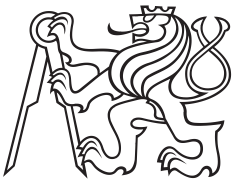
Doctoral Thesis



Czech Technical University in Prague

F1

Faculty of Civil Engineering
Department of Mathematics

Advanced spectral methods for computational homogenization of periodic media

Ing. Martin Ladecký

Supervisor: doc. RNDr. Ivana Pultarová, Ph.D.
Supervisors–specialists: prof. Ing. Jan Zeman, Ph.D. and Ing. Jan Novák, Ph.D.
Field of study: (P3604) Civil Engineering
Subfield: (3607V034) Mathematics in Civil Engineering
June 2022

Acknowledgements

First and most importantly, I wish to thank my supervisors Ivana Pultarová and Jan Zeman. Your extraordinary support and deep understanding helped me to overcome many difficulties during my studies. Only the given opportunities and the trust you had in me allowed me to explore the charming world of academia. Thank you.

Secondly, I would like to thank Jan Novák for inspiring discussions and all my friends, colleagues, and coauthors for their collaboration. Jaroslav Vondřejc, Till Junge, Stephanie Krueger, Ali Falsafi, Richard Leute, Lars Pastewka, Dishi Liu, and Hermann G. Matthies, thank you for your kindness.

During my studies, I was fortunate to spend several months at the Technical University of Braunschweig and the Swiss Federal Institute of Technology Lausanne. Hereby, I would like to thank Jaroslav Vondřejc and Till Junge for memorable experiences.

Finally, none of this would have been possible without the support of my wife Veronika and daughter Zuzka. Thank you for always being there for me.

Declaration

PhD candidate: Martin Ladecký

Title: Advanced spectral methods for computational homogenization of periodic media

Herewith, I declare that the enclosed doctoral thesis is my own work, written under the professional guidance of Ivana Pultarová, Jan Zeman and Jan Novák. I also certify that all used resources and data are cited and referenced.

The doctoral thesis has been written in connection with the research projects supported by the Center of Advanced Applied Sciences, the European Regional Development Fund (project No. CZ 02.1.01/0.0/0.0/16 019/0000778), the Czech Science Foundation (projects Nos. 20-14736S and 17-04150J), and the Grant Agency of the Czech Technical University in Prague (projects Nos. SGS18/005-, SGS19/002-, SGS20/002-, SGS21/003-, and SGS22/004/OHK1/1T/11)

In Prague, 22. June 2022

Abstract

Multiscale material modeling is one of the enabling fields for future industries. To fully exploit the opportunities of multiscale structures, multiscale modeling techniques must be accurate and accessible. This thesis focuses on iterative computational homogenization methods specialized for digitized microstructures, i.e. microstructures with geometries defined on regular grids. These so-called spectral methods exploit discrete Green's operator preconditioning to maintain mesh-size independent iteration count and the fast Fourier transform to achieve $n \log(n)$ computational complexity. This thesis focuses on three topics through a collection of five manuscripts. First, it discusses the effect of discrete Green's operator preconditioning on the spectra of linear system matrices. The first and second chapters provide guaranteed, easily computable, two-sided bounds on individual eigenvalues. These bounds reveal the distribution of eigenvalues which helps to understand grid-size independence of spectral methods. Second, the thesis discusses the problem of ringing artifacts that pollute solution gradient fields of spectral methods. The third chapter provides a detailed description of the finite element discretization approach that eliminates ringing artifacts while keeping the efficiency of spectral methods. The fourth chapter then analyzes several discretizations to confirm that the finite elements deliver the solutions with the least discretization artifacts. Third, the thesis discusses the reduction of computational costs by using reduced-order modeling. The fifth chapter shows the potential and efficiency of low-rank tensor techniques in spectral methods for large-scale problems.

Keywords: computational homogenization, FFT-based methods, spectral methods, finite element method, eigenvalue bounds, discrete Green's operator preconditioning

Abstrakt

Víceúrovňové materiálové modelování je jednou ze klíčových oblastí vývoje pro budoucí průmyslová odvětví. Pro plné využití možností víceúrovňových struktur musí být techniky víceúrovňového modelování přesné a dostupné. Tato práce se zaměřuje na iterační výpočetní homogenizační metody specializované na digitalizované mikrostruktury, tedy mikrostruktury s geometrií definovanou na pravidelných mřížkách. Tyto takzvané spektrální metody využívají předpodmínění pomocí diskrétního Greenova operátoru k udržení počtu iterací nezávislých na velikosti sítě a rychlé Fourierovy transformace k dosažení výpočetní složitosti $n \log(n)$. Prostřednictvím souboru pěti rukopisů se tato práce zaměřuje na tři témata. Nejprve se práce zabývá vlivem předpodmínění diskrétním Greenovým operátorem na spektra matic lineárních systémů. První a druhá kapitola popisuje lehce dostupné, garantované oboustranné odhady jednotlivých vlastních čísel. Tyto odhady popisují distribuci vlastních čísel, což pomáhá pochopit nezávislost rychlosti konvergence spektrálních metod na velikosti síte. Zadruhé se práce zabývá problémem oscilujících diskretizačích chyb, které degradují rešení spektrálních metod. Třetí kapitola podrobně popisuje přístup založený na metodě konečných prvků, který eliminuje tyto oscilace při zachování účinnosti spektrálních metod. Čtvrtá kapitola pak analyzuje několik diskretizací a potvrzuje, že konečné prvky poskytují řešení s nejmenšími diskretizačími chybami. Za třetí, pojednává práce o snížení výpočetních nákladů pomocí modelování s redukovaným řádem. Pátá kapitola ukazuje potenciál a efektivitu využití tenzorů nízké hodnosti ve spektrálních metodách pro rozsáhlé problémy.

Klíčová slova: výpočetní homogenizace, metody založené na FFT, spektrální metody, metoda konečných prvků, odhady vlastných čísel, předpodmínění pomocí diskrétního Greenova operátoru

Contents

https://doi.org/10.14311/dis.fsv.2022.007

Chapter 1

Introduction

Many natural bodies such as bamboo and bone exhibit excellent strength and durability despite having rather low density [156]. Their extraordinary macroscopic mechanical properties arise from efficient distribution of the bulk material across scales. In similar manner, the non-linear macroscopic behavior of concrete structures is determined by mechanical properties of constituents and their geometrical distribution on microscale. Besides the analysis of existing structures, the design of microstructures becomes an interesting topic as additive manufacturing technology moves to microscales [113]. Multiscale design enables the creation of architectured (meta)materials with microstructures beyond those that emerge naturally in manufacturing processes [62].

This intrinsic multiscale aspect of materials behavior creates a demand for the development of specialized scale-bridging techniques such as computational homogenization [83, 90, 38]. For structures with well-separated scales, a concept of periodic homogenization with a periodic unit cell as a representative volume element of the microstructure can be applied, and microstructure geometries can be characterized by high-resolution images (originating, e.g., from micro-computed tomography [87] or geometry-based models [138]).

Precise multiscale modeling for additive manufacturing or analysis of existing structures is one of the promising fields for future industries. To fully exploit the opportunities of multiscale structures, numerical modeling must be accurate and accessible. However, multiscale simulations that operate concurrently on micro- and macroscales remain too computationally demanding for everyday use [41]. This is caused primarily by the cost of micro-scale simulations, i.e. a numerical solution of an underlying partial differential equation (PDE) with periodic boundary conditions.

The pixel/voxel nature of microcomputed tomography and additive manufacturing processes allow us to consider microstructure geometries defined on regular grids. Conventional discretizations of micromechanical problems with high-resolution microstructures lead to systems of linear equations with millions to billions of unknowns, which favor iterative solvers over direct solvers. However, the convergence speed of iterative solvers can deteriorate with increasing system size. For high-resolution micromechanical problems, a special class of spectral iterative computational homogenization solvers has been developed. The convergence of these solvers is independent of the grid size. The grid size independence is achieved by the discrete Green's operator of problem with homogeneous reference data that is usually used as a preconditioner or projection operator. Thanks to the periodic boundary conditions and a regular discretization grid, the discrete Green's operator has a sparse representation in the Fourier space and can be efficiently applied using the fast Fourier transform (FFT) algorithm, which renders the computational complexity of spectral solvers $\mathcal{O}(n \log(n))$, where n is the number of pixels/voxels.

The FFT-based methods were pioneered by Moulinec and Suquet in the mid-1990s, in the seminal works [102, 103] that introduced with their fixed-point iterative scheme. Since then, numerous adjustments, improvements, and applications of their scheme have appeared, as comprehensively reviewed in [129, 84]. Algorithms developed over the years differ in solvers of non-/linear systems of equations, discretization approaches, or even micromechanical problem formulations.

The outstanding performance of the spectral methods is often compromised by the low accuracy of the solution fields [149]. Fourier-basis or trigonometric polynomial bases are not well suited for the solution of PDEs with discontinuous data because of their global supports [17]. Classical spectral methods that employ Fourier basis functions for approximation of solution produce undesired oscillations in solution fields that e.g. propagate even through the void regions [80]. Oscillatory solution fields prevent the precise localization of inelastic deformations that are necessary for predicting complex macroscopic phenomena such as plastic yielding or crack propagation in materials. Evolution of these, for engineering practice very important, nonlinear processes are governed by localization of inelastic deformation in meso- or microstructures. Therefore, the solution of these non-linear models is intrinsically affected by the accuracy of local solution fields.

Despite the efficiency of FFT-based methods and the fact that standard spectral solvers use only a single quadrature point/deformation gradient per pixel/voxel, computational requirements are still considerable for high-resolution microstructures. A cubic millimeter discretized on a grid with micrometer voxels consists of billions of voxels, which is a dataset hard to handle without extensive computational resources [1]. A fine discretization is crucial around material interfaces, where solution fields change rapidly. However, far from interfaces, the coarser mesh would be sufficient, and such a fine discretization is inefficient. Unfortunately, standard mesh coarsening techniques destroy the regular discretization structure that is essential for FFT-based methods. Therefore, alternative mesh coarsening techniques or model order reduction techniques for FFT-based methods are of interest.

1.1 Thesis objectives

This thesis focuses on three major research topics summarized in the following objectives.

(i) Understanding the effect of discrete Green's operator preconditioning.

(ii) Minimization of discretization artifacts of spectral methods.

(iii) Reduction of computational requirements of spectral methods.

Reaching these objectives will contribute to a deeper understanding of FFT-based methods, expand their application range, and further strengthen their role in multiscale simulations.

The first research topic discusses the effect of reference material on the convergence of iterative solvers. Iterative solvers are used to obtain solutions of systems of linear equations that arise from discretization of homogenization problems. For the symmetric and positive definite matrices, the conjugate gradient (CG) method is the method of choice; see, e.g., [82, 143, 123]. The convergence of the CG method can be affected by the distribution (clustering) of eigenvalues of the linear system matrix. Well-separated clusters of eigenvalues are favorable for the convergence rate, see, e.g., [82, 136] or [45, Section 2]. However, using finite precision arithmetic, similar types of spectra can slow down convergence; see, e.g., [93, 140, 44]. Knowing the distribution of the eigenvalues can help to better estimate the

quality of the preconditioner for the CG method. Additionally, guaranteed lower bounds on the smallest eigenvalue of the preconditioned problem give us access to accurate algebraic error estimates; see, e.g., [94].

In **Chapter 2**, we investigate the spectra of general diffusion or elasticity problems, discretized by the conforming finite element (FE) method, and preconditioned by the discrete Green's operator of the reference homogeneous problem. We propose an approach for obtaining guaranteed two-sided bounds on all individual eigenvalues. These bounds depend solely on the local coefficients, namely on their extremes over supports of the FE basis functions. We explore how the distribution of the eigenvalues depends on the choice of reference material and how this affects the number of iterations of the CG solver.

Grid-size independence is not the privilege of Galerkin discretization approach but it is observed for finite difference or collocation discretization approaches preconditioned by the discretized Green's operator. Therefore, in **Chapter 3**, we generalize our approach for eigenvalues bounds from **Chapter 2** such that it can be applied to other discretization methods. We use the assumption that the global matrix of the linear system can be obtained as a sum of local symmetric positive semidefinite matrices. In all these cases, the eigenvalue bounds depend solely on local material data and on connections between the degrees of freedom, i.e., on the properties of the discretization. We demonstrate the approach of obtaining eigenvalue bounds for the finite difference method, the stochastic Galerkin FE method, and the method of algebraic multilevel preconditioning.

The second research topic focuses on minimizing discretization artifacts. Knowing that the effect of the discrete Green's operator preconditioner is not restricted to a Fourier basis, we use standard FE basis functions with localized supports. The approximation with locally supported basis functions does not suffer from the Gibbs phenomenon. Additionally, the regular FE discretization preserves the efficient structure of the discrete Green's operator. Therefore, the FFT technique can be used to maintain quasilinear computational complexity typical for spectral homogenization methods also for FE discretizations. In **Chapter 4**, we provide a detailed discretization guideline for a discrete Green's operator preconditioned FFT-accelerated FE homogenization scheme. We generalize the approach pioneered by Schneider et al. [131] and Leuschner and Fritzen [79] and provide an alternative viewpoint based on linear algebra. Besides reducing ringing artifacts, we focus on the minimization of mesh-grids anisotropy that generates nonphysical preferential directions in the discretization. Localized deformations, e.g., cracks in the concrete, are prone to propagate in these directions. Therefore, we applied spectral methods to more general grids. In the end of **Chapter 4**, we discuss the equivalence between our displacement-based scheme and the strain-based homogenization scheme with the FE projection operator, used in the next chapter.

In **Chapter 5**, we discuss the problem of discretization artifacts in the strain-based framework of compatibility projection that considers the deformation gradient as the primary degree of freedom [75, 150, 25, 25]. We derive a formulation for the projection operator based on a general gradient stencil and test several finite-difference stencils, a least-square stencil, and a FE stencil. We observe that the only FE discretization stencil fully eliminates all ringing artifacts and delivers oscillation-free results.

The third research topic focuses on reducing computational requirements of the FFT-based methods. Despite their excellent computational efficiency, source requirements are still considerable for high-resolution microstructures. Therefore, in the last **Chapter 6**, we focus on the reduction of computational costs of FFT-based methods using model order reduction techniques. We employ low-rank tensor techniques that approximate a d-dimensional tensor by a sum of rank outer products of d vectors. For a sufficiently small rank, this data

compression can lead to a huge reduction in requirements for computer memory, e.g., the memory requirement of rank-50 approximation of microstructures with resolution 1024^3 voxels can be approximately equivalent to the requirements of full-field storage of microstructure with resolution 55^3 voxels [152]. On the series of scalar linear elliptic homogenization problems, we explore the performance of canonical polyadic, Tucker and Tensor-Train low-rank tensors format [54, 72].

This thesis compiles five manuscripts (four published and one under the second round of reviews) adapted into chapters. In addition to intensive collaboration with my supervisors, I was incorporated in three international research groups during my doctoral studies. Because of the very collaborative nature of the doctoral study, I am the first author of two manuscripts, the second author of two manuscripts, and the third author of one manuscript. Detailed descriptions of my contributions to these manuscripts are provided on the first pages of the corresponding chapters. I implemented algorithms used in this thesis to the C++-based open-source platform μSpectre [60] for efficient FFT-based continuum mesoscale modeling, and Python-based open-source library FFTHomPy [153] for numerical homogenization.

Chapter 2

Guaranteed two-sided bounds on all eigenvalues of preconditioned diffusion and elasticity problems solved by the finite element method

Abstract: *A method of estimating all eigenvalues of a preconditioned discretized scalar diffusion operator with Dirichlet boundary conditions has been recently introduced in T. Gergelits, K.-A. Mardal, B. F. Nielsen, and Z. Strakoš: Laplacian preconditioning of elliptic PDEs: Localization of the eigenvalues of the discretized operator, SIAM Journal on Numerical Analysis 57(3) (2019), 1369–1394. Motivated by this paper, we offer a slightly different approach that extends the previous results in some directions. Namely, we provide bounds on all increasingly ordered eigenvalues of a general diffusion or elasticity operator with tensor data, discretized with the conforming finite element method, and preconditioned by the inverse of a matrix of the same operator with different data. Our results hold for mixed Dirichlet and Robin or periodic boundary conditions applied to the original and preconditioning problems. The bounds are two-sided, guaranteed, easily accessible, and depend solely on the material data.*

Reproduced from:

[99] **M. Ladecký**, I. Pultarová, and J. Zeman. Guaranteed two-sided bounds on all eigenvalues of preconditioned diffusion and elasticity problems solved by the finite element method. *Applications of Mathematics*, 66(1):21–42, 2021. DOI: 10.21136/AM.2020.0217-19

My contribution:

I was involved in the numerical investigation of the algorithms, implementation of all examples, creation of all results used in the publication, revision and editing of the manuscript.

