \sum_k N_{\xi k}(\xi) \cdot \{\underline{b}_\xi^i\}_k, \qquad \xi = x, y, z \tag{4.53}$$

where $N_{\xi k}$ is the (scalar valued) shape function corresponding to the degree of freedom k in the solution vector $\underline{b}_\xi^i$. For the sum over the degrees of freedom the short hand notation introduced in Section 4.1.1 is used, and the index h is dropped, since only the FE discretized components are used henceforth. With this approximation, mode i can be written as

$$\tilde{d}^i(\boldsymbol{x}) = \Big(\sum_k N_{xk} \cdot \{\underline{b}_x^i\}_k \Big) \cdot \Big(\sum_l N_{yl} \cdot \{\underline{b}_y^i\}_l \Big) \cdot \Big(\sum_m N_{zm} \cdot \{\underline{b}_z^i\}_m \Big) \tag{4.54}$$

and thus the total damage field, including all modes, can be written as

$$d_h^M = \sum_{i=1}^{M} \Big(\sum_k N_{xk} \cdot \{\underline{b}_x^i\}_k \Big) \cdot \Big(\sum_l N_{yl} \cdot \{\underline{b}_y^i\}_l \Big) \cdot \Big(\sum_m N_{zm} \cdot \{\underline{b}_z^i\}_m \Big) \tag{4.55}$$

To express the gradient of the damage field, with FE discretized modes, we define the gradient of the shape functions as

$$\boldsymbol{B}_{xk} = \begin{pmatrix} \partial_x N_{xk} \\ N_{xk} \\ N_{xk} \end{pmatrix}, \quad \boldsymbol{B}_{xl} = \begin{pmatrix} N_{yl} \\ \partial_y N_{yl} \\ N_{yl} \end{pmatrix}, \quad \boldsymbol{B}_{zm} = \begin{pmatrix} N_{zm} \\ N_{zm} \\ \partial_z N_{zm} \end{pmatrix} \tag{4.56}$$

Note the difference from Eqs. (4.22)-(4.24), which are second order tensors. With Eq. (4.56), the gradient of the damage can be written as

$$\boldsymbol{\nabla} d^M = \sum_{i=1}^{M} \Big(\sum_k B_{xk} \cdot \{\underline{b}_x^i\}_k \Big) \cdot \Big(\sum_l B_{yl} \cdot \{\underline{b}_y^i\}_l \Big) \cdot \Big(\sum_m B_{zm} \cdot \{\underline{b}_z^i\}_m \Big) \tag{4.57}$$

The test function, Eq. (4.43), are discretized in the same way as the damage in Eq. (4.54), which gives

$$\begin{aligned} v_d = & \underbrace{\Big(\sum_k N_{xk} \cdot \{\delta\underline{b}_x^n\}_k \Big) \cdot \Big(\sum_l N_{yl} \cdot \{\underline{b}_y^n\}_l \Big) \cdot \Big(\sum_m N_{zm} \cdot \{\underline{b}_z^n\}_m \Big)}_{v_{dx}} + \\ & + \underbrace{\Big(\sum_k N_{xk} \cdot \{\underline{b}_x^n\}_k \Big) \cdot \Big(\sum_l N_{yl} \cdot \{\delta\underline{b}_y^n\}_l \Big) \cdot \Big(\sum_m N_{zm} \cdot \{\underline{b}_z^n\}_m \Big)}_{v_{dy}} + \\ & + \underbrace{\Big(\sum_k N_{xk} \cdot \{\underline{b}_x^n\}_k \Big) \cdot \Big(\sum_l N_{yl} \cdot \{\underline{b}_y^n\}_l \Big) \cdot \Big(\sum_m N_{zm} \cdot \{\delta\underline{b}_z^n\}_m \Big)}_{v_{dz}} = \\ & = v_{dx} + v_{dy} + v_{dz} \end{aligned} \tag{4.58}$$

and for the gradient in Eq. (4.44)

$$\begin{aligned} \boldsymbol{\nabla} v_d = & \underbrace{\Big(\sum_k B_{xk} \cdot \{\delta\underline{b}_x^n\}_k \Big) \cdot \Big(\sum_l B_{yl} \cdot \{\underline{b}_y^n\}_l \Big) \cdot \Big(\sum_m B_{zm} \cdot \{\underline{b}_z^n\}_m \Big)}_{v_{dx}} + \\ & + \underbrace{\Big(\sum_k B_{xk} \cdot \{\underline{b}_x^n\}_k \Big) \cdot \Big(\sum_l B_{yl} \cdot \{\delta\underline{b}_y^n\}_l \Big) \cdot \Big(\sum_m B_{zm} \cdot \{\underline{b}_z^n\}_m \Big)}_{v_{dy}} + \\ & + \underbrace{\Big(\sum_k B_{xk} \cdot \{\underline{b}_x^n\}_k \Big) \cdot \Big(\sum_l B_{yl} \cdot \{\underline{b}_y^n\}_l \Big) \cdot \Big(\sum_m B_{zm} \cdot \{\delta\underline{b}_z^n\}_m \Big)}_{v_{dz}} = \\ & = \boldsymbol{\nabla} v_{dx} + \boldsymbol{\nabla} v_{dy} + \boldsymbol{\nabla} v_{dz} \end{aligned} \tag{4.59}$$

The FE discretized damage, and corresponding test function, can now be used in Eqs. (4.50)-(4.52) to solve for the new discrete mode.

4.2.2 Solution procedure for the discrete damage field

The discrete component functions for the new damage mode are obtained by solving Eqs. (4.50)-(4.52) using the alternating direction strategy, similarly as for the displacement field, cf. Section 4.1.2. The first step reads

– Given D_y^n and D_z^n; find D_x^n such that

$$a_d(\tilde{d}^n, v_{dx}) = -a_d(d^{n-1}, v_{dx}) + \ell_d(v_{dx}) \qquad \forall v_{dx} \in \mathbb{V}_{dx} \tag{4.60}$$

which, by inserting the FE discretized damage and test function from Section 4.2.1, simplifies to

$$\underline{K}_{dx} \cdot \underline{b}_x^n = \underline{f}_{dx} \tag{4.61}$$

where the components of the stiffness matrix $\underline{K}_{dx}$ and the force vector $\underline{f}_{dx}$ are defined as

$$\{\underline{K}_{dx}\}_{kp} = a_d(v_{dxp}, v_{dxk}) \tag{4.62}$$

$$\{\underline{f}_{dx}\}_k = -a_d(d^{n-1}, v_{dxk}) + \ell_d(v_{dxk}) \tag{4.63}$$

and the k:th and p:th test functions defined as

$$v_{dxk} = N_{xk} \cdot \Big(\sum_l N_{yl} \cdot \{\underline{b}_y^i\}_l \Big) \cdot \Big(\sum_m N_{zm} \cdot \{\underline{b}_z^i\}_m \Big) \tag{4.64}$$

$$v_{dxp} = N_{xp} \cdot \Big(\sum_l N_{yl} \cdot \{\underline{b}_y^i\}_l \Big) \cdot \Big(\sum_m N_{zm} \cdot \{\underline{b}_z^i\}_m \Big) \tag{4.65}$$

The system of equations in Eq. (4.61) can now be solved to obtain the solution vector $\underline{b}_x^n$, corresponding to component function D_x^n, by inverting the stiffness matrix, i.e.

$$\underline{b}_x^n = \left\{\underline{K}_{dx}\right\}^{-1} \cdot \underline{f}_{dx} \tag{4.66}$$

The steps of Box 4.2 are iterated in this fashion, until a stationary point is reached, and the new mode, $\tilde{d}^n$, is found.

5 Numerical examples

Some numerical example problems are solved and the results are presented here. First for elasticity problems, without any damage field, and then for some cases with couples damage. To describe the accuracy of the PGD approximated function, a FEM solution is used as a reference. To compare these, an error measure is defined as

$$E_{\bullet}^{i} = \frac{\int_{\Omega} \left(\| \bullet^{i} \| - \| \bullet_{\text{FEM}} \| \right)^{2} \mathrm{d}\Omega}{\int_{\Omega} \left(\| \bullet_{\text{FEM}} \| \right)^{2} \mathrm{d}\Omega} \tag{5.1}$$

where $\bullet^{i}$ is the approximated function with i modes, and $\bullet_{\text{FEM}}$ the corresponding FEM solution for the same problem, with the same mesh size. As will now be presented, the PGD and FEM solution show quite good agreement.