CRediT: Methodology, Software, Investigation, Visualization, Writing - Review & Editing

2.1 Introduction

In 2009, Nielsen, Tveito, and Hackbusch studied in [110] spectra of elliptic differential operators of the type $\nabla \cdot k\nabla$ defined on infinite-dimensional spaces which are preconditioned using the inverse of the Laplacian. They proved that the range of the scalar coefficient k is contained in the spectrum of the preconditioned operator, provided that k is continuous. Ten years later, Gergelits, Mardal, Nielsen, and Strakoš showed in [45] without any assumptions about the continuity of the scalar function k that there exists a one-to-one pairing between the eigenvalues of the discretized operator of the type $\nabla \cdot k\nabla$ preconditioned by the inverse of the discretized Laplacian and the intervals determined by the images under k of the supports of the conforming finite element (FE) nodal basis functions used for the discretization.

The present paper contributes to the results of [45] and generalizes some of them. While in [45], a one-to-one pairing between the eigenvalues and images of the scalar data k defined on supports of the FE basis function is proved, we introduce guaranteed two-sided bounds on all individual eigenvalues. Our approach is based on the Courant–Fischer min-max principle. Similarly as in [45], the bounds can be obtained solely from the data of the original and preconditioning problems defined on supports of the FE basis functions. While in [110] and [45] only the diffusion operator with scalar data is considered and the Laplacian operator is used for preconditioning, we treat also the diffusion operator with tensor data and with Dirichlet or Robin boundary conditions for both the original and preconditioning operators. Our theory also applies to operators with non-zero null spaces and to operators with vector valued unknown functions; as an example we study the elasticity operator with general tensor data. Any kind of conforming FE basis functions can be employed for discretization; the sets of the FE basis functions must be the same for the original and preconditioning operators. For the sake of brevity, the name preconditioning matrix (operator) will be used for the matrix $\widetilde{\mathsf{M}}$ (or operator) which is (spectrally) close to the original matrix M (or operator, respectively) rather than for the inverse of $\widetilde{\mathsf{M}}$. In contrast, in literature, including [45], $\widetilde{\mathsf{M}}^{-1}$ is often called the preconditioning matrix.

For numerical solution of sparse discretized elliptic partial differential equations, the conjugate gradient method (or Krylov subspace methods for symmetric problems, in general) is a method of choice; see, e.g., [82, 143, 123]. It is well known, that its convergence depends on distribution (clustering) of eigenvalues of the related matrices and on sizes of components of the initial residual in directions of the associated invariant subspaces. For example, well separated clusters of eigenvalues are favorable for the convergence rate, see, e.g., [82, 136] or the example in [45, Section 2]. Using finite precision arithmetic, however, similar types of the spectra can slow down the convergence; see, e.g. [93, 140] and the recent comprehensive paper [44]. Therefore, being aware of the bounds on the individual eigenvalues we can better estimate the quality of the preconditioner. Our approach can also provide guaranteed easily accessible lower bounds on the smallest eigenvalue of the preconditioned problem, which is demanded, for example, for accurate algebraic error estimates; see, e.g., [94].

The structure of the paper is as follows. In the subsequent section, we introduce the diffusion and linear elasticity equations as examples of scalar and vector valued elliptic differential equations which our approach can be applied to. In the third section, the discretization and the preconditioning setting are described. In the fourth section, the main part of the paper, we suggest a method of estimating the eigenvalues of the preconditioned matrices. The theoretical developments are accompanied with illustrative examples. Finally, we compare our method with the recent results from [45]. A short conclusion summarizes the paper.

2.2 Diffusion and elasticity problems

Our theory of estimating the eigenvalues will be applied to two frequent types of scalar and vector valued elliptic partial differential equations: the diffusion and linear elasticity equations, respectively. To this end, let us briefly introduce the associated definitions and notation; see, e.g., [14, 21, 32, 109] for further details. We assume general mixed boundary conditions for the diffusion equation, and for simplicity of exposition, homogeneous Dirichlet boundary conditions for the elasticity equation.

Let $\Omega \subset \mathbb{R}^d$ be a polygonal bounded domain, where $d = 2$ or 3. We consider the *diffusion equation* with Dirichlet and Robin boundary conditions

$$-\nabla \cdot \boldsymbol{A}\nabla u = f \text{ in } \Omega, \quad u = g_1 \text{ on } \partial\Omega_1, \quad \boldsymbol{n} \cdot \boldsymbol{A}\nabla u = g_2 - g_3 u \text{ on } \partial\Omega_2,$$

where $\partial\Omega_1$ and $\partial\Omega_2$ are two disjoint parts of the boundary $\partial\Omega$, $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$, and $\boldsymbol{n}$ denotes the outer normal to $\partial\Omega_2$. After lifting the solution u by a differentiable function u_0 that fulfills the non-homogeneous Dirichlet boundary condition and substituting $u := u + u_0$, the weak form of the new problem reads: find $u \in V = \{v \in H^1(\Omega);\ v = 0 \text{ on } \partial\Omega_1\}$ such that

$$(u, v)_A = l_{A,f}(v), \quad v \in V, \tag{2.1}$$

where

$$\begin{aligned} (u, v)_A &= \int_\Omega \nabla v \cdot \boldsymbol{A}\nabla u \,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega_2} g_3 uv \,\mathrm{d}S, \\ l_{A,f}(v) &= \int_\Omega fv \,\mathrm{d}\boldsymbol{x} - \int_\Omega \nabla v \cdot \boldsymbol{A}\nabla u_0 \,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega_2} g_2 v \,\mathrm{d}S + \int_{\partial\Omega_2} \boldsymbol{n} \cdot \boldsymbol{A}\nabla u_0 v \,\mathrm{d}S, \end{aligned}$$

for $u, v \in V$; see, e.g., [32] for details. We assume $f \in L^2(\Omega)$, $g_2 \in L^2(\partial\Omega_2)$, and $g_3 \in L^\infty(\partial\Omega_2)$, $g_3(\boldsymbol{x}) \geq 0$ on $\partial\Omega_2$. The material data $\boldsymbol{A} : \Omega \to \mathbb{R}^{d\times d}$ are assumed to be essentially bounded, i.e. $\boldsymbol{A} \in L^\infty(\Omega; \mathbb{R}^{d\times d})$, symmetric, and uniformly elliptic (positive definite) in Ω. Thus there exist constants $0 < c_A \leq C_A < \infty$ such that

$$c_A \|\boldsymbol{v}\|^2_{\mathbb{R}^d} \leq (\boldsymbol{A}(\boldsymbol{x})\boldsymbol{v}, \boldsymbol{v})_{\mathbb{R}^d} \leq C_A \|\boldsymbol{v}\|^2_{\mathbb{R}^d}, \quad \boldsymbol{x} \in \Omega,\ \boldsymbol{v} \in \mathbb{R}^d. \tag{2.2}$$

The weak form of the *linear elasticity problem* with homogeneous boundary conditions reads: find $\boldsymbol{u} \in V_0^d$, $V_0 = \{v \in H^1(\Omega);\ v = 0 \text{ on } \partial\Omega\}$, such that

$$(\boldsymbol{u}, \boldsymbol{v})_C = l_{C,F}(\boldsymbol{v}), \quad \boldsymbol{v} \in V_0^d, \tag{2.3}$$

where

$$\begin{aligned} (\boldsymbol{u}, \boldsymbol{v})_C &= \int_\Omega \sum_{i,j,k,l=1}^{d} c_{ijkl} \frac{\partial u_k}{\partial x_l} \frac{\partial v_i}{\partial x_j} \,\mathrm{d}\boldsymbol{x}, \\ l_{C,F}(\boldsymbol{v}) &= \int_\Omega \sum_{i=1}^{d} F_i v_i \,\mathrm{d}\boldsymbol{x}, \end{aligned}$$

for $\boldsymbol{u}, \boldsymbol{v} \in V_0^d$, where $\boldsymbol{F} \in (L^2(\Omega))^d$ are body forces. Due to the homogeneous Dirichlet boundary conditions on $\partial\Omega_1 = \partial\Omega$, we use the special notation V_0 of the solution space. Let

$$\tau_{ij} = \sum_{k,l=1}^{d} c_{ijkl}\, e_{kl}(\boldsymbol{u}), \quad i, j = 1, \ldots, d, \tag{2.4}$$

be the components of the Cauchy stress tensor $\boldsymbol{\tau}$ with the strain components e_{ij} obtained from the displacement vector $\boldsymbol{u}$ as

$$e_{kl}(\boldsymbol{u}) = \frac{1}{2}\left(\frac{\partial u_k}{\partial x_l} + \frac{\partial u_l}{\partial x_k}\right), \quad k,l = 1,\dots,d.$$

Assuming $d = 3$ and denoting $e_i = e_{ii}$, $i = 1,\dots,d$, we can write

$$\boldsymbol{e} = \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ 2e_{12} \\ 2e_{23} \\ 2e_{31} \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial}{\partial x_3} \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} & 0 \\ 0 & \frac{\partial}{\partial x_3} & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_3} & 0 & \frac{\partial}{\partial x_1} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \boldsymbol{\partial u}.$$

We assume that the coefficients c_{ijkl} of the tensor $\boldsymbol{c}$ in (2.4) are bounded measurable functions defined in Ω, $c_{ijkl} \in L^\infty(\Omega)$, fulfilling the symmetry conditions

$$c_{ijkl} = c_{jikl} = c_{klij}, \quad i,j,k,l = 1,\dots,d. \tag{2.5}$$

Further, we assume there exists a constant $\mu > 0$ such that

$$\mu \sum_{i,j=1}^{d} \xi_{ij}^2 \le \sum_{i,j,k,l=1}^{d} c_{ijkl}(\boldsymbol{x})\xi_{ij}\xi_{kl} \quad \text{for all symmetric tensors } \boldsymbol{\xi} \in \mathbb{R}^{d\times d},\ \boldsymbol{x} \in \Omega.$$

Assuming $d = 3$ and denoting $\tau_i := \tau_{ii}$, $i = 1,\dots,d$, due to the symmetries (2.5) of $\boldsymbol{c}$, there exist coefficients $c_{ij} \in L^\infty(\Omega)$, $i,j = 1,\dots,6$, such that the stress vector $\boldsymbol{\tau}$ can be obtained from the strain vector as

$$\boldsymbol{\tau} = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_{12} \\ \tau_{23} \\ \tau_{31} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\ c_{12} & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \\ c_{13} & c_{23} & c_{33} & c_{34} & c_{35} & c_{36} \\ c_{14} & c_{24} & c_{34} & c_{44} & c_{45} & c_{46} \\ c_{15} & c_{25} & c_{35} & c_{45} & c_{55} & c_{56} \\ c_{16} & c_{26} & c_{36} & c_{46} & c_{56} & c_{66} \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ 2e_{12} \\ 2e_{23} \\ 2e_{31} \end{pmatrix} = \boldsymbol{C}\boldsymbol{e}.$$

Starting from this place, we will use only the new set of material coefficients c_{ij}, $i,j = 1,\dots,6$, (instead of c_{ijkl}, $i,j,k,l = 1,\dots,d$) and call the associated matrix $\boldsymbol{C}$. Certain material symmetries imply special structures of $\boldsymbol{C}$. For example, homogeneous cubic 3D materials correspond to $c_{11} = c_{22} = c_{33}$, $c_{44} = c_{55} = c_{66}$, $c_{12} = c_{13} = c_{23}$, and annihilates the other components, where $c_{11} > c_{12}$, $c_{11} + 2c_{12} > 0$ and $c_{44} > 0$. Especially, for isotropic material, we have

$$c_{11} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)}, \quad c_{12} = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad c_{44} = \frac{E}{2(1+\nu)},$$

where $E > 0$ is the Young's modulus and $\nu \in (-1, \frac{1}{2})$ is the Poisson ratio [109].

The vector $\boldsymbol{F}$ of external forces fulfills

$$-\boldsymbol{\partial}^T\boldsymbol{\tau} = -\begin{pmatrix} \frac{\partial}{\partial x_1} & 0 & 0 & \frac{\partial}{\partial x_2} & 0 & \frac{\partial}{\partial x_3} \\ 0 & \frac{\partial}{\partial x_2} & 0 & \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_3} & 0 \\ 0 & 0 & \frac{\partial}{\partial x_3} & 0 & \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} \end{pmatrix} \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_{12} \\ \tau_{23} \\ \tau_{31} \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix} = \boldsymbol{F}$$

yielding

$$-\boldsymbol{\partial}^T \boldsymbol{C} \boldsymbol{\partial} \boldsymbol{u} = \boldsymbol{F}.$$

Thus $(\boldsymbol{u}, \boldsymbol{v})_C$ and $l_{C,F}(\boldsymbol{v})$ can be equivalently written as

$$\begin{aligned} (\boldsymbol{u}, \boldsymbol{v})_C &= \int_\Omega (\boldsymbol{\partial} \boldsymbol{v})^T \boldsymbol{C} \boldsymbol{\partial} \boldsymbol{u} \, \mathrm{d}\boldsymbol{x}, \\ l_{C,F}(\boldsymbol{v}) &= \int_\Omega \boldsymbol{v}^T \boldsymbol{F} \, \mathrm{d}\boldsymbol{x}. \end{aligned}$$

If $d = 2$, the dimensions of the arrays naturally reduce. For example, for cubic materials we get

$$\boldsymbol{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \quad \boldsymbol{\tau} = \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_{12} \end{pmatrix}, \quad \boldsymbol{\partial} = \begin{pmatrix} \frac{\partial}{\partial x_1} & 0 \\ 0 & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} \end{pmatrix}, \quad \boldsymbol{C} = \begin{pmatrix} c_{11} & c_{12} & 0 \\ c_{12} & c_{11} & 0 \\ 0 & 0 & c_{44} \end{pmatrix}.$$

2.3 Discretization and preconditioning

We assume that a conforming FE method is employed to discretize the diffusion and elasticity problems defined by (2.1) and (2.3), respectively. The domain Ω is thus decomposed into a finite number of elements $\mathcal{E}_j$, $j = 1, \ldots, N_e$. Some continuous FE basis functions (with compact supports) denoted by φ_k, $k = 1, \ldots, N$, are used as approximation and test functions. By $\mathcal{P}_k$ we denote the smallest patch of elements covering the support of φ_k. Correspondingly to Section 2.2, we denote the material data by $\boldsymbol{A}$ and $\boldsymbol{C}$ of the diffusion and elasticity operators, respectively, and the data of the associated preconditioning operators by $\widetilde{\boldsymbol{A}}$ and $\widetilde{\boldsymbol{C}}$, respectively. The function g_3 entering the Robin boundary conditions is allowed to be different in the original and preconditioning operators; therefore, it is denoted by $\widetilde{g}_3$ in the latter.

The stiffness matrices A and C of the systems of linear equations of the discretized problems (2.1) and (2.3), respectively, have elements

$$\mathsf{A}_{kl} = \int_\Omega \nabla \varphi_l(\boldsymbol{x}) \cdot \boldsymbol{A}(\boldsymbol{x}) \nabla \varphi_k(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} + \int_{\partial\Omega_2} g_3(\boldsymbol{x}) \varphi_l(\boldsymbol{x}) \varphi_k(\boldsymbol{x}) \, \mathrm{d}S$$

and

$$\mathsf{C}_{\boldsymbol{kl}} = \int_\Omega (\boldsymbol{\partial}(\varphi_{l_1}(\boldsymbol{x}), \ldots, \varphi_{l_d}(\boldsymbol{x}))^T)^T \boldsymbol{C}(\boldsymbol{x}) \boldsymbol{\partial}(\varphi_{k_1}(\boldsymbol{x}), \ldots, \varphi_{k_d}(\boldsymbol{x}))^T \, \mathrm{d}\boldsymbol{x}, \tag{2.6}$$

respectively, where $k, l = 1, \ldots, N$, and $\boldsymbol{k}, \boldsymbol{l} \in \{1, \ldots, N\}^d$. The preconditioning matrices $\widetilde{\mathsf{A}}$ and $\widetilde{\mathsf{C}}$ obtained for the material data $\widetilde{\boldsymbol{A}}$ and $\widetilde{\boldsymbol{C}}$, respectively, have elements

$$\widetilde{\mathsf{A}}_{kl} = \int_\Omega \nabla \varphi_l(\boldsymbol{x}) \cdot \widetilde{\boldsymbol{A}}(\boldsymbol{x}) \nabla \varphi_k(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} + \int_{\partial\Omega_2} \widetilde{g}_3(\boldsymbol{x}) \varphi_l(\boldsymbol{x}) \varphi_k(\boldsymbol{x}) \, \mathrm{d}S$$

and

$$\widetilde{\mathsf{C}}_{\boldsymbol{kl}} = \int_\Omega (\boldsymbol{\partial}(\varphi_{l_1}(\boldsymbol{x}), \ldots, \varphi_{l_d}(\boldsymbol{x}))^T)^T \widetilde{\boldsymbol{C}}(\boldsymbol{x}) \boldsymbol{\partial}(\varphi_{k_1}(\boldsymbol{x}), \ldots, \varphi_{k_d}(\boldsymbol{x}))^T \, \mathrm{d}\boldsymbol{x},$$

respectively. All integrals are supposed to be carried out exactly.