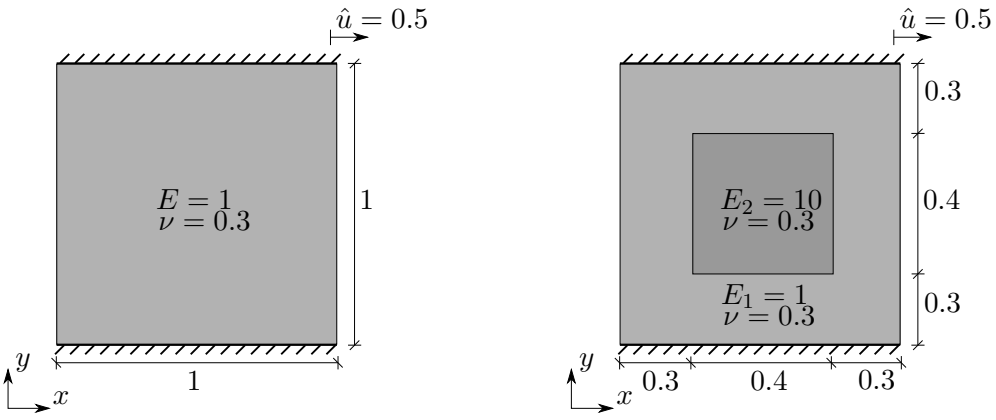
5.1 Two dimensional elasticity

Two simple load cases can be seen in Figure 5.1. The two squares are loaded in the same way; with a prescribed displacement at the top, and fixed displacement at the bottom. The second case also has a hard inclusion in the middle, with stiffness ten times higher than the surrounding material. The material is linear elastic with Young's modulus $E = 1$ and Poisson's ratio $\nu = 0.3$ and plane strain is assumed. The non-homogeneous Dirichlet boundary conditions in this example are, as described previously, handled by a first mode, which fulfills them. For this case it is taken as

$$\tilde{\boldsymbol{u}}^{1} = \begin{pmatrix} U_x^1(x) \cdot U_y^1(y) \\ V_x^1(x) \cdot V_y^1(y) \end{pmatrix} = \begin{pmatrix} 1 \cdot \frac{y}{L}\hat{u} \\ 0 \cdot 0 \end{pmatrix} \tag{5.2}$$

and the rest of the components will be zero at the Dirichlet boundary, i.e.

$$\begin{aligned} U_y^i(y=0) = U_y^i(y=1) = 0 \\ V_y^i(y=0) = V_y^i(y=1) = 0 \end{aligned} \qquad i = 2, 3, 4 \ldots \tag{5.3}$$

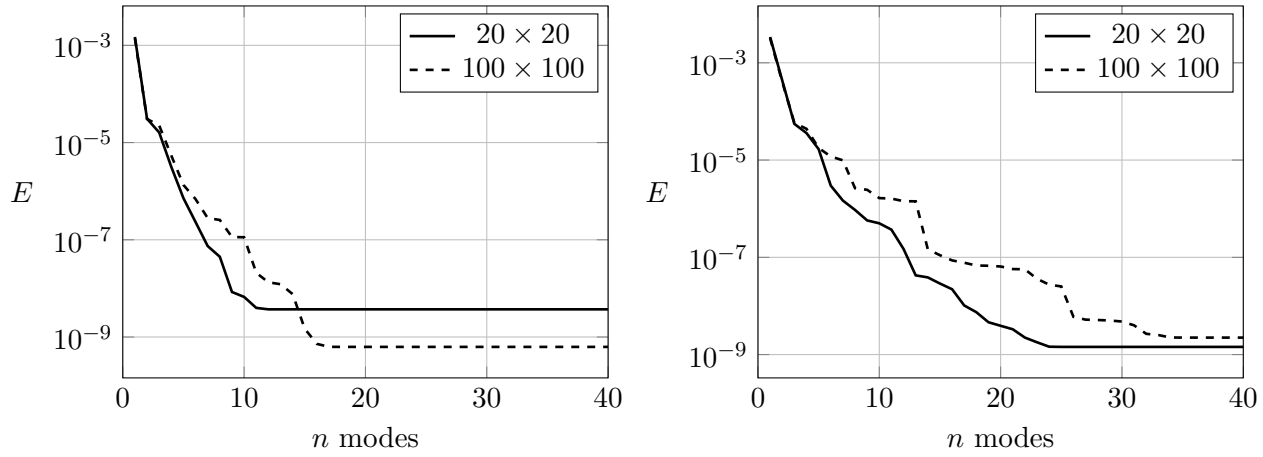


(a) *Square with homogeneous material.* (b) *Square with hard inclusion in the middle.*

Figure 5.1: *Example cases; (a) with homogeneous material and (b) with a hard inclusion in the middle. Both loaded with a prescribed displacement at the top and fixed at the bottom.*

5.1.1 Case 1

The first case is solved with two different meshes; one with 20×20 elements and one with 100×100 elements. The difference, as defined in Eq. (5.1), between the PGD solution and the corresponding FE solution is shown in Figure 5.2a for solutions containing up to 40 modes. As the figure shows, the displacements corresponds well to the FE solution. Note that after 15 modes, the PGD solution is not enhanced more, and the solution has thus converged. This means that the rest of the modes does not contribute to the solution.



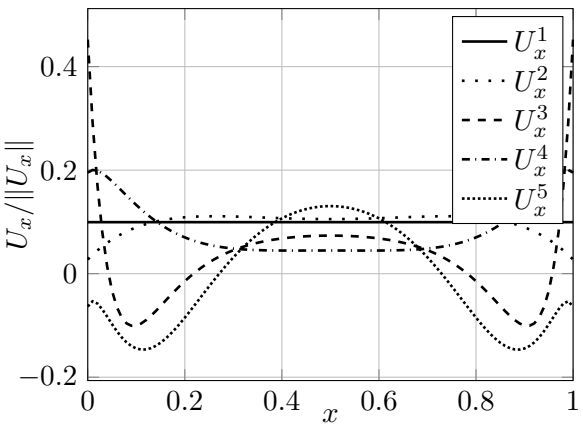
(a) *Case 1.* (b) *Case 2.*

Figure 5.2: *Difference between the PGD solution and the corresponding FEM solution for the two cases in Figure 5.1, the difference is evaluated according to Eq.* (5.1).