The idea of preconditioning, see, e.g. [50, Section 10.3] or [123, Chapters 9 and 10], is based on assumptions that a system of linear equations with a matrix $\widetilde{\mathsf{M}}$ is relatively easily solvable and that the spectrum of $\widetilde{\mathsf{M}}^{-1}\mathsf{M}$ is more favorable than that of M regarding some

iterative solution method, which does not necessarily mean a smaller condition number [45]. Substituting the equation $\mathsf{M}\mathsf{u} = \mathsf{B}$ with

$$\widetilde{\mathsf{M}}^{-1}\mathsf{M}\mathsf{u} = \widetilde{\mathsf{M}}^{-1}\mathsf{B} \quad \text{or} \quad \widetilde{\mathsf{M}}^{-1/2}\mathsf{M}\widetilde{\mathsf{M}}^{-1/2}\mathsf{v} = \widetilde{\mathsf{M}}^{-1/2}\mathsf{B},\ \mathsf{u} = \widetilde{\mathsf{M}}^{-1/2}\mathsf{v},$$

thus leads to equivalent problems that can be solved more efficiently than the original one.

2.4 Bounds on eigenvalues of preconditioned problems

The main results of the paper are introduced in this section. Instead of presenting our results for a general elliptic second order partial differential equation with tensor data and a vector valued unknown function $\boldsymbol{u}$, we first present our theory for the (scalar) diffusion equation with tensor data in full detail. Then we apply the same approach to the elasticity equation. The section is concluded by some general remarks mainly on relationship between our results and the recent results from [45].

2.4.1 Diffusion equation

The lower and upper bounds on the eigenvalues $0 \le \lambda_1 \le \cdots \le \lambda_N$ of $\widetilde{\mathsf{A}}^{-1}\mathsf{A}$ for any uniformly positive definite measurable data $\boldsymbol{A}, \widetilde{\boldsymbol{A}} : \Omega \to \mathbb{R}^{d\times d}$ are introduced in this part. The boundary conditions of the original and preconditioning problems may differ at most in the function g_3, i.e. instead of g_3, the function $\widetilde{g}_3$ can be used in Robin boundary condition of the preconditioning problem. We assume, however, that there exist constants $0 < c_g \le C_g < \infty$ such that

$$0 \le c_g\, \widetilde{g}_3(\boldsymbol{x}) \le g_3(\boldsymbol{x}) \le C_g\, \widetilde{g}_3(\boldsymbol{x}), \quad \boldsymbol{x} \in \partial\Omega_2.$$

Since N is the number of the FE basis functions then $\mathsf{A}, \widetilde{\mathsf{A}} \in \mathbb{R}^{N\times N}$. We now build two sequences of positive real numbers λ_k^{L} and λ_k^{U}, $k = 1, \dots, N$. Let us first set

$$\begin{aligned} \alpha_j^{\min} &= \operatorname{ess\,inf}_{\boldsymbol{x}\in\mathcal{E}_j} \lambda_{\min}\left(\widetilde{\boldsymbol{A}}^{-1}(\boldsymbol{x})\boldsymbol{A}(\boldsymbol{x})\right), \\ \alpha_j^{\max} &= \operatorname{ess\,sup}_{\boldsymbol{x}\in\mathcal{E}_j} \lambda_{\max}\left(\widetilde{\boldsymbol{A}}^{-1}(\boldsymbol{x})\boldsymbol{A}(\boldsymbol{x})\right), \end{aligned}$$

if no edge of $\mathcal{E}_j$ lies in $\partial\Omega_2$, and

$$\begin{aligned} \alpha_j^{\min} &= \min\left\{\operatorname{ess\,inf}_{\boldsymbol{x}\in\partial\Omega_2\cap\overline{\mathcal{E}}_j,\, g_3(\boldsymbol{x})\neq 0} \widetilde{g}_3^{-1}(\boldsymbol{x})g_3(\boldsymbol{x}),\ \operatorname{ess\,inf}_{\boldsymbol{x}\in\mathcal{E}_j} \lambda_{\min}\left(\widetilde{\boldsymbol{A}}^{-1}(\boldsymbol{x})\boldsymbol{A}(\boldsymbol{x})\right)\right\}, \\ \alpha_j^{\max} &= \max\left\{\operatorname{ess\,sup}_{\boldsymbol{x}\in\partial\Omega_2\cap\overline{\mathcal{E}}_j,\, g_3(\boldsymbol{x})\neq 0} \widetilde{g}_3^{-1}(\boldsymbol{x})g_3(\boldsymbol{x}),\ \operatorname{ess\,sup}_{\boldsymbol{x}\in\mathcal{E}_j} \lambda_{\max}\left(\widetilde{\boldsymbol{A}}^{-1}(\boldsymbol{x})\boldsymbol{A}(\boldsymbol{x})\right)\right\} \end{aligned}$$

if at least one edge of $\mathcal{E}_j$ lies in $\partial\Omega_2$, $j = 1, \dots, N_e$. If $\boldsymbol{A}(\boldsymbol{x})$ and $\widetilde{\boldsymbol{A}}(\boldsymbol{x})$ are element-wise constant and if g_3 and $\widetilde{g}_3$ are constant on every edge (of any element) lying in $\partial\Omega_2$, the computation of $\alpha_j^{\min}$ and $\alpha_j^{\max}$ reduces to calculating the extreme eigenvalues of $d\times d$ matrices on all individual elements $\mathcal{E}_j$, $j = 1, \dots, N_e$, and eventually comparing them with $\widetilde{g}_3^{-1}(\boldsymbol{x})g_3(\boldsymbol{x})$ on some of the attached edges. For every function φ_k, supported on the patch $\mathcal{P}_k$, let us set

$$\lambda_k^{\mathrm{L}} = \min_{\mathcal{E}_j\subset\mathcal{P}_k} \alpha_j^{\min}, \quad \lambda_k^{\mathrm{U}} = \max_{\mathcal{E}_j\subset\mathcal{P}_k} \alpha_j^{\max}, \quad j = 1, \dots, N. \tag{2.7}$$

Thus λ_k^{L} and λ_k^{U} are in the above sense the smallest and the largest, respectively, eigenvalues of $\widetilde{\boldsymbol{A}}^{-1}(\boldsymbol{x})\boldsymbol{A}(\boldsymbol{x})$ on the patch $\mathcal{P}_k$, or the extremes of $\widetilde{g}_3^{-1}g_3$ along the parts of the boundary of

$\mathcal{P}_k$ lying in $\partial\Omega_2$. After inspecting all patches, we sort the two series in (2.7) non-decreasingly. Thus we obtain two bijections

$$r, s : \{1, \dots, N\} \to \{1, \dots, N\}$$

such that

$$\lambda^{\mathrm{L}}_{r(1)} \le \lambda^{\mathrm{L}}_{r(2)} \le \cdots \le \lambda^{\mathrm{L}}_{r(N)}, \qquad \lambda^{\mathrm{U}}_{s(1)} \le \lambda^{\mathrm{U}}_{s(2)} \le \cdots \le \lambda^{\mathrm{U}}_{s(N)}. \tag{2.8}$$

Note that we could define and compute λ^{L}_k and λ^{U}_k directly without defining $\alpha^{\min}_j$ and $\alpha^{\max}_j$. However, dealing with the constants $\alpha^{\min}_j$ and $\alpha^{\max}_j$ is more algorithmically acceptable, because it allows to avoid multiple evaluation of eigenvalues of $\widetilde{\boldsymbol{A}}^{-1}\boldsymbol{A}$ on every element.

Next we prove an auxiliary lemma. Let $\sigma(\mathsf{M})$ denote the spectrum of the matrix M.

Lemma 2.1. *Let $\boldsymbol{A}(\boldsymbol{x}), \widetilde{\boldsymbol{A}}(\boldsymbol{x}) \in \mathbb{R}^{d\times d}$ be symmetric and positive definite for all $\boldsymbol{x} \in \mathcal{D} \subset \Omega$. Let there exist constants $0 < c_1 \le c_2 < \infty$ and $0 < c_3 \le c_4 < \infty$ such that*

$$\sigma(\widetilde{\boldsymbol{A}}^{-1}(\boldsymbol{x})\boldsymbol{A}(\boldsymbol{x})) \subset [c_1, c_2], \quad \boldsymbol{x} \in \mathcal{D}, \tag{2.9}$$

and

$$0 \le c_3\, \widetilde{g}_3(\boldsymbol{x}) \le g_3(\boldsymbol{x}) \le c_4\, \widetilde{g}_3(\boldsymbol{x}), \quad \boldsymbol{x} \in \partial\Omega_2 \cap \overline{\mathcal{D}}.$$

Then for $u \in H^1_0(\Omega)$ we get

$$c_1 \int_{\mathcal{D}} \nabla u \cdot \widetilde{\boldsymbol{A}} \nabla u \,\mathrm{d}\boldsymbol{x} \le \int_{\mathcal{D}} \nabla u \cdot \boldsymbol{A} \nabla u \,\mathrm{d}\boldsymbol{x} \le c_2 \int_{\mathcal{D}} \nabla u \cdot \widetilde{\boldsymbol{A}} \nabla u \,\mathrm{d}\boldsymbol{x} \tag{2.10}$$

and

$$\begin{aligned} &\min\{c_1, c_3\} \left(\int_{\mathcal{D}} \nabla u \cdot \widetilde{\boldsymbol{A}} \nabla u \,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega_2 \cap \overline{\mathcal{D}}} \widetilde{g}_3 u^2 \,\mathrm{d}S \right) \\ &\le \int_{\mathcal{D}} \nabla u \cdot \boldsymbol{A} \nabla u \,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega_2 \cap \overline{\mathcal{D}}} g_3 u^2 \,\mathrm{d}S \\ &\le \max\{c_2, c_4\} \left(\int_{\mathcal{D}} \nabla u \cdot \widetilde{\boldsymbol{A}} \nabla u \,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega_2 \cap \overline{\mathcal{D}}} \widetilde{g}_3 u^2 \,\mathrm{d}S \right). \end{aligned} \tag{2.11}$$

Proof. Since for all $\boldsymbol{v} \in \mathbb{R}^d$ and $\boldsymbol{x} \in \mathcal{D}$ it follows from (2.9) that

$$c_1\, \boldsymbol{v}^T \widetilde{\boldsymbol{A}}(\boldsymbol{x}) \boldsymbol{v} \le \boldsymbol{v}^T \boldsymbol{A}(\boldsymbol{x}) \boldsymbol{v} \le c_2\, \boldsymbol{v}^T \widetilde{\boldsymbol{A}}(\boldsymbol{x}) \boldsymbol{v},$$

we get (2.10) by setting $\boldsymbol{v} = \nabla u$ and integrating all three terms over $\mathcal{D}$. Inequalities (2.11) follow obviously using $g_3 \ge 0$. □

Now we introduce the first part of the main results of this paper.

Theorem 2.2. *Let us assume that the $(d-1)$-dimensional measure of $\partial\Omega_1$ is positive. The lower and upper bounds on the eigenvalues $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_N$ of $\widetilde{\mathsf{A}}^{-1}\mathsf{A}$ are given by* (2.8)*, i.e.,*

$$\lambda^{\mathrm{L}}_{r(k)} \le \lambda_k \le \lambda^{\mathrm{U}}_{s(k)}, \qquad k = 1, \dots, N. \tag{2.12}$$

Proof. Due to the positive measure of $\partial\Omega_1$, the matrices $\widetilde{\mathsf{A}}$ and A are positive definite. We only prove the lower bounds of (2.12); the upper bounds can be proved analogously. Due to the Courant–Fischer min-max theorem, e.g. [50, Theorem 8.1.2],

$$\lambda_k = \max_{S,\, \dim S = N-k+1} \min_{\mathsf{v} \in S,\, \mathsf{v} \ne 0} \frac{\mathsf{v}^T \mathsf{A} \mathsf{v}}{\mathsf{v}^T \widetilde{\mathsf{A}} \mathsf{v}},$$

where S denotes a subspace of $\mathbb{R}^N$. Then we have

$$\lambda_1 \;=\; \max_{S,\,\dim S=N}\ \min_{\mathsf{v}\in S,\,\mathsf{v}\neq 0} \frac{\mathsf{v}^T\mathsf{A}\mathsf{v}}{\mathsf{v}^T\widetilde{\mathsf{A}}\mathsf{v}} = \min_{\mathsf{v}\in\mathbb{R}^N,\,\mathsf{v}\neq 0} \frac{\mathsf{v}^T\mathsf{A}\mathsf{v}}{\mathsf{v}^T\widetilde{\mathsf{A}}\mathsf{v}} \geq \lambda^{\mathrm{L}}_{r(1)},$$

where the inequality follows from Lemma 2.1. Indeed, using $u=\sum_{i=1}^N \mathsf{v}_i\varphi_i$, definitions (2.7) and Lemma 2.1 with $\mathcal{D}=\Omega$, we get

$$\frac{\mathsf{v}^T\mathsf{A}\mathsf{v}}{\mathsf{v}^T\widetilde{\mathsf{A}}\mathsf{v}} = \frac{\int_\Omega \nabla u\cdot\boldsymbol{A}\nabla u\,\mathrm{d}\boldsymbol{x}+\int_{\partial\Omega_2} g_3u^2\,\mathrm{d}S}{\int_\Omega \nabla u\cdot\widetilde{\boldsymbol{A}}\nabla u\,\mathrm{d}\boldsymbol{x}+\int_{\partial\Omega_2} \widetilde{g}_3u^2\,\mathrm{d}S} \geq \min_{\mathcal{E}_j\subset\Omega}\alpha_j^{\min} = \min_{\mathcal{P}_k\subset\Omega}\lambda^{\mathrm{L}}_k = \lambda^{\mathrm{L}}_{r(1)}.$$

Then we proceed to

$$\lambda_2 \;=\; \max_{S,\,\dim S=N-1}\ \min_{\mathsf{v}\in S,\,\mathsf{v}\neq 0} \frac{\mathsf{v}^T\mathsf{A}\mathsf{v}}{\mathsf{v}^T\widetilde{\mathsf{A}}\mathsf{v}} \geq \min_{\mathsf{v}\in\mathbb{R}^N,\,\mathsf{v}\neq 0,\,\mathsf{v}_{r(1)}=0} \frac{\mathsf{v}^T\mathsf{A}\mathsf{v}}{\mathsf{v}^T\widetilde{\mathsf{A}}\mathsf{v}} \geq \lambda^{\mathrm{L}}_{r(2)},$$

where the last inequality follows from Lemma 2.1 where (due to $\mathsf{v}_{r(1)}=0$) $\mathcal{D}$ contains only the patches associated to the FE basis functions φ_j, $j\neq r(1)$,

$$\mathcal{D}=\cup_{j\in\{1,\dots,N\}\setminus\{r(1)\}}\mathcal{P}_j,$$

and from

$$\begin{aligned}\min_{\mathsf{v}\in\mathbb{R}^N,\,\mathsf{v}\neq 0,\,\mathsf{v}_{r(1)}=0} \frac{\mathsf{v}^T\mathsf{A}\mathsf{v}}{\mathsf{v}^T\widetilde{\mathsf{A}}\mathsf{v}} \;&=\; \min_{u=\sum_{i=1}^N \mathsf{v}_i\varphi_i,\,\mathsf{v}_{r(1)}=0} \frac{\int_{\mathcal{D}} \nabla u\cdot\boldsymbol{A}\nabla u\,\mathrm{d}\boldsymbol{x}+\int_{\partial\Omega_2\cap\overline{\mathcal{D}}} g_3u^2\,\mathrm{d}S}{\int_{\mathcal{D}} \nabla u\cdot\widetilde{\boldsymbol{A}}\nabla u\,\mathrm{d}\boldsymbol{x}+\int_{\partial\Omega_2\cap\overline{\mathcal{D}}} \widetilde{g}_3u^2\,\mathrm{d}S}\\ &\geq \min_{\mathcal{E}_j\subset\mathcal{D}}\alpha_j^{\min} = \min_{\mathcal{P}_k\subset\mathcal{D}}\lambda^{\mathrm{L}}_k = \lambda^{\mathrm{L}}_{r(2)}.\end{aligned}$$

We can proceed further in the same manner to get all inequalities $\lambda^{\mathrm{L}}_{r(k)}\leq\lambda_k$ of (2.12). □

In Theorem 3.2, we consider positive definite problems with homogeneous Dirichlet and/or general Robin boundary conditions (with $g_3\geq 0$). Neumann boundary condition is a special type of Robin boundary condition with $g_3=0$. In practical implementation of nonhomogeneous Dirichlet boundary conditions, the lifting function u_0 does not necessarily have to be employed. If the same non-homogeneous Dirichlet boundary conditions are considered for the original and preconditioning problems, the method of getting the lower and upper bounds (2.8) can be used unchanged. Our theory, however, does not cover the settings where the original and preconditioning problems are considered under different non-homogeneous Dirichlet boundary conditions or different functions g_2 in Robin boundary conditions, or if $\partial\Omega_1$ in the preconditioning problem does not coincide with $\partial\Omega_1$ used for the original problem.