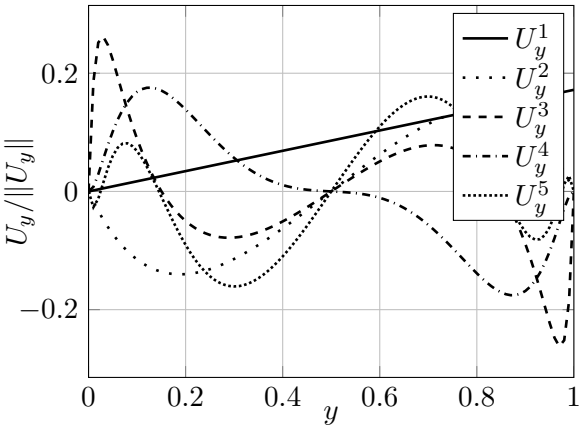
The component functions for the first 5 modes are plotted in Figure 5.3. Interesting to note is the simple first mode, which fulfills the non-homogeneous boundary conditions.

5.1.2 Case 2

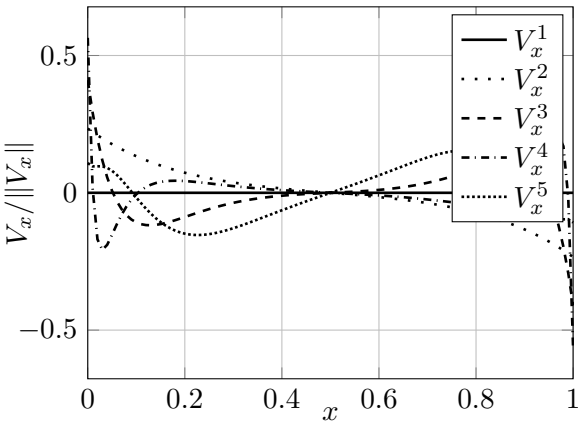
The second case in Figure 5.1 is more interesting since for this case the problem is not as smooth as the first one, due to the inclusion. This case is also solved with two different mesh sizes; one with 20×20 elements and one with 100×100 elements. The difference between the PGD solution and a FEM solution is plotted in Figure 5.2b for solutions up to 40 modes. For this second case more modes are needed, around 25, before the rest of the modes does not contribute. The component functions corresponding to the first five modes are plotted in Figure 5.4. The effect of the inclusion, with harder material in the middle, is clearly seen from the modes.



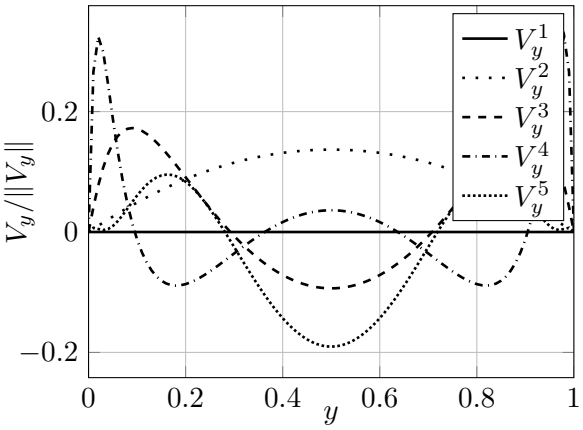
(a) *Component* U_x^i.



(b) *Component* U_y^i.

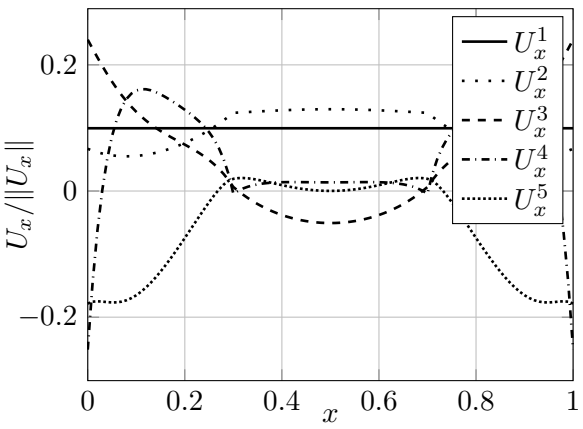


(c) *Component* V_x^i.

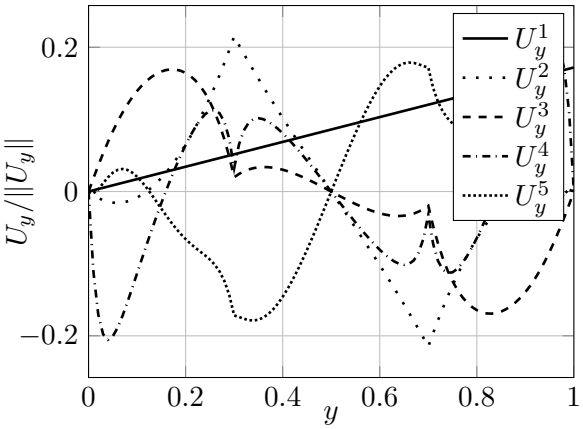


(d) *Component* V_y^i.

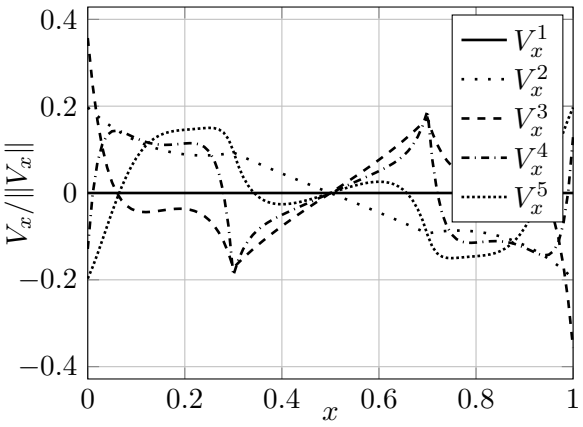
Figure 5.3: *The normalized components to the first five modes for case 1 using a 100x100 mesh.*



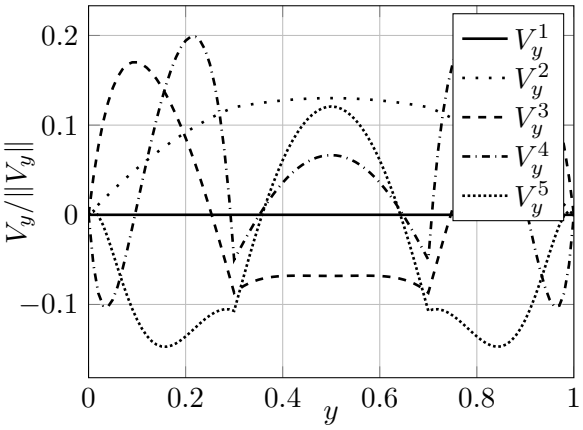
(a) *Component* U_x^i.



(b) *Component* U_y^i.



(c) *Component* V_x^i.



(d) *Component* V_y^i.

Figure 5.4: *The first five modes for case 2 using a 100x100 mesh.*

5.2 Three dimensional elasticity

The equilibrium equations are also solved in three dimensions. In Figure 5.5 a cantilever beam is shown, which is subjected to a uniform volume load in the negative z-direction. The length to width to thickness ratio is $10:1:1$ and the beam is divided into cubic elements, with ratio $1:1:1$. The problem is solved for different meshes where the width/thickness sides are divided into N elements, resulting in meshes with $(10N \times N \times N)$ elements.

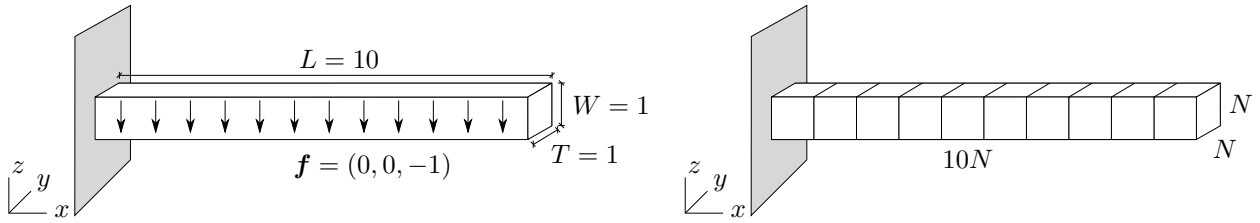


Figure 5.5: *Cantilever beam with length L, width W and thickness T loaded with a uniform volume load $\boldsymbol{f}$, with a discretization using N elements along the thickness/width sides and $10N$ elements in the length direction.*

The difference between the PGD solution and FEM solution, for a mesh with $N = 5$, is plotted in Figure 5.6 for PGD solutions with up to 60 modes. The solutions corresponds well, and around 20 modes the solutions are indistinguishable from each other.