If periodic or Neumann boundary conditions are applied along $\partial\Omega$ and if they are the same for the original and preconditioning problems, then A and $\widetilde{\mathsf{A}}$ are singular; they share the smallest eigenvalue $\lambda_1=0$ and the associated eigenvector. Then we can use the same method again to get the bounds on all of the eigenvalues of the preconditioned matrix; however, we must omit the null space of A (which is the same as the null space of $\widetilde{\mathsf{A}}$) from the respective formulas. To justify the method, we can proceed analogously as in the proof of Theorem 3.2, where the vectors v are now additionally considered fulfilling $\widetilde{\mathsf{A}}\mathsf{v}\neq 0$. Then

$$\lambda_2 \geq \min_{\mathsf{v}\in\mathbb{R}^N,\,\widetilde{\mathsf{A}}\mathsf{v}\neq 0} \frac{\mathsf{v}^T\mathsf{A}\mathsf{v}}{\mathsf{v}^T\widetilde{\mathsf{A}}\mathsf{v}} \geq \lambda^{\mathrm{L}}_{r(1)}.$$

We can proceed further, analogously to the proof of Theorem 3.2,

$$\lambda_3 \geq \min_{\mathsf{v}\in\mathbb{R}^N,\, \widetilde{\mathsf{A}}\mathsf{v}\neq\mathsf{0},\, \mathsf{v}_{r(1)}=0} \frac{\mathsf{v}^T\mathsf{A}\mathsf{v}}{\mathsf{v}^T\widetilde{\mathsf{A}}\mathsf{v}} \geq \lambda^{\mathrm{L}}_{r(2)}.$$

In this way we get $N-1$ lower bounding numbers on the non-zero eigenvalues of $\widetilde{\mathsf{A}}^{-1}\mathsf{A}$, where both A and $\widetilde{\mathsf{A}}$ are now considered restricted to the subspace of $\mathbb{R}^N$ that is orthogonal to the null space of A. Analogously, we get the upper bounds; thus finally,

$$\lambda^{\mathrm{L}}_{r(k-1)} \leq \lambda_k \leq \lambda^{\mathrm{U}}_{s(k)}, \quad k = 2, \ldots, N.$$

Let us now apply our method to some examples.

Example 2.3. Assume $d = 2$, $\Omega = (-\pi, \pi)^2$, $\partial\Omega_2 = \{\boldsymbol{x};\, x_1 = \pi\}$,

$$\boldsymbol{A}(\boldsymbol{x}) = \begin{pmatrix} 1 + 0.3\,\mathrm{sign}(\sin(x_2)) & 0.3 + 0.1\cos(x_1) \\ 0.3 + 0.1\cos(x_1) & 1 + 0.3\,\mathrm{sign}(\sin(x_2)) \end{pmatrix},$$

and a simple and a more sophisticated preconditioning operators with

$$\widetilde{\boldsymbol{A}}_1(\boldsymbol{x}) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \text{and} \quad \widetilde{\boldsymbol{A}}_2(\boldsymbol{x}) = \begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix},$$

respectively. Let us consider one of the following settings:
(a) uniform grid with piece-wise bilinear FE functions, $N = 10^2$ or 30^2, $g_3 = 0$; see Figure 2.1;
(b) uniform grid with piece-wise bilinear FE functions, periodic boundary conditions, $N = 21^2$; see Figure 2.2;
(c) nonuniform grid and triangular elements with piece-wise linear FE functions, $g_3 = \widetilde{g}_3 = 1 + x_2^2$, $N = 400$; see Figure 2.3.
The numerical experiments illustrate the results of Theorem 3.2, i.e. that the bounds on the eigenvalues are guaranteed for different types of boundary conditions. We can also notice that since $\boldsymbol{A}$ is point-wise closer to $\widetilde{\boldsymbol{A}}_2$ than to $\widetilde{\boldsymbol{A}}_1$, the spectrum of the second preconditioned problem (together with its bounds) is closer to unity than the spectrum of the problem preconditioned by using $\widetilde{\boldsymbol{A}}_1$. Note also that refining the mesh does not lead to more accurate bounds, in general. This is caused by the difference between the extreme eigenvalues of $\widetilde{\boldsymbol{A}}_i^{-1}\boldsymbol{A}$, $i = 1, 2$, on individual elements; see also Section 2.4.3.

The numbers of the CG steps needed to reduce the energy norm of the errors by the factor 10^{-9} (starting with zero initial vectors) for setting (a) with $f = 1$ in Ω are 17 and 13 for $\widetilde{\boldsymbol{A}}_1$ and $\widetilde{\boldsymbol{A}}_2$, respectively, for $N = 10^2$, and 20 and 15 for $\widetilde{\boldsymbol{A}}_1$ and $\widetilde{\boldsymbol{A}}_2$, respectively, for $N = 30^2$.

Let us emphasize that the error analysis of CG requires not only the eigenvalue distribution, but also (an estimate of) the components of the initial residual in directions of the associated eigenvectors; see, e.g., [45, Formula (2.7) and Remark 4.1]. In some cases, however, the eiegenvalue distribution can lead to a quite accurate estimate of the number of CG steps:

Example 2.4. Assume $d = 2$, $\Omega = (-\pi, \pi)^2$, the homogeneous Dirichlet boundary conditions, a uniform grid, $N = 18^2$, and bilinear FE functions. Let Ω_1 and Ω_2 be two small subdomains in Ω (each covering four elements). Let $\boldsymbol{A}(\boldsymbol{x}) = b(\boldsymbol{x})\boldsymbol{I}$, where

$$b(\boldsymbol{x}) = 1 + z,\ \boldsymbol{x} \in \Omega_1, \quad b(\boldsymbol{x}) = 1 - z,\ \boldsymbol{x} \in \Omega_2, \quad b(\boldsymbol{x}) = 1,\ \boldsymbol{x} \in \Omega \setminus (\Omega_1 \cup \Omega_2), \tag{2.13}$$

where z is some constant in $(-1, 1)$. For preconditioning we use Laplacian, i.e. $\widetilde{\boldsymbol{A}} = \boldsymbol{I}$. In Figure 2.4, it is seen that the spectrum of $\widetilde{\mathsf{A}}^{-1}\mathsf{A}$ contains only a few outlying eigenvalues; the

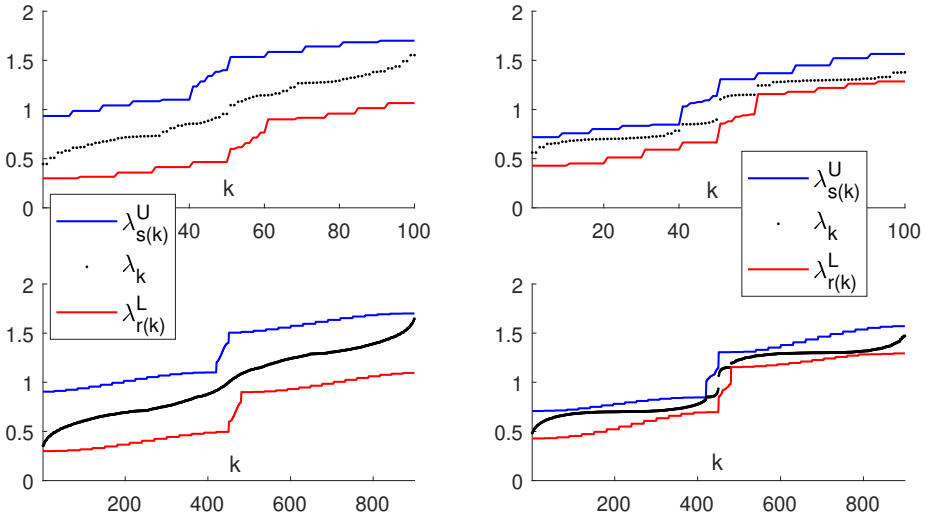


Figure 2.1: Lower ($\lambda^{\mathrm{L}}_{r(k)}$) and upper ($\lambda^{\mathrm{U}}_{s(k)}$) bounds on eigenvalues λ_k of Example 2.3 (a) with $N = 10^2$ (top graphs) and $N = 30^2$ (bottom graphs) preconditioned by operators with $\widetilde{\boldsymbol{A}}_1$ (left) and $\widetilde{\boldsymbol{A}}_2$ (right).

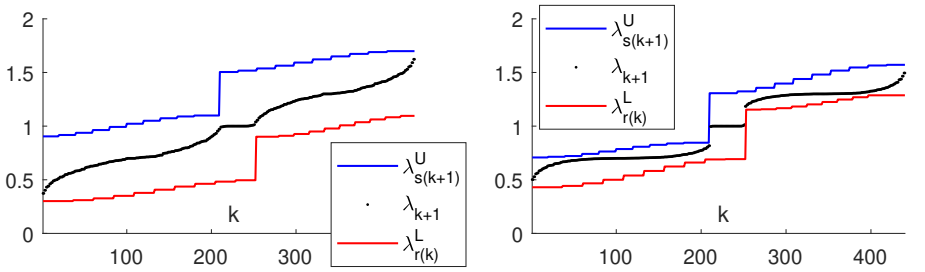


Figure 2.2: Lower ($\lambda^{\mathrm{L}}_{r(k)}$) and upper ($\lambda^{\mathrm{U}}_{s(k)}$) bounds on eigenvalues λ_k of Example 2.3 (b) with $N = 21^2$ preconditioned by operators with $\widetilde{\boldsymbol{A}}_1$ (left) and $\widetilde{\boldsymbol{A}}_2$ (right).

number of them does not depend on z. In accordance with this, the number of CG steps to reduce the energy norm of the error by the factor 10^{-9} is constant (equal to 11) independently of $z \in [0.9, 0.999]$. Note that such a z yields the condition numbers of $\widetilde{\mathsf{A}}^{-1}\mathsf{A}$ varying from 19 to 1999.

2.4.2 Elasticity equation

In the elasticity problem, or in vector valued problems in general, the searched function has multiple components, $\boldsymbol{u}(\boldsymbol{x}) = (u_1(\boldsymbol{x}), \ldots, u_d(\boldsymbol{x}))^T$, where individual components are coupled within the equation. For approximation of the scalar functions u_j, $j = 1, \ldots, d$, we use the same sets of the FE basis functions φ_k, $k = 1, \ldots, N$, supported again inside the patches $\mathcal{P}_k$. Recall that for the sake of simplicity, we consider homogeneous Dirichlet boundary conditions only.

Lemma 2.5. *Let $\boldsymbol{C}(\boldsymbol{x}), \widetilde{\boldsymbol{C}}(\boldsymbol{x}) \in \mathbb{R}^{m\times m}$, where $m = 3$ if $d = 2$, and $m = 6$ if $d = 3$. Let $\boldsymbol{C}$ and $\widetilde{\boldsymbol{C}}$ be symmetric and positive definite for all $\boldsymbol{x} \in \mathcal{D} \subset \Omega$. Let there exist constants*

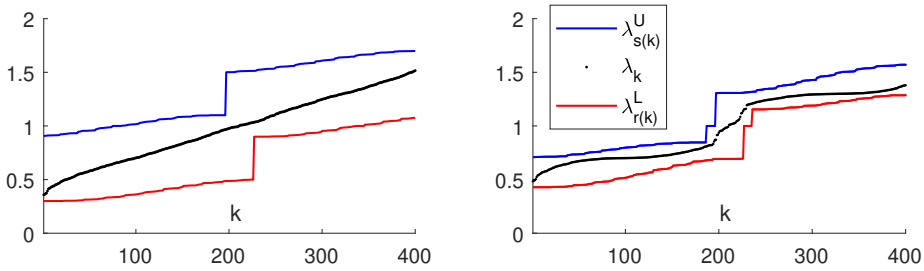


Figure 2.3: Lower $(\lambda^{\mathrm{L}}_{r(k)})$ and upper $(\lambda^{\mathrm{U}}_{s(k)})$ bounds on eigenvalues λ_k of Example 2.3 (c) with $N = 400$ preconditioned by operators with $\widetilde{\boldsymbol{A}}_1$ (left) and $\widetilde{\boldsymbol{A}}_2$ (right) with $g_3 = \widetilde{g}_3 = 1 + x_2^2$.

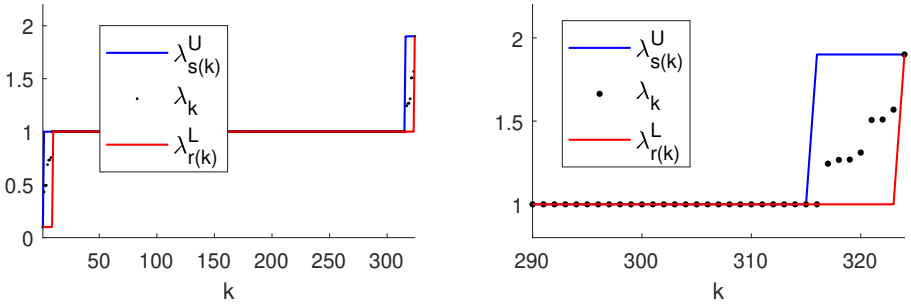


Figure 2.4: Lower $(\lambda^{\mathrm{L}}_{r(k)})$ and upper $(\lambda^{\mathrm{U}}_{s(k)})$ bounds on eigenvalues λ_k of Example 2.4 for $z = 0.9$ (left) and the detail view (right).

$0 < c_1 \le c_2 < \infty$ *such that*

$$\sigma(\widetilde{\boldsymbol{C}}^{-1}(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{x})) \subset [c_1, c_2], \quad \boldsymbol{x} \in \mathcal{D}. \tag{2.14}$$

Then for $\boldsymbol{u} \in V_0^d$ *we get*

$$c_1 \int_{\mathcal{D}} (\boldsymbol{\partial u})^T \widetilde{\boldsymbol{C}} \boldsymbol{\partial u} \,\mathrm{d}\boldsymbol{x} \le \int_{\mathcal{D}} (\boldsymbol{\partial u})^T \boldsymbol{C} \boldsymbol{\partial u} \,\mathrm{d}\boldsymbol{x} \le c_2 \int_{\mathcal{D}} (\boldsymbol{\partial u})^T \widetilde{\boldsymbol{C}} \boldsymbol{\partial u} \,\mathrm{d}\boldsymbol{x} \tag{2.15}$$

Proof. From (2.14) for all $\boldsymbol{v} \in \mathbb{R}^d$, $\boldsymbol{x} \in \mathcal{D}$, we get

$$c_1 \, \boldsymbol{v}^T \widetilde{\boldsymbol{C}}(\boldsymbol{x}) \boldsymbol{v} \le \boldsymbol{v}^T \boldsymbol{C}(\boldsymbol{x}) \boldsymbol{v} \le c_2 \, \boldsymbol{v}^T \widetilde{\boldsymbol{C}}(\boldsymbol{x}) \boldsymbol{v}.$$

Then by setting $\boldsymbol{v} = \boldsymbol{\partial u}$ and integrating over $\mathcal{D}$, we obtain (2.15). □

We now show how to obtain the guaranteed bounds on all individual eigenvalues $0 < \lambda_1 \le \cdots \le \lambda_{dN}$ of the preconditioned elasticity problem $\widetilde{\mathsf{C}}^{-1}\mathsf{C}$ for any positive definite material data $\boldsymbol{C}$ and $\widetilde{\boldsymbol{C}}$. Since N is the number of the FE basis functions defined on Ω used to approximate each component of $\boldsymbol{u}$, the number of unknowns is dN. We now build two sequences λ^{L}_k and λ^{U}_k, $k = 1, \ldots, dN$, to bound the eigenvalues of $\widetilde{\mathsf{C}}^{-1}\mathsf{C}$. In contrast to Section 2.4.1, for the sake of brevity, we do not define $\alpha_j^{\min}$ and $\alpha_j^{\max}$, but we directly set

$$\begin{aligned} \widehat{\lambda}^{\mathrm{L}}_k &= \operatorname{ess\,inf}_{\boldsymbol{x} \in \mathcal{P}_k} \lambda_{\min}\left(\widetilde{\boldsymbol{C}}^{-1}(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{x})\right), \\ \widehat{\lambda}^{\mathrm{U}}_k &= \operatorname{ess\,sup}_{\boldsymbol{x} \in \mathcal{P}_k} \lambda_{\max}\left(\widetilde{\boldsymbol{C}}^{-1}(\boldsymbol{x})\boldsymbol{C}(\boldsymbol{x})\right), \end{aligned}$$

$k = 1, \ldots, N$. Similarly to the case of the diffusion equation in Section 2.4.1, we sort these two series non-decreasingly, and thus get bijections

$$R, S : \{1, \ldots, N\} \to \{1, \ldots, N\},$$

such that

$$\widehat{\lambda}^{\mathrm{L}}_{R(1)} \le \cdots \le \widehat{\lambda}^{\mathrm{L}}_{R(N)}, \quad \widehat{\lambda}^{\mathrm{U}}_{S(1)} \le \cdots \le \widehat{\lambda}^{\mathrm{U}}_{S(N)}.$$

Moreover, we double (if $d = 2$) or triple (if $d = 3$) all items in the two series of $\widehat{\lambda}^{\mathrm{L}}_k$ and $\widehat{\lambda}^{\mathrm{U}}_k$ and get two new d-times longer series

$$\lambda^{\mathrm{L}}_{(k-1)d+1} = \cdots = \lambda^{\mathrm{L}}_{kd} = \widehat{\lambda}^{\mathrm{L}}_k, \quad \lambda^{\mathrm{U}}_{(k-1)d+1} = \cdots = \lambda^{\mathrm{U}}_{kd} = \widehat{\lambda}^{\mathrm{U}}_k, \quad k = 1, \dots, N,$$

that can be sorted non-decreasingly. Thus we obtain two bijections

$$r, s : \{1, \dots, dN\} \to \{1, \dots, dN\},$$

such that

$$\begin{aligned} \lambda^{\mathrm{L}}_{r(1)} = \cdots = \lambda^{\mathrm{L}}_{r(d)} \le \lambda^{\mathrm{L}}_{r(d+1)} = \cdots = \lambda^{\mathrm{L}}_{r(2d)} \le \dots \\ \cdots \le \lambda^{\mathrm{L}}_{r(dN-d+1)} = \cdots = \lambda^{\mathrm{L}}_{r(dN)}, \end{aligned} \tag{2.16}$$

$$\begin{aligned} \lambda^{\mathrm{U}}_{s(1)} = \cdots = \lambda^{\mathrm{U}}_{s(d)} \le \lambda^{\mathrm{U}}_{s(d+1)} = \cdots = \lambda^{\mathrm{U}}_{s(2d)} \le \dots \\ \cdots \le \lambda^{\mathrm{U}}_{s(dN-d+1)} = \cdots = \lambda^{\mathrm{U}}_{s(dN)}. \end{aligned} \tag{2.17}$$

Note that for $k = 1, \dots, N$,

$$\widehat{\lambda}^{\mathrm{L}}_{R(k)} = \lambda^{\mathrm{L}}_{r((k-1)d+1)} = \cdots = \lambda^{\mathrm{L}}_{r(kd)}, \quad \widehat{\lambda}^{\mathrm{U}}_{S(k)} = \lambda^{\mathrm{U}}_{s((k-1)d+1)} = \cdots = \lambda^{\mathrm{U}}_{s(kd)}.$$

Now we can introduce the second part of the main results of this paper.

Theorem 2.6. *The lower and upper bounds on all eigenvalues $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_{dN}$ of $\widetilde{\mathsf{C}}^{-1}\mathsf{C}$ can be obtained from* (2.16) *and* (2.17)*, namely*

$$\lambda^{\mathrm{L}}_{r(k)} \le \lambda_k \le \lambda^{\mathrm{U}}_{s(k)}, \qquad k = 1, \dots, dN. \tag{2.18}$$

Proof. The proof is similar to the proof of Theorem 3.2. By the Courant–Fischer min-max theorem,

$$\lambda_k = \max_{S,\, \dim S = dN-k+1} \min_{\mathsf{v} \in S,\, \mathsf{v} \neq 0} \frac{\mathsf{v}^T \mathsf{C} \mathsf{v}}{\mathsf{v}^T \widetilde{\mathsf{C}} \mathsf{v}}.$$

Then

$$\lambda_d \ge \cdots \ge \lambda_1 = \min_{\mathsf{v} \in \mathbb{R}^{dN},\, \mathsf{v} \neq 0} \frac{\mathsf{v}^T \mathsf{C} \mathsf{v}}{\mathsf{v}^T \widetilde{\mathsf{C}} \mathsf{v}} \ge \lambda^{\mathrm{L}}_{r(1)} = \cdots = \lambda^{\mathrm{L}}_{r(d)},$$

where the last inequality follows from Lemma 2.5. Indeed, representing the coefficients of the components of $\boldsymbol{u} = (u_1, \dots, u_d)$ with respect to the FE basis functions in a single vector $\mathsf{v} = (\mathsf{v}_{(1)}^T, \dots, \mathsf{v}_{(d)}^T)^T = (\mathsf{v}_1, \dots, \mathsf{v}_{Nd})^T$, $\mathsf{v}_{(j)} \in \mathbb{R}^N$, $j = 1, \dots, d$, we get

$$\frac{\mathsf{v}^T \mathsf{C} \mathsf{v}}{\mathsf{v}^T \widetilde{\mathsf{C}} \mathsf{v}} = \frac{\int_\Omega (\boldsymbol{\partial u})^T \boldsymbol{C} \boldsymbol{\partial u} \, \mathrm{d}\boldsymbol{x}}{\int_\Omega (\boldsymbol{\partial u})^T \widetilde{\boldsymbol{C}} \boldsymbol{\partial u} \, \mathrm{d}\boldsymbol{x}} \ge \min_{\mathcal{P}_k \subset \Omega} \widehat{\lambda}^{\mathrm{L}}_k = \widehat{\lambda}^{\mathrm{L}}_{R(1)} = \lambda^{\mathrm{L}}_{r(1)} = \cdots = \lambda^{\mathrm{L}}_{r(d)}.$$

Next, we remove $\varphi_{R(1)}$ from all d bases approximating the components of $\boldsymbol{u} = (u_1, \dots, u_d)$. Then

$$\lambda_{2d} \ge \dots \ge \lambda_{d+1} \ge \min_{\mathsf{v} \in \mathbb{R}^N,\, \mathsf{v} \neq 0,\, \mathsf{v}_{R(1)} = 0, \dots, \mathsf{v}_{(d-1)N+R(1)} = 0,} \frac{\mathsf{v}^T \mathsf{C} \mathsf{v}}{\mathsf{v}^T \widetilde{\mathsf{C}} \mathsf{v}} \ge \lambda^{\mathrm{L}}_{r(d+1)} = \cdots = \lambda^{\mathrm{L}}_{r(2d)},$$

where the last inequality follows from

$$\frac{\mathsf{v}^T\mathsf{C}\mathsf{v}}{\mathsf{v}^T\widetilde{\mathsf{C}}\mathsf{v}} = \frac{\int_{\mathcal{D}}(\boldsymbol{\partial u})^T\boldsymbol{C}\boldsymbol{\partial u}\,\mathrm{d}\boldsymbol{x}}{\int_{\mathcal{D}}(\boldsymbol{\partial u})^T\widetilde{\boldsymbol{C}}\boldsymbol{\partial u}\,\mathrm{d}\boldsymbol{x}} \geq \min_{\mathcal{P}_k\subset\mathcal{D}} \widehat{\lambda}^{\mathrm{L}}_k = \widehat{\lambda}^{\mathrm{L}}_{R(2)} = \lambda^{\mathrm{L}}_{r(d+1)} = \cdots = \lambda^{\mathrm{L}}_{r(2d)},$$

where $\mathsf{v}_{R(1)} = 0, \ldots, \mathsf{v}_{(d-1)N+R(1)} = 0$, and correspondingly,

$$\mathcal{D} = \cup_{j\in\{1,\ldots,N\}\setminus\{R(1)\}}\mathcal{P}_j.$$

Continuing further in this way, we can prove the lower bounds in (2.12). Analogously, we can get the upper bounds. □

Example 2.7. Assume the elasticity equation with homogeneous Dirichlet boundary conditions, $d = 2$, $\Omega = (-\pi, \pi)^2$, $N = 21^2$, and the data

$$\boldsymbol{C}(\boldsymbol{x}) = \frac{E(\boldsymbol{x})}{(1+\nu)(1-2\nu)}\begin{pmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & 0.5\,(1-2\nu) \end{pmatrix}, \tag{2.19}$$

where

$$E(\boldsymbol{x}) = 1 + 0.3\,\mathrm{sign}\,(x_1x_2), \quad \nu = 0.2.$$

Preconditioning is performed with the constant (homogeneous) data of the type (2.19) with $E = 1$ and either $\nu = 0$ or $\nu = 0.2$, denoted by $\widetilde{\boldsymbol{C}}_1$ and $\widetilde{\boldsymbol{C}}_2$, respectively. A uniform grid with piece-wise bilinear FE functions is employed. We can see in Figure 2.5 that the preconditioning matrix using the data $\widetilde{\boldsymbol{C}}_2$, which are closer to $\boldsymbol{C}$, yields the spectrum of the preconditioned matrix closer to unity. Moreover, we can notice two clusters of eigenvalues approximately equal to 0.7 and 1.3, respectively. The numbers of the CG steps to reduce the energy norms of the errors by the factor of 10^{-9} are 14 and 11 for $\widetilde{\boldsymbol{C}}_1$ and $\widetilde{\boldsymbol{C}}_2$, respectively, when we consider $\boldsymbol{F} = (1, 0)^T$. In this example, $\widetilde{\boldsymbol{C}}_1$ is diagonal, while $\widetilde{\boldsymbol{C}}_2$ is more dense. Therefore, the overall efficiency strongly depends on implementation of the preconditioner. These considerations are, however, behind the scope of this paper.

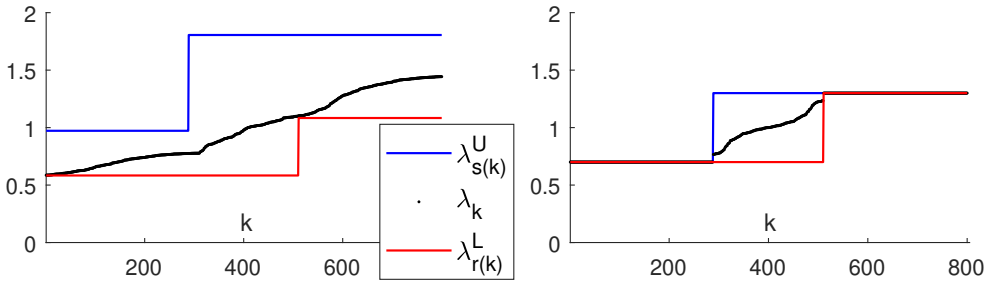


Figure 2.5: Lower ($\lambda^{\mathrm{L}}_{r(k)}$) and upper ($\lambda^{\mathrm{U}}_{s(k)}$) bounds on eigenvalues λ_k of the elasticity problem of Example 2.7 with $N = 21^2$ preconditioned by operators with $\widetilde{\boldsymbol{C}}_1$ (left) and $\widetilde{\boldsymbol{C}}_2$ (right).

Remark 2.8. The bilinear form $(\boldsymbol{u}, \boldsymbol{v})_C$ associated with the linear elasticity operator is equivalent with the following bilinear forms defined in V_0^d, see [14],

$$\begin{aligned}
(\boldsymbol{u}, \boldsymbol{v})_{C,\triangle} &= \int_\Omega \sum_{i,j=1}^{d} \frac{\partial v_i}{\partial x_j}\frac{\partial u_i}{\partial x_j}\,\mathrm{d}\boldsymbol{x} \\
(\boldsymbol{u}, \boldsymbol{v})_{C,\varepsilon} &= \int_\Omega (\boldsymbol{\partial v})^T\boldsymbol{\partial u}\,\mathrm{d}\boldsymbol{x} \\
(\boldsymbol{u}, \boldsymbol{v})_{C,d} &= \int_\Omega \sum_{i=1}^{d} (\boldsymbol{\partial}(0,\ldots,0,v_i,0,\ldots,0)^T)^T\boldsymbol{C}\boldsymbol{\partial}(0,\ldots,0,u_i,0,\ldots,0)^T,
\end{aligned}$$

where $\boldsymbol{v} = (v_1, \ldots, v_d)^T$. The equivalence constants and the proofs can be found in [14] and in the references therein. We may notice that our preconditioning matrix $\widetilde{\mathsf{C}}$ with the data in the form $\widetilde{\boldsymbol{C}}(\boldsymbol{x}) = \boldsymbol{I}$ is the same as the matrix of the discretized form $(\boldsymbol{u}, \boldsymbol{v})_{C,\varepsilon}$. Therefore, using our method for obtaining the bounds on the eigenvalues of preconditioned problems can be used to estimate the equivalence constants of the above forms defined in finite-dimensional subspaces of V_0^d spanned by the FE basis functions; for example, we can immediately get

$$\lambda^{\mathrm{L}}_{r(1)}(\boldsymbol{u}, \boldsymbol{u})_{C,\varepsilon} \leq (\boldsymbol{u}, \boldsymbol{u})_C \leq \lambda^{\mathrm{U}}_{s(dN)}(\boldsymbol{u}, \boldsymbol{u})_{C,\varepsilon}.$$

2.4.3 General remarks

Let us now compare our results obtained for the diffusion equation with the recent results from [45]. Analogies for the elasticity equation can be considered straightforwardly. In [45], the existence of a pairing between the eigenvalues of the preconditioned matrix and the intervals obtained from the scalar data defined on the patches is proved. Especially, in any of the intervals, some eigenvalue must be found. This allows us to estimate the accuracy of the bounds provided that the scalar data are continuous or mildly changing in (parts of) Ω. In our paper, instead, we get that $\lambda_k \in [\lambda^{\mathrm{L}}_{r(k)}, \lambda^{\mathrm{U}}_{s(k)}]$, or $\lambda_k \in [\lambda^{\mathrm{L}}_{r(k-1)}, \lambda^{\mathrm{U}}_{s(k)}]$ if the operator is semi-definite with the null space of the dimension 1. Let us note that

$$\lambda^{\mathrm{L}}_k \leq \lambda^{\mathrm{U}}_k, \quad \lambda^{\mathrm{L}}_{r(k)} \leq \lambda^{\mathrm{U}}_{s(k)}, \quad r(k) \leq s(k), \quad k = 1, \ldots, N,$$

but $r(k) \neq s(k)$ in general, thus the intervals containing the individual eigenvalues are different than the intervals obtained in [45]. Sometimes, however, the intervals obtained by our method and by the method of [45] (ordered appropriately) coincide; see the following example.