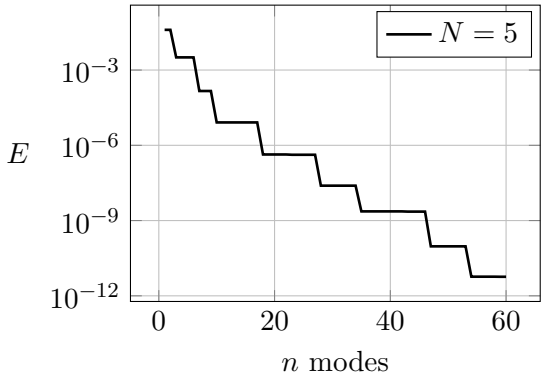


Figure 5.6: *Difference between the PGD and FEM solution for the cantilever beam in Figure 5.5 for a mesh with $100 \times 10 \times 10$ elements.*

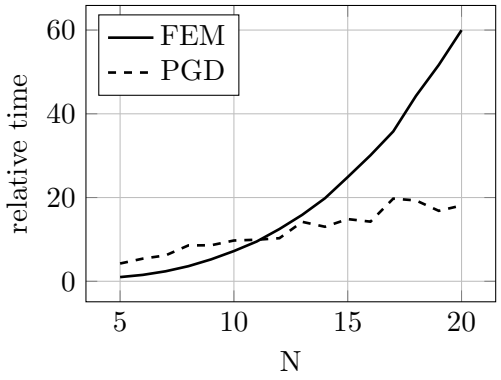


Figure 5.7: *Solution time for solving the cantilever beam in Figure 5.5 for FEM and PGD. The times are normalized with respect to the time of the FEM solution for the coarsest grid with $N = 5$.*

It is of interest to compare the computational time of the PGD solution with the FEM solution. The cantilever beam in Figure 5.5 is solved for a number of different meshes, with $N = 5, 10, 15$ and 20. The time to solve each problem is presented in Table 5.1, and the times are normalized with respect to the time for the FEM solution of the coarsest mesh. For the PGD solution, 20 modes are used, which, from Figure 5.6, should be accurate enough. The relative time is also plotted in Figure 5.7. Note that the time for the FEM solution increases exponentially, but the time for the PGD solution increases linearly.

Table 5.1: *Solution time, degrees of freedom, and number of iterations for the cantilever beam with different meshes N resulting in $(10N \times N \times N)$ elements. The times are normalized with respect to the time of the FEM solution for the coarsest grid.*

	$N = 5$		$N = 10$		$N = 15$		$N = 20$	
	FEM	PGD	FEM	PGD	FEM	PGD	FEM	PGD
rel. time	1.0	4.6	8.3	11.4	29.4	17.4	72.7	21.6
DOFS	5 508	189	36 663	369	115 968	549	265 923	729
iterations	-	757	-	916	-	924	-	869

5.3 Phase field damage

An example problem with phase field damage is shown in Figure 5.8. It is a square which is loaded with a prescribed displacement in the y-direction at the top boundary. An initial crack, Λ, are also included. The initial crack is inserted by prescribing the driving force, $\mathcal{H}$ along Λ, as described in Section 2.2.

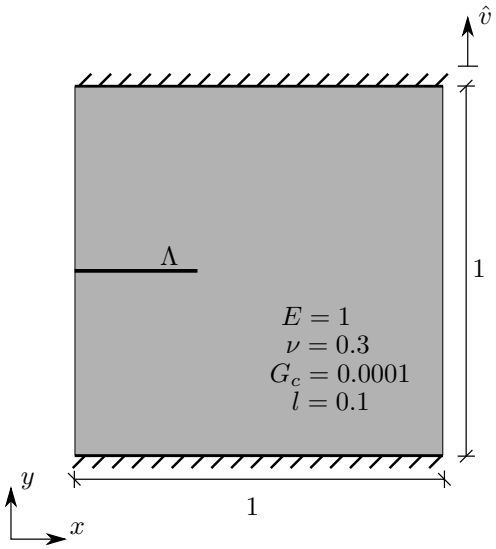


Figure 5.8: *Square with initial crack, Λ, loaded with a prescribed displacement in the y-direction at the upper boundary.*

The resulting damage field for a mesh with 50×50 elements, and 40 load steps, can be seen in Figure 5.9 for both a PGD solution and a reference FEM solution. As seen in the figure, the field shows good agreement in the beginning, but the evolution differs somewhat. For this case 10 modes where used, which gave good correspondence with the finite element solution for the initial crack. It is possible that more modes are needed to describe the damage field as damage evolves, which could be one reason the fields does not agree fully after some time steps.

The convergence for the damage field has been a huge problem. Especially since a large number of iterations are needed to find each mode. This means that the PGD solution cannot compete with the corresponding FEM solution in terms of computational time.