Example 2.9. Let us consider the test problem from [45, Section 4]: the diffusion equation, $\Omega = (0,1)^2$, $\boldsymbol{A}(\boldsymbol{x}) = \sin(x_1 + x_2)\boldsymbol{I}$, and homogeneous Dirichlet boundary conditions on $\partial\Omega$. Let us use a uniform grid with piece-wise bilinear FE functions, $N = 9^2$ or $N = 19^2$. For preconditioning we use $\widetilde{\boldsymbol{A}}(\boldsymbol{x}) = \boldsymbol{I}$. The appropriatelly ordered bounds provided by [45] and the bounds obtained by our method coincide; they are displayed on Figure 2.6.

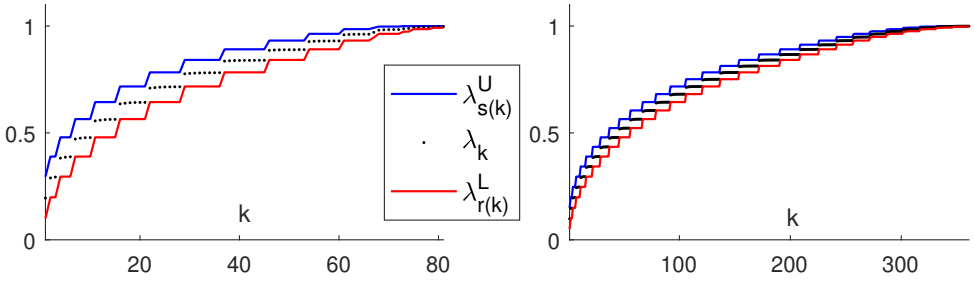


Figure 2.6: Lower ($\lambda^{\mathrm{L}}_{r(k)}$) and upper ($\lambda^{\mathrm{U}}_{s(k)}$) bounds on eigenvalues λ_k of Example 2.9 with $N = 9^2$ (left) and $N = 19^2$ (right).

The approach developed in [45] can be modified to the case of tensor data and existence of a permutation $p : \{1, \ldots, N\} \to \{1, \ldots, N\}$ can be proved, such that

$$\lambda_k \in [\lambda^{\mathrm{L}}_{p(k)}, \lambda^{\mathrm{U}}_{p(k)}], \quad k = 1, \ldots, N. \tag{2.20}$$

The Weyl's inequality (see, e.g., [133, Section 3.5]) is used in the proof in the same way as in [45]; the only change is in substituting the extremes of the scalar material data on every

patch $\mathcal{P}_j$ by the extremes of the eigenvalues of $\widetilde{\boldsymbol{A}}^{-1}(\boldsymbol{x})\boldsymbol{A}(\boldsymbol{x})$ on $\mathcal{P}_j$. Therefore, we do not provide the proof here. The bounds obtained from (2.12) and from (2.20) are compared in Example 2.11.

Using (2.20), under some special conditions, analogously to the results of [45], some eigenvalues can be identified exactly including their multiplicity. Since we do not present the proof of (2.20), let us formulate and prove this statement separately. For the sake of brevity, we formulate it for the case of the nonsingular diffusion equation with the tensor data only. Generalization to problems with vector valued unknowns is straightforward; see also Example 2.7.

Lemma 2.10. *Let there exist $c > 0$ such that $\widetilde{\boldsymbol{A}}^{-1}(\boldsymbol{x})\boldsymbol{A}(\boldsymbol{x}) = c\boldsymbol{I}$ on a union of m patches $\mathcal{D} = \cup_{k=1}^{m}\mathcal{P}_{j_k}$. Let none of the patches $\mathcal{P}_{j_k}$, $k = 1, \dots, m$, attache to $\partial\Omega_2$ where $g_3 \neq 0$, and let the patches be associated with m linearly independent FE functions $\varphi_{j_1}, \dots, \varphi_{j_m}$. Let A be nonsingular. Then c is an eigenvalue of $\widetilde{\mathsf{A}}^{-1}\mathsf{A}$ of multiplicity at least m.*

Proof. Let $\mathsf{e}^{(j)} \in \mathbb{R}^N$, $(\mathsf{e}^{(j)})_i = \delta_{ij}$, where δ_{ij} is the Kronecker delta symbol. Then for every $j = j_1, \dots, j_m$,

$$\frac{\mathsf{v}^T\mathsf{A}\mathsf{e}^{(j)}}{\mathsf{v}^T\widetilde{\mathsf{A}}\mathsf{e}^{(j)}} = \frac{\int_\Omega \nabla v \cdot \boldsymbol{A}\nabla\varphi_j \,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega_2} g_3\varphi_j v \,\mathrm{d}S}{\int_\Omega \nabla v \cdot \widetilde{\boldsymbol{A}}\nabla\varphi_j \,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega_2} \widetilde{g}_3\varphi_j v \,\mathrm{d}S} = \frac{c\int_\Omega \nabla v \cdot \boldsymbol{A}\nabla\varphi_j \,\mathrm{d}\boldsymbol{x}}{\int_\Omega \nabla v \cdot \widetilde{\boldsymbol{A}}\nabla\varphi_j \,\mathrm{d}\boldsymbol{x}} = c$$

for all $\mathsf{v} \in \mathbb{R}^N$, $\mathsf{v} \neq \mathsf{0}$. This means that c is an eigenvalue of $\widetilde{\mathsf{A}}^{-1}\mathsf{A}$ associated with the eigenvectors $\mathsf{e}^{(j)}$, $j = j_1, \dots, j_m$. Since the eigenvectors are linearly independent, the multiplicity of c is at least m. □

Example 2.11. In this example, we compare our method of estimating the eigenvalues of $\widetilde{\mathsf{A}}^{-1}\mathsf{A}$ with the method of [45] adapted for tensor data. Especially, we compare the bounds (2.12) with the intervals (2.20). Since we do not know the permutation p, we order the intervals according to the permutation r given by (2.8). Let us consider $d = 2$, $\Omega = (-1, 1)^2$, $N = 18^2$, and bilinear FE basis functions. Let

$$\boldsymbol{A}(\boldsymbol{x}) = \begin{pmatrix} 1.2 + 0.5\,(1 + \operatorname{sign}(\mathrm{x}_1))\,\mathrm{x}_1 & 0 \\ 0 & 1.1 - 0.5\,(1 + \operatorname{sign}(\mathrm{x}_2))\,\mathrm{x}_2 \end{pmatrix},$$

and us use $\widetilde{\boldsymbol{A}} = \boldsymbol{I}$ for preconditioning. The eigenvalues of $\widetilde{\mathsf{A}}^{-1}\mathsf{A}$ and their bounds are displayed in Figure 2.7. The guaranteed bounds (2.12) are found on the left, while the guaranteed (unordered) intervals from (2.20) are displayed on the right. In this example, the bounds do not provide sharp localization of the eigenvalues (left). The intervals, however, provide very sharp localization of a half of the spectrum (right).

Let us finally focus on limitations of our theory. We could see that in some examples the bounds did not get closer to the true eigenvalues when the mesh-size decreases. As a representative 2D example we can take the diffusion equation with constant data, preconditioned by the Laplacian, say,

$$\boldsymbol{A} = \operatorname{diag}(2, 1), \quad \widetilde{\boldsymbol{A}} = \operatorname{diag}(1, 1). \tag{2.21}$$

While the constant lower and upper bounds are obtained

$$\lambda_k^{\mathrm{L}} = 1, \quad \lambda_k^{\mathrm{U}} = 2, \quad k = 1, \dots, N,$$

the true eigenvalues of $\widetilde{\mathsf{A}}^{-1}\mathsf{A}$ are distributed between these two bounds almost achieving both extremes 1 and 2. We could conclude that if the data are of the tensor type and if the preconditioner is poor, i.e. $\widetilde{\boldsymbol{A}}^{-1}(\boldsymbol{x})\boldsymbol{A}(\boldsymbol{x})$ is not close enough to a multiple of the identity $\boldsymbol{I}$

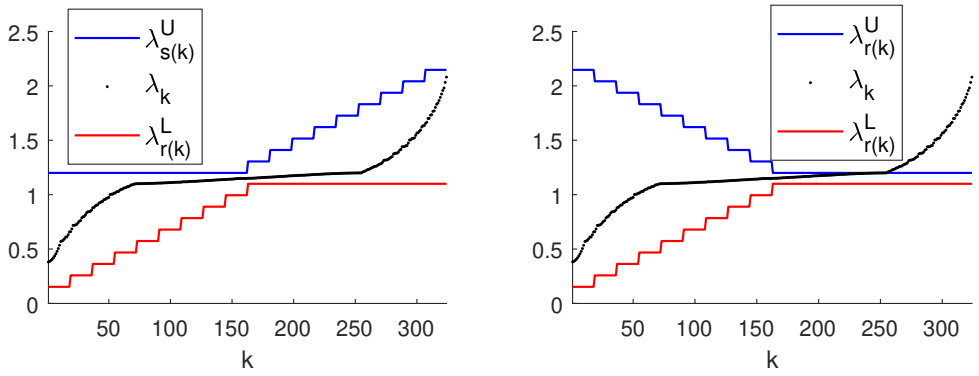


Figure 2.7: Lower ($\lambda^{\mathrm{L}}_{r(k)}$) and upper ($\lambda^{\mathrm{U}}_{s(k)}$) bounds on eigenvalues λ_k of Example 2.11 (left) and intervals $[\lambda^{\mathrm{L}}_{r(k)}, \lambda^{\mathrm{U}}_{r(k)}]$ (right).

in Ω, the bounds $\lambda^{\mathrm{L}}_{r(k)}$ and $\lambda^{\mathrm{U}}_{s(k)}$ may not say much about the true eigenvalues; the types of the FE basis functions and of the mesh influence the distribution of the true eigenvalues as well. Interestingly, from very recent results of Gergelits et al. [46] we can conclude that the spectrum of the operator $\triangle^{-1}[\nabla \cdot (\boldsymbol{A}\nabla)]$, i.e. the continuous form of example (2.21), is equal to $[1, 2]$. We hope that further study elucidates a relationship between the eigenvalues of $\widetilde{\mathsf{A}}^{-1}\mathsf{A}$ and the continuous case.

2.5 Conclusion

To the best of our knowledge, [45] is the first paper on estimating all eigenvalues of a preconditioned discretized diffusion operator. Motivated by [45], we complement to this theory by introducing another approach based on the Courant–Fisher min-max principle. This allows generalizing some of the results of [45] to vector valued equations with tensor data and with more general boundary conditions preconditioned by arbitrary operators of the same type. We provide guaranteed bounds (defined by (2.8) and by (2.16)–(2.17) for scalar and vector problems, respectively) to every particular eigenvalue. On the other hand, the approach of [45] can provide more accurate estimates of (parts of) the spectra in general. Analogously to [45], the bounds are easily accessible and obtained solely from the data defined on supports of the FE basis functions. If the data are element-wise constant, only $O(N)$ arithmetic operations and sorting of two series of N numbers must be performed. Although we applied our method to only two types of differential equations, we are convinced that the same approach can be used in a wide variety of problems.

Chapter 3

Two-sided guaranteed bounds to individual eigenvalues of preconditioned finite element and finite difference problem

Abstract: *Numerical methods for elliptic partial differential equations usually lead to systems of linear equations with sparse, symmetric and positive definite matrices. In many methods, these matrices can be obtained as sums of local symmetric positive semi-definite matrices. In this paper, we use this assumption and introduce a method which provides guaranteed lower and upper bounds to all individual eigenvalues of the preconditioned matrices. We apply the method for preconditioners arising from the same discretization problem but with simplified coefficients. The method uses solely the data over the solution domain and local connections between the degrees of freedom defined by the discretization.*

Reproduced from:

[118] I. Pultarová and **M. Ladecký**. Two-sided guaranteed bounds to individual eigenvalues of preconditioned finite element and finite difference problems. *Numerical Linear Algebra with Applications*, 28(5):e2382, 2021. DOI: 10.1002/nla.2382

My contribution:

I was involved in the numerical investigation of the algorithms, implementation of examples, creation of results used in the publication, revision and editing of the manuscript.

CRediT: Methodology, Software, Investigation, Visualization, Writing - Review & Editing

3.1 Introduction

Many numerical methods for partial differential equations (PDEs) lead to the solution of systems of linear equations. Elliptic PDEs can yield sparse symmetric positive definite matrices that can be obtained as sums of sparse symmetric positive semi-definite matrices which only use the data of a small piece of the solution domain. Such matrices are sometimes called element matrices in the context of the finite element method (FEM). We shall call them local matrices throughout the paper. For example, in FEM, the entries of such a local matrix can be defined as energy inner products of pairs of all basis functions over a single element. Similarly, the matrices resulting from the stochastic Galerkin finite element method (SGFEM) or from the finite difference method (FDM) can also be obtained as sums of symmetric positive semi-definite matrices with only a few non-zero entries. Another common feature of these discretization methods is that the conditioning of the matrices becomes worse if the grid is getting finer and/or the scales of magnitudes of varying data of the underlying problems are growing. Therefore, in practical computation, some kind of preconditioning is to be applied. In this paper, we consider a preconditioning matrix obtained by the same discretization method as the original system matrix but with different data or using a different operator. Then we introduce a method providing guaranteed two-sided bounds to all individual eigenvalues of the preconditioned system or, equivalently, to all generalized eigenvalues of a system defined by these two matrices. The bounds are based solely on comparing local data of the problem and of the preconditioning problem and on local connections among the degrees of freedom (DOFs). Coefficients of the preconditioning operator can be constant, for example, or admit some other advantageous features which allow an efficient solution method. For example, discretized diffusion problems can be preconditioned by the discretized Laplacian, and linear elasticity stiffness matrices can be preconditioned by a discretized homogeneous, possibly anisotropic elasticity operator. This is efficient if (the approximation of) the inverse of the preconditioning matrix can be easily obtained, for example, by using the discrete Fourier transformation, or if the LU-decomposition of the preconditioning matrix can be obtained and applied to many problems with variable coefficients.

For the numerical solution of sparse discretized elliptic PDEs, the conjugate gradient (CG) method is a method of choice; see e.g. [82, 93, 140, 143]. Typically, preconditioners leading to small condition numbers (ratios of the largest and the smallest eigenvalues) are searched for. However, the localization of the whole spectrum can help to better characterize the convergence in some particular cases. For example, a few large outlying eigenvalues can be linked to fast convergence. A detailed exposition is provided in recent paper [44] where, assuming exact arithmetic, the so called composite bounds are used to demonstrate the annihilation of the influence of the outlying eigenvalues. In finite precision arithmetic, well separated outlying eigenvalues may mimic tight clusters of eigenvalues, and thus cause a delay of convergence of CG. Nevertheless, even when outlying clusters are present, the rounding errors may not spoil the computation during a sufficient number of steps of CG; see the example in [45, Section 2].

The idea of estimating generalized eigenvalues based on local properties of the underlying operators first appeared in [110], where the eigenvalues of elliptic differential operators of the $\nabla \cdot k\nabla$ type defined in infinite-dimensional spaces and preconditioned by the Laplacian were shown to coincide with the data k. Analogous statements characterizing the spectra of problems discretized by FEM were introduced in [45, 46] where a one-to-one mapping between the eigenvalues and a set of intervals defined by the data over the elements was proved. In algebraic multilevel (AML) preconditioning, a similar idea of inspecting all individual elements

for obtaining the lower and upper bounds to the minimal and maximal eigenvalues, respectively, was used [3, 15, 31, 57]. This motivated the spectral estimates in the preconditioning of SGFEM, see e.g. [23, 74, 116, 117]. In this paper, we extract the core idea of the approach presented in [99] and show that it can also be applied to some other discretization schemes: SGFEM, FDM and AML. In all cases, the eigenvalue bounds depend solely on local material data and on local properties of the discretization. The method is formulated in a general way, and thus can be applied to other discretized problems.

The outline of the paper is as follows. In the subsequent section, we introduce the method of getting guaranteed two-sided bounds to all eigenvalues of a preconditioned matrix, or, equivalently, to generalized eigenvalues of the system matrix with respect to the preconditioning matrix. Though we focus on problems arising from discretized PDEs, the method is formulated in a general way. In the third section, we present five frequent discretization methods applied to some standard problems and show what specific forms of the general estimation method (what choices of local matrices) can provide reasonable bounds. A short discussion concludes the paper.

3.2 Two sided bounds to all eigenvalues of a preconditioned discretized problem

We assume that the stiffness matrix $\mathsf{A} \in \mathbb{R}^{N_{\mathrm{dof}} \times N_{\mathrm{dof}}}$ of a discretized problem is symmetric and positive definite, and

$$\mathsf{A} = \sum_{n=1}^{N_{\mathrm{e}}} \mathsf{A}_n, \tag{3.1}$$

where $\mathsf{A}_n \in \mathbb{R}^{N_{\mathrm{dof}} \times N_{\mathrm{dof}}}$, $n = 1, \ldots, N_{\mathrm{e}}$, are symmetric positive semi-definite (local) matrices. The number of DOFs of the discretized problem is N_{dof}. When we use FEM, the set of matrices A_n can correspond to the construction of A element-by-element. In such a case, N_{e} equals the number of elements, and the non-zero entries of every matrix A_n are only in the cross-sections of the rows and columns attached to the basis functions supported in the n-th element. The following two notations will appear as useful: let S_n, $n = 1, \ldots, N_{\mathrm{e}}$, be sets of indices of non-zero rows (columns) of A_n, and let E_j, $j = 1, \ldots, N_{\mathrm{dof}}$, denote a set of such indices n of $\{1, \ldots, N_{\mathrm{e}}\}$ that $j \in S_n$. Again, if we use FEM, then S_n is a set of DOFs attached to the FE basis functions supported in n-th element. On the other hand, E_j is a set of element numbers, where the j-th basis function is supported. Let us denote the m-th column of the $N_{\mathrm{dof}} \times N_{\mathrm{dof}}$ identity matrix by $\mathsf{e}^{(m)}$.

Let $\mathsf{A}^{\mathrm{p}} \in \mathbb{R}^{N_{\mathrm{dof}} \times N_{\mathrm{dof}}}$ be a preconditioning matrix,

$$\mathsf{A}^{\mathrm{p}} = \sum_{n=1}^{N_{\mathrm{e}}} \mathsf{A}^{\mathrm{p}}_n, \tag{3.2}$$

where $\mathsf{A}^{\mathrm{p}}_n \in \mathbb{R}^{N_{\mathrm{dof}} \times N_{\mathrm{dof}}}$, $n = 1, \ldots, N_{\mathrm{e}}$, are symmetric positive semi-definite matrices. We assume that the sets S_n and E_j constructed for matrices $\mathsf{A}^{\mathrm{p}}_n$ are the same as those built for A_n. We also assume that the kernel of A_n is the same as the kernel of $\mathsf{A}^{\mathrm{p}}_n$ for all $n = 1, \ldots, N_{\mathrm{e}}$. This assumption is not necessary for the main theorem of this section, but it is substantial for practical application of the theory, since it allows getting sensible bounds to the eigenvalues of a preconditioned matrix. Note also, that we cannot obtain the same kernels for local matrices obtained from stiffness and from mass (or identity) matrices in FEM.

Throughout paper, a formula F valid for all variables v in the set S can be denoted as $F(v)$, $v \in S$, i.e. without using "for all" or $\forall$. Similarly, $F(n)$, $n = 1, \dots, m$, means that F is valid for all n in the set $\{1, \dots, m\}$. In many places, however, we consider as helpful to use the quantifiers explicitly.

We will study the bounds to the generalized eigenvalues of

$$\mathsf{A}\mathsf{u} = \lambda\, \mathsf{A}^{\mathrm{p}}\mathsf{u}, \tag{3.3}$$

or, equivalently, to the eigenvalues of the preconditioned matrix $(\mathsf{A}^{\mathrm{p}})^{-1}\mathsf{A}$. Let us denote these eigenvalues by

$$0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_{N_{\mathrm{dof}}}.$$

If the boundary conditions are periodic or homogeneous Robin, we get one or more smallest eigenvalues of A and A^{p} equal to zero and the kernels of A and A^{p} are equal. In such a case, we can still apply our method with a small restriction. Instead of the inverse of A^{p}, we can use the pseudo-inverse $(\mathsf{A}^{\mathrm{p}})^{\#}$ [50]; and we can search for the bounds to (nonzero) eigenvalues of $(\mathsf{A}^{\mathrm{p}})^{\#}\mathsf{A}$ by the restriction of $\mathbb{R}^{N_{\mathrm{dof}} \times N_{\mathrm{dof}}}$ to the space orthogonal to the kernel of A. For simplicity, however, we avoid singular matrices A and A^{p} in our exposition. We start with a theorem providing criteria for identifying exact eigenvalues of $(\mathsf{A}^{\mathrm{p}})^{-1}\mathsf{A}$.

Theorem 3.1. *Let there exist J pair-wise different indices $m_j \in \{1, 2, \dots, N_{\mathrm{dof}}\}$, $j = 1, \dots, J$, and a constant $\beta > 0$ such that*

$$\mathsf{A}_n = \beta\, \mathsf{A}^{\mathrm{p}}_n$$

for all $n \in \cup_{j=1}^{J} E_{m_j}$. Then β is a J-tuple eigenvalue of $(\mathsf{A}^{\mathrm{p}})^{-1}\mathsf{A}$, or more precisely,

$$\mathsf{A}\mathsf{e}^{(m_j)} = \beta\, \mathsf{A}^{\mathrm{p}}\mathsf{e}^{(m_j)}, \quad j = 1, \dots, J.$$

Proof. Let $j \in \{1, \dots, J\}$ be arbitrary. Then

$$\mathsf{A}\mathsf{e}^{(m_j)} = \sum_{n=1}^{N_{\mathrm{e}}} \mathsf{A}_n \mathsf{e}^{(m_j)} = \sum_{n \in E_{m_j}} \mathsf{A}_n \mathsf{e}^{(m_j)} = \sum_{n \in E_{m_j}} \beta\, \mathsf{A}^{\mathrm{p}}_n \mathsf{e}^{(m_j)} = \beta \sum_{n=1}^{N_{\mathrm{e}}} \mathsf{A}^{\mathrm{p}}_n \mathsf{e}^{(m_j)} = \beta\, \mathsf{A}^{\mathrm{p}} \mathsf{e}^{(m_j)}.$$

Since $\mathsf{e}^{(m_j)}$ are linearly independent for $j = 1, \dots, J$, the proof is completed. □

Now we proceed with the main theorem of the paper providing guaranteed lower and upper bounds to all individual eigenvalues of $(\mathsf{A}^{\mathrm{p}})^{-1}\mathsf{A}$.

Theorem 3.2. *Let matrices A and A^{p} be symmetric and positive definite and fulfill* (3.1) *and* (3.2). *Let us define for $k = 1, \dots, N_{\mathrm{dof}}$,*

$$\lambda^{\mathrm{L}}_k = \max \left\{ \lambda;\ \mathsf{v}^T \mathsf{A}_n \mathsf{v} \ge \lambda\, \mathsf{v}^T \mathsf{A}^{\mathrm{p}}_n \mathsf{v},\ n = 1, \dots, N_{\mathrm{e}},\ \mathsf{v} \in \mathbb{R}^{N_{\mathrm{dof}}},\ \mathsf{v}_j = 0 \text{ for all } j \in T^{\mathrm{L}}_{k-1} \right\}, \tag{3.4}$$

where T^{L}_k, $k = 0, 1, \dots, N_{\mathrm{dof}} - 1$, are built consecutively: $T^{\mathrm{L}}_0 = \emptyset$ and $T^{\mathrm{L}}_k = T^{\mathrm{L}}_{k-1} \cup \{m_k\}$, where m_k is a single (arbitrary) integer such that the maximum in (3.4) *is achieved for $n = n_k$ and*

$$m_k \in S_{n_k} \setminus T^{\mathrm{L}}_{k-1}.$$

Analogously, let

$$\lambda^{\mathrm{U}}_{N_{\mathrm{dof}}-k+1} = \min \left\{ \lambda;\ \mathsf{v}^T \mathsf{A}_n \mathsf{v} \le \lambda\, \mathsf{v}^T \mathsf{A}^{\mathrm{p}}_n \mathsf{v},\ n = 1, \dots, N_{\mathrm{e}},\ \mathsf{v} \in \mathbb{R}^{N_{\mathrm{dof}}},\ \mathsf{v}_j = 0 \text{ for all } j \in T^{\mathrm{U}}_{k-1} \right\}, \tag{3.5}$$

where $T_0^{\mathrm{U}} = \emptyset$ *and* $T_k^{\mathrm{U}} = T_{k-1}^{\mathrm{U}} \cup \{m_k\}$, *where* m_k *is an integer such that the minimum in* (3.5) *is achieved for* $n = n_k$ *and*

$$m_k \in S_{n_k} \setminus T_{k-1}^{\mathrm{U}}.$$

Then

$$\lambda_k^{\mathrm{L}} \le \lambda_k \le \lambda_k^{\mathrm{U}}, \quad k = 1, \dots, N_{\mathrm{dof}}. \tag{3.6}$$

Proof. We have

$$\lambda_1 = \min_{\mathsf{v} \neq 0} \frac{\mathsf{v}^T \mathsf{A} \mathsf{v}}{\mathsf{v}^T \mathsf{A}^{\mathrm{p}} \mathsf{v}},$$

or, equivalently,

$$\lambda_1 = \max\{\lambda;\ \mathsf{v}^T \mathsf{A} \mathsf{v} \ge \lambda \mathsf{v}^T \mathsf{A}^{\mathrm{p}} \mathsf{v},\ \forall\, \mathsf{v} \in \mathbb{R}^{N_{\mathrm{dof}}}\}.$$

Thus any $\lambda_1^{\mathrm{L}} \in \mathbb{R}$ fulfilling

$$\mathsf{v}^T \mathsf{A} \mathsf{v} \ge \lambda_1^{\mathrm{L}}\, \mathsf{v}^T \mathsf{A}^{\mathrm{p}} \mathsf{v}, \quad \forall\, \mathsf{v} \in \mathbb{R}^{N_{\mathrm{dof}}} \tag{3.7}$$

is a lower bound to λ_1. Condition (3.7) is equivalent to

$$\sum_{n=1}^{N_{\mathrm{e}}} \mathsf{v}^T \mathsf{A}_n \mathsf{v} \ge \lambda_1^{\mathrm{L}} \sum_{n=1}^{N_{\mathrm{e}}} \mathsf{v}^T \mathsf{A}_n^{\mathrm{p}} \mathsf{v}, \quad \forall\, \mathsf{v} \in \mathbb{R}^{N_{\mathrm{dof}}};$$

and a sufficient condition to (3.7) is

$$\mathsf{v}^T \mathsf{A}_n \mathsf{v} \ge \lambda_1^{\mathrm{L}}\, \mathsf{v}^T \mathsf{A}_n^{\mathrm{p}} \mathsf{v}, \quad \forall\, n = 1, \dots, N_{\mathrm{e}},\ \forall\, \mathsf{v} \in \mathbb{R}^{N_{\mathrm{dof}}}. \tag{3.8}$$

The maximal λ_1^{L} fulfilling (3.8) is equivalent to the λ_1^{L} defined by (3.4). Thus the first inequality of (3.6) is proved for $k = 1$. Let for $k = 1$ the maximum in (3.4) be achieved with $n = n_1$. Then let us choose an arbitrary but a unique $m_1 \in S_{n_1}$ and set $T_1^{\mathrm{L}} = \{m_1\}$. In other words, we exclude m_1 in some sense from the set of DOFs in our further consideration. Note that since the maximum in (3.4) is finite and is achieved for $n = n_1$, the set S_{n_1} is not empty.

Due to the Courant-Fischer min-max principle

$$\lambda_2 = \max_{\dim V = N_{\mathrm{dof}} - 1} \min_{\mathsf{v} \neq 0,\, \mathsf{v} \in V} \frac{\mathsf{v}^T \mathsf{A} \mathsf{v}}{\mathsf{v}^T \mathsf{A}^{\mathrm{p}} \mathsf{v}} \ge \min_{\mathsf{v} \neq 0,\, \mathsf{v}_{m_1} = 0} \frac{\mathsf{v}^T \mathsf{A} \mathsf{v}}{\mathsf{v}^T \mathsf{A}^{\mathrm{p}} \mathsf{v}},$$

and thus any $\lambda_2^{\mathrm{L}} \in \mathbb{R}$ such that

$$\mathsf{v}^T \mathsf{A} \mathsf{v} \ge \lambda_2^{\mathrm{L}}\, \mathsf{v}^T \mathsf{A}^{\mathrm{p}} \mathsf{v}, \quad \forall\, \mathsf{v} \in \mathbb{R}^{N_{\mathrm{dof}}},\ \mathsf{v}_{m_1} = 0,$$

or, equivalently,

$$\mathsf{v}^T \mathsf{A} \mathsf{v} \ge \lambda_2^{\mathrm{L}}\, \mathsf{v}^T \mathsf{A}^{\mathrm{p}} \mathsf{v}, \quad \forall\, \mathsf{v} \in \mathbb{R}^{N_{\mathrm{dof}}},\ \mathsf{v}_j = 0 \text{ for all } j \in T_1^{\mathrm{L}}, \tag{3.9}$$

is a lower bound to λ_2. A sufficient condition for λ_2^{L} fulfilling (3.9) is, similarly as in (3.8), obtained from inspecting all pairs of matrices A_n and $\mathsf{A}_n^{\mathrm{p}}$ separately, namely as

$$\mathsf{v}^T \mathsf{A}_n \mathsf{v} \ge \lambda_2^{\mathrm{L}}\, \mathsf{v}^T \mathsf{A}_n^{\mathrm{p}} \mathsf{v}, \quad \forall\, n = 1, \dots, N_{\mathrm{e}},\ \forall\, \mathsf{v} \in \mathbb{R}^{N_{\mathrm{dof}}} \text{ such that } \mathsf{v}_j = 0,\ \forall\, j \in T_1^{\mathrm{L}}. \tag{3.10}$$

The maximal λ_2^{L} fulfilling (3.10) equals the λ_2^{L} defined by (3.4). Thus the first inequality of (3.6) is proved for $k = 2$. Then we choose such an index n_2 that the maximum in (3.4) is achieved for $n = n_2$. Since the maximum is finite, the set $S_{n_2} \setminus T_1^{\mathrm{L}}$ is not empty. Then we can choose some (unique) $m_2 \in S_{n_2} \setminus T_1^{\mathrm{L}}$, and set $T_2^{\mathrm{L}} = T_1^{\mathrm{L}} \cup \{m_2\}$. We proceed in the same manner up to $\lambda_{N_{\mathrm{dof}}}^{\mathrm{L}}$, the lower bound to $\lambda_{N_{\mathrm{dof}}}$. Analogously, the upper bounds λ_k^{U} to all eigenvalues of $(\mathsf{A}^{\mathrm{p}})^{-1}\mathsf{A}$ can be obtained starting from $\lambda_{N_{\mathrm{dof}}}^{\mathrm{U}}$ and finishing with λ_1^{U}. □

The findings of Theorem 3.2 can be summarized as algorithms. A practical implementation of the algorithms is based on effective computing the extreme λ's in (3.4). The extremes can be obtained by inspecting all pairs of local matrices A_n and $\mathsf{A}_n^{\mathrm{p}}$ separately, namely, by solving very small eigenvalue problems. Moreover, since A_n and $\mathsf{A}_n^{\mathrm{p}}$ use only local data, these extremes can be obtained by inspecting (element-by-element) the data of the original problem and of the preconditioning operator. More details are presented in section 3 for some particular problems and discretization methods.

Algorithm 1 (lower bounds).
1. Set T as an empty set.
2. For $k = 1, \dots, N_{\mathrm{dof}}$ repeat (a)–(c):
(a) Find λ_k^{L} as the maximal λ fulfilling inequalities $\mathsf{v}^T\mathsf{A}_n\mathsf{v} \geq \lambda\, \mathsf{v}^T\mathsf{A}_n^{\mathrm{p}}\mathsf{v}$ for all $n = 1, \dots, N_{\mathrm{e}}$, and for all $\mathsf{v} \in \mathbb{R}^{N_{\mathrm{dof}}}$ such that $\mathsf{v}_j = 0$ for all $j \in T$.
(b) Take an arbitrary index n for which the equality in (a) is achieved with $\lambda = \lambda_k^{\mathrm{L}}$ and $\mathsf{v} \neq 0$.
(c) Take an arbitrary $m \in S_n \setminus T$, and set $T := T \cup \{m\}$.