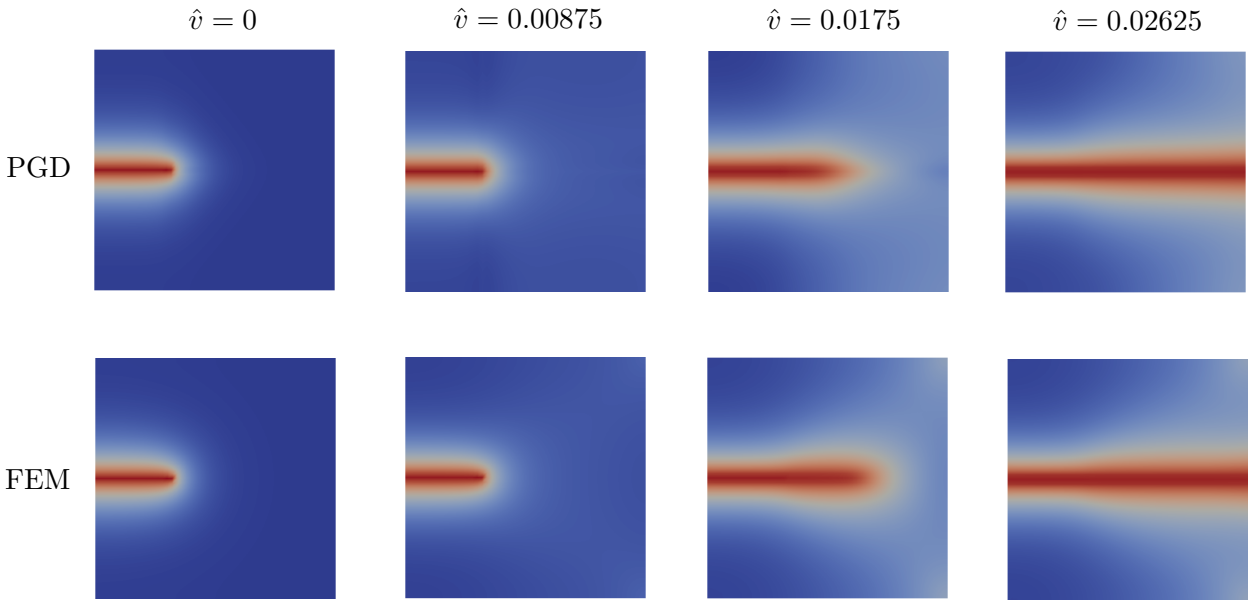


Figure 5.9: *Evolution of the damage for the example in Figure 5.8 for a PGD solution and a corresponding FEM solution.*

6 Conclusions

In this thesis the model reduction technique PGD has been studied. It has been applied to two different differential equations with varying results. For the elasto-static problem the method worked quite well, and the solution agrees well with the finite element solution. However, some issues with convergence were encountered, and a large number of iterations were sometimes needed. For larger mesh sizes, the PGD is still faster than standard finite elements, provided that one dimensional integration is used for the stiffness matrices and force vectors.

The final problem studied in this thesis was the coupled problem with displacement and damage field as unknowns. For this application severe convergence problems where encountered, especially for the damage field, for which the modes where harder to find. Although it is possible to describe the damage with PGD, and get good agreement with the finite element solution for an initial damage field, the evolution of the damage does not coincide. One explanation could be that the damage field is not a very smooth field, and the force that drives the damage, the energy field, is not very smooth either.

The PGD shows some promising results for some of the applications investigated in this thesis. The main complication, for all of the investigated problems, is associated with the solution method used to find the new modes, it takes many iterations for each mode to converge. Therefore, to obtain a faster solution procedure, the stiffness matrices needs to be integrated in the lower domains in order to compete with the standard finite element method for the types of problems studied here.

6.1 Future work

There are a number of topics regarding PGD that needs to be studied more. The first thing is the solution method as indicated above. The alternating directions strategy, which is used in this thesis, indeed proved to be more stable than Newton's method for this application as well However, it is still not very good, since the convergence rate is slow, and a (sometimes) large number of iterations is needed to find each new mode. It also turned out to be quite sensitive to the provided initial guess. This is another problem connected to the solution method, since it is hard to provide the solver with an educated guess on the new mode. There seem to be no good methods developed to get a good initial guess. But with a good initial guess the number of iterations decreases substantially.

Another complication with PGD, which also calls for further investigation, is the difficulties it comes with, especially with regard to boundary conditions and varying domains. Connected to this is also the use of internal variables, which needs to stored for the next time step. In this thesis this was dealt with by integrating the full domain, and saving the variables in the integration points, as one would do for a normal FE problem. However, this is not desired, since the property of integrating in lower dimensions is wanted for computational efficiency.

References

[1] M. Ambati, T. Gerasimov, and L. De Lorenzis. A review on phase-field models of brittle fracture and a new fast hybrid formulation. *Computational Mechanics* **55**.2 (Feb. 2015), 383–405. ISSN: 0178-7675. DOI: 10.1007/s00466-014-1109-y. URL: http://link.springer.com/10.1007/s00466-014-1109-y.

[2] A. Ammar et al. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. *Journal of Non-Newtonian Fluid Mechanics* **139**.3 (Dec. 2006), 153–176. ISSN: 03770257. DOI: 10.1016/j.jnnfm.2006.07.007. URL: http://linkinghub.elsevier.com/retrieve/pii/S0377025706001662.

[3] A. Ammar et al. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids. *Journal of Non-Newtonian Fluid Mechanics* **144**.2-3 (July 2007), 98–121. ISSN: 03770257. DOI: 10.1016/j.jnnfm.2007.03.009. URL: http://linkinghub.elsevier.com/retrieve/pii/S0377025707000821.

[4] H. Amor, J. J. Marigo, and C. Maurini. Regularized formulation of the variational brittle fracture with unilateral contact: Numerical experiments. *Journal of the Mechanics and Physics of Solids* **57**.8 (2009), 1209–1229. ISSN: 00225096. DOI: 10.1016/j.jmps.2009.04.011. URL: http://dx.doi.org/10.1016/j.jmps.2009.04.011.

[5] B. Bognet et al. Advanced simulation of models defined in plate geometries: 3D solutions with 2D computational complexity. *Computer Methods in Applied Mechanics and Engineering* **201-204** (Jan. 2012), 1–12. ISSN: 00457825. DOI: 10.1016/j.cma.2011.08.025. URL: http://linkinghub.elsevier.com/retrieve/pii/S0045782511002891.

[6] M. J. Borden et al. A phase-field description of dynamic brittle fracture. *Computer Methods in Applied Mechanics and Engineering* **217-220** (2012), 77–95. ISSN: 00457825. DOI: 10.1016/j.cma.2012.01.008. URL: http://dx.doi.org/10.1016/j.cma.2012.01.008.

[7] B. Bourdin, G. A. Francfort, and J. J. Marigo. Numerical experiments in revisited brittle fracture. *Journal of the Mechanics and Physics of Solids* **48**.4 (Apr. 2000), 797–826. ISSN: 00225096. DOI: 10.1016/S0022-5096(99)00028-9. URL: http://linkinghub.elsevier.com/retrieve/pii/S0022509699000289.

[8] F. Chinesta, R. Keunings, and A. Leygue. *The Proper Generalized Decomposition for Advanced Numerical Simulations*. SpringerBriefs in Applied Sciences and Technology. Cham: Springer International Publishing, 2014, p. 117. ISBN: 978-3-319-02864-4. DOI: 10.1007/978-3-319-02865-1. URL: http://link.springer.com/10.1007/978-3-319-02865-1.

[9] G. Francfort and J.-J. Marigo. Revisiting brittle fracture as an energy minimization problem. *Journal of the Mechanics and Physics of Solids* **46**.8 (Aug. 1998), 1319–1342. ISSN: 00225096. DOI: 10.1016/S0022-5096(98)00034-9. URL: http://linkinghub.elsevier.com/retrieve/pii/S0022509698000349.

[10] D. González et al. Recent advances on the use of separated representations. *International Journal for Numerical Methods in Engineering* July 2009 (2009), n/a–n/a. ISSN: 00295981. DOI: 10.1002/nme.2710. arXiv: 1010.1724. URL: http://doi.wiley.com/10.1002/nme.2710.

[11] A. Leygue and E. Verron. A First Step Towards the Use of Proper General Decomposition Method for Structural Optimization. *Archives of Computational Methods in Engineering* **17**.4 (Dec. 2010), 465–472. ISSN: 1134-3060. DOI: 10.1007/s11831-010-9052-3. URL: http://link.springer.com/10.1007/s11831-010-9052-3.

[12] S. Metoui et al. Simulation of Laminated Structures Using the Proper. June (2014), 22–26.

[13] C. Miehe, M. Hofacker, and F. Welschinger. A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits. *Computer Methods in Applied Mechanics and Engineering* **199**.45-48 (Nov. 2010), 2765–2778. ISSN: 00457825. DOI:

10.1016/j.cma.2010.04.011. URL: http://linkinghub.elsevier.com/retrieve/pii/S0045782510001283.

[14] P. Vidal, L. Gallimard, and O. Polit. Assessment of a composite beam finite element based on the proper generalized decomposition. *Composite Structures* **94**.5 (Apr. 2012), 1900–1910. ISSN: 02638223. DOI: 10.1016/j.compstruct.2011.12.016. URL: http://dx.doi.org/10.1016/j.compstruct.2011.12.016%20http://linkinghub.elsevier.com/retrieve/pii/S0263822311004843.

[15] S. Zlotnik et al. Proper generalized decomposition of a geometrically parametrized heat problem with geophysical applications. *International Journal for Numerical Methods in Engineering* **103**.10 (Sept. 2015), 737–758. ISSN: 00295981. DOI: 10.1002/nme.4909. arXiv: 1010.1724. URL: http://arxiv.org/abs/1010.1724%20http://doi.wiley.com/10.1002/nme.4909.