Algorithm 2 (upper bounds).
1. Set T as an empty set.
2. For $k = 1, \dots, N_{\mathrm{dof}}$ repeat (a)–(c):
(a) Find $\lambda_{N_{\mathrm{dof}}-k+1}^{\mathrm{U}}$ as the minimal λ fulfilling inequalities $\mathsf{v}^T\mathsf{A}_n\mathsf{v} \leq \lambda \mathsf{v}^T\mathsf{A}_n^{\mathrm{p}}\mathsf{v}$ for all $n = 1, \dots, N_{\mathrm{e}}$, and for all $\mathsf{v} \in \mathbb{R}^{N_{\mathrm{dof}}}$ such that $\mathsf{v}_j = 0$ for all $j \in T$.
(b) Take an arbitrary index n for which the equality in (a) is achieved with $\lambda = \lambda_k^{\mathrm{U}}$ and $\mathsf{v} \neq 0$.
(c) Take an arbitrary $m \in S_n \setminus T$, and set $T := T \cup \{m\}$.

Note that while inspecting all pairs of matrices A_n and $\mathsf{A}_n^{\mathrm{p}}$, $n = 1, \dots, N_{\mathrm{dof}}$, for some $k > 1$ in steps 2(a), we can exclude such matrices A_n and $\mathsf{A}_n^{\mathrm{p}}$ in which $S_n \subset T$, because $\mathsf{A}_n\mathsf{v} = \mathsf{A}_n^{\mathrm{p}}\mathsf{v} = 0$ if $\mathsf{v}_j = 0$ for all $j \in S_n$.

Algorithms 1 and 2 are formulated generally, and thus can be applied to various PDEs and discretization methods in different ways. The details of the algorithms, especially the choice of local matrices A_n and $\mathsf{A}_n^{\mathrm{p}}$, are essential for the practical implementation. In the subsequent section, we introduce five frequent problems and discretization methods and show some particular choices of A_n and $\mathsf{A}_n^{\mathrm{p}}$ that can lead to acceptable bounds in practical computation. However, new choices of local matrices and/or new strategies for choosing n in step 2(b) and m in step 2(c) leading to more accurate bounds can be found in a future study.

3.3 Five model problems

We present five model problems of discretized PDEs. In all problems, the PDE is defined in the physical domain $D \subset \mathbb{R}^2$. The shape of D is considered polygonal, or even rectangular if FDM is used. One of these problems is parametrized by a (random) parameter; see section 3.3.4. The PDEs considered are either the heat equation or linear elasticity equation with Dirichlet or Robin boundary conditions. The discretization is obtained via FEM, FDM or SGFEM for the parametrized PDE. The preconditioning matrices are obtained in the same manner as the stiffness matrices of the underlying problem but for different data or for a slightly different operator. A different operator is used in AML or SGFEM methods, where we consider hierarchical finite elements and associated preconditioning and a truncation based operator, respectively. In FEM based discretizations, we consider the material data as constant element-wise.

In all cases, a second order elliptic differential equation is considered, either reformulated into a weak form and discretized, or discretized directly using FDM. Both schemes lead to systems of linear equations with symmetric a positive definite matrices A. Our concern is to

construct a preconditioning matrix A^{p} and to give guaranteed lower and upper bounds to all eigenvalues of the resulting preconditioned matrix $(\mathsf{A}^{\mathrm{p}})^{-1}\mathsf{A}$. In other words, we search for the bounds to all particular eigenvalues of the generalized eigenvalue problem

$$\mathsf{A}\mathsf{v} = \lambda\, \mathsf{A}^{\mathrm{p}}\mathsf{v}. \tag{3.11}$$

In the subsequent parts we introduce five model problems illustrating the application of Theorems 3.1 and 3.2 and of Algorithms 1 and 2. In all five model problems, we focus on the principles of our new algorithms, therefore the problems are simple without any complicated geometry or mesh, and with a relatively small number of DOFs. Then we can calculate all the eigenvalues and their bounds almost exactly; considering rounding errors is beyond the scope of this paper.

3.3.1 Finite element method and heat equation

The second order scalar elliptic differential equation

$$-\nabla\cdot(\boldsymbol{a}(\boldsymbol{x})\nabla u(\boldsymbol{x})) = f(\boldsymbol{x}), \quad \boldsymbol{x}\in D, \tag{3.12}$$

is considered with Dirichlet or Robin boundary conditions on $\partial D = \partial D_{\mathrm{D}} \cup \partial D_{\mathrm{R}}$,

$$\begin{aligned} u(\boldsymbol{x}) &= g_1(\boldsymbol{x}), \quad \boldsymbol{x}\in\partial D_{\mathrm{D}} \\ \boldsymbol{n}(\boldsymbol{x})\cdot(\boldsymbol{a}(\boldsymbol{x})\nabla u(\boldsymbol{x})) &= g_2(\boldsymbol{x}) - g_3(\boldsymbol{x})u(\boldsymbol{x}), \quad \boldsymbol{x}\in\partial D_{\mathrm{R}}, \end{aligned}$$

where $\boldsymbol{n}$ is the outer normal to ∂D_{R}. The coefficient tensor $\boldsymbol{a}$ defined on $\overline{D}$ is uniformly positive definite, measurable and uniformly bounded over $\overline{D}$, and $f\in L^2(D)$. Problem (3.12) is transformed into the weak form and discretized using FEM with continuous and piecewise polynomial basis functions ϕ_i, $i=1,\dots,N_{\mathrm{dof}}$; see e.g. [32]. This leads to a system of linear equations with a symmetric and positive definite matrix A. Our concern is to build a preconditioning matrix A^{p} and to find bounds to all eigenvalues of the resulting preconditioned matrix $(\mathsf{A}^{\mathrm{p}})^{-1}\mathsf{A}$. We assume that the preconditioning matrix A^{p} is obtained by the discretization of the operator

$$-\nabla\cdot(\boldsymbol{a}^{\mathrm{p}}(\boldsymbol{x})\nabla u(\boldsymbol{x})), \tag{3.13}$$

with the same type of boundary conditions as those of the original problem. We note that the boundary conditions can be considered different in building A and A^{p}. This only leads to a small modification of the algorithm; see a more detailed description in [99]. The coefficient tensor $\boldsymbol{a}^{\mathrm{p}}$ is positive definite, measurable and bounded uniformly in $\overline{D}$.

It is well known that the entries of A are obtained as energy scalar products

$$\mathsf{A}_{ij} = \int_D \nabla\phi_j\cdot\boldsymbol{a}\nabla\phi_i\,\mathrm{d}\boldsymbol{x} + \int_{\partial D_{\mathrm{R}}} g_3 uv\,\mathrm{d}\boldsymbol{x}, \quad i,j=1,\dots,N_{\mathrm{dof}}. \tag{3.14}$$

The entries of A^{p} are obtained in the same manner as those of A but with a coefficient tensor $\boldsymbol{a}^{\mathrm{p}}$,

$$\mathsf{A}^{\mathrm{p}}_{ij} = \int_D \nabla\phi_j\cdot\boldsymbol{a}^{\mathrm{p}}\,\nabla\phi_i\,\mathrm{d}\boldsymbol{x} + \int_{\partial D_{\mathrm{R}}} g_3 uv\,\mathrm{d}\boldsymbol{x}, \quad i,j=1,\dots,N_{\mathrm{dof}}.$$

The local matrices A_n in (3.1) and, analogously, $\mathsf{A}^{\mathrm{p}}_n$ in (3.2) are obtained as

$$(\mathsf{A}_n)_{ij} = \int_{D_n} \nabla\phi_j\cdot\boldsymbol{a}\nabla\phi_i\,\mathrm{d}\boldsymbol{x} + \int_{\overline{D_k}\cap\partial D_{\mathrm{R}}} g_3 uv\,\mathrm{d}\boldsymbol{x}, \quad i,j=1,\dots,N_{\mathrm{dof}},\ n=1,\dots,N_{\mathrm{e}}, \tag{3.15}$$

and

$$(\mathsf{A}_n^{\mathrm{p}})_{ij} = \int_{D_n} \nabla\phi_j \cdot \boldsymbol{a}^{\mathrm{p}} \nabla\phi_i \,\mathrm{d}\boldsymbol{x} + \int_{\overline{D_k}\cap\partial D_{\mathrm{R}}} g_3 uv \,\mathrm{d}\boldsymbol{x}, \quad i,j = 1,\dots,N_{\mathrm{dof}},\ n = 1,\dots,N_{\mathrm{e}}, \tag{3.16}$$

respectively, where $D_n \subset D$, $n = 1,\dots,N_{\mathrm{e}}$, are the elements defined by FEM. Then A_n and $\mathsf{A}_n^{\mathrm{p}}$ are sparse with only a few non-zero entries in the rows and columns indexed by such DOFs i associated with basis functions ϕ_i that do not annihilate in D_n.

Let us now briefly describe how to efficiently use Algorithm 1 for this particular problem and discretization. For simplicity of exposition, let us consider $g_1 = g_3 = 0$. Before starting Algorithm 1, let us define $\lambda^{\min}(\boldsymbol{x})$ as the smallest eigenvalue of $(\boldsymbol{a}^{\mathrm{p}}(\boldsymbol{x}))^{-1}\boldsymbol{a}(\boldsymbol{x})$ for almost all $\boldsymbol{x} \in \overline{D}$. Let us go through all elements D_n and set

$$\alpha_n^{\mathrm{L}} = \operatorname{ess\,inf}\left\{\lambda^{\min}(\boldsymbol{x});\ \boldsymbol{x} \in D_n\right\}.$$

If $\boldsymbol{a}$ and $\boldsymbol{a}^{\mathrm{p}}$ were element-wise constant, α_n^{L} is simply the smallest eigenvalue of $(\boldsymbol{a}^{\mathrm{p}}(\boldsymbol{x}))^{-1}\boldsymbol{a}(\boldsymbol{x})$ for arbitrary $\boldsymbol{x} \in D_n$. Using this setting, we can easily find such a constant λ that fulfills

$$\mathsf{v}^T\mathsf{A}_n\mathsf{v} \geq \lambda\,\mathsf{v}^T\mathsf{A}_n^{\mathrm{p}}\mathsf{v}, \quad \forall\, n = 1,\dots,N_{\mathrm{e}},\ \forall\,\mathsf{v} \in \mathbb{R}^{N_{\mathrm{dof}}}. \tag{3.17}$$

Indeed, since (3.17) is equivalent to

$$\int_{D_n} \nabla v \cdot \boldsymbol{a} \nabla v \,\mathrm{d}\boldsymbol{x} \geq \lambda \int_{D_n} \nabla v \cdot \boldsymbol{a}^{\mathrm{p}} \nabla v \,\mathrm{d}\boldsymbol{x}, \quad \forall\, n = 1,\dots,N_{\mathrm{e}},\ \forall\, v \in V_{\mathrm{FEM}},$$

where $V_{\mathrm{FEM}} = \operatorname{span}\{\phi_1,\dots,\phi_{N_{\mathrm{dof}}}\}$, it is enough to set

$$\lambda_1^{\mathrm{L}} = \min_{n=1,\dots,N_{\mathrm{e}}} \alpha_n^{\mathrm{L}}.$$

Then we choose some element, say D_{n_1}, where the minimum is achieved, i.e. $\lambda_1^{\mathrm{L}} = \alpha_{n_1}^{\mathrm{L}}$, and choose some DOF attached to this element, say m_1. Starting from now, in all used vectors $\mathsf{v} \in \mathbb{R}^{N_{\mathrm{dof}}}$ the m_1-entry annihilates (is excluded). To get λ_2^{L} we need such λ that

$$\mathsf{v}^T\mathsf{A}_n\mathsf{v} \geq \lambda\,\mathsf{v}^T\mathsf{A}_n^{\mathrm{p}}\mathsf{v}, \quad \forall\, n = 1,\dots,N_{\mathrm{e}},\ \forall\,\mathsf{v} \in \mathbb{R}^{N_{\mathrm{dof}}},\ \mathsf{v}_{m_1} = 0.$$

This is equivalent to

$$\int_{D_n} \nabla v \cdot \boldsymbol{a} \nabla v \,\mathrm{d}\boldsymbol{x} \geq \lambda \int_{D_n} \nabla v \cdot \boldsymbol{a}^{\mathrm{p}} \nabla v \,\mathrm{d}\boldsymbol{x}, \quad \forall\, n = 1,\dots,N_{\mathrm{e}},\ \forall\, v = \sum_{j=1}^{N_{\mathrm{dof}}} \mathsf{v}_j\phi_j,\ \mathsf{v}_{m_1} = 0.$$

Thus it is enough to set

$$\lambda_2^{\mathrm{L}} = \min\{\alpha_n^{\mathrm{L}};\ n = 1,\dots,N_{\mathrm{e}},\ \text{not all DOFs on } D_n \text{ are excluded}\}.$$

Now we choose n_2 such that the minimum is achieved on D_{n_2}, i.e. $\lambda_2^{\mathrm{L}} = \alpha_{n_2}^{\mathrm{L}}$. Then we choose a DOF attached to D_{n_2}, say m_2, that has not been excluded yet. We can proceed in a similar manner to get λ_3^{L}, λ_4^{L}, ... Analogously, we can also get the upper bounds λ_k^{U}.

Example 3.3. Let equation (3.12) be defined on $D = (0,1)\times(0,1)$ with

$$\boldsymbol{a}(\boldsymbol{x}) = \begin{pmatrix} 1 + 0.3\operatorname{sign}(x_2 - 0.5) & 0.3 + 0.1\operatorname{sign}(x_1 - 0.5) \\ 0.3 + 0.1\operatorname{sign}(x_1 - 0.5) & 1 + 0.3\operatorname{sign}(x_2 - 0.5) \end{pmatrix}, \tag{3.18}$$

Let us consider a homogeneous Dirichlet boundary condition $u = g_1 = 0$ on ∂D_{D}, and a Robin boundary condition

$$\boldsymbol{n} \cdot (\boldsymbol{a} \nabla u) = 0$$

(thus $g_2 = g_3 = 0$) on $\partial D_{\mathrm{R}} = \{(x_1, x_2);\, x_1 = 1,\, x_2 \in (0,1)\}$, where $\partial D = \partial D_{\mathrm{R}} \cup \partial D_{\mathrm{D}}$. Let us partition D into $N_{\mathrm{e}} = 450$ conforming triangles and consider continuous and piece-wise linear basis functions attached to $N_{\mathrm{dof}} = 210$ nodes with undefined solution values. Let us use two preconditioning matrices for the same boundary conditions and for the data

$$\boldsymbol{a}^{\mathrm{p}1}(\boldsymbol{x}) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \text{or} \quad \boldsymbol{a}^{\mathrm{p}2}(\boldsymbol{x}) = \begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix}, \tag{3.19}$$

respectively. The eigenvalues λ_k, $k = 1, \ldots, N_{\mathrm{dof}}$, of $(\mathsf{A}^{\mathrm{p}})^{-1}\mathsf{A}$ as well as the lower and upper bounds λ_k^{L} and λ_k^{U}, respectively, obtained by Algorithms 1 and 2, are shown in Figure 3.1.

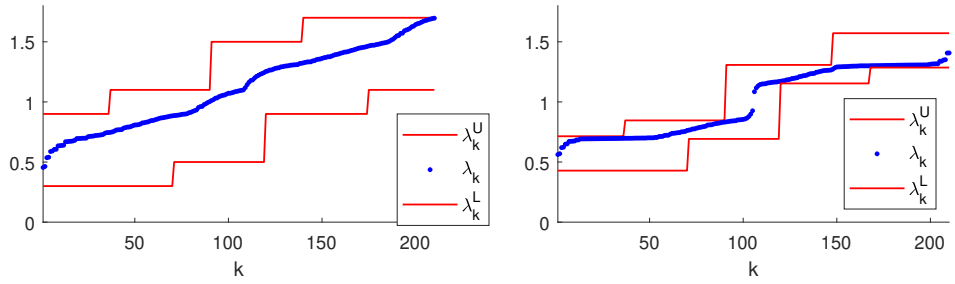


Figure 3.1: Eigenvalues of preconditioned stiffness matrices (Example 3.3) obtained by FEM (blue dots) and their lower and upper bounds (solid red lines) for preconditioners with data $\boldsymbol{a}^{\mathrm{p}1}$ (left) and $\boldsymbol{a}^{\mathrm{p}2}$ (right), respectively.

Example 3.4. The same setting and preconditioning is used as in Example 3.3 but with data

$$\boldsymbol{a}(\boldsymbol{x}) = \left(1 + 0.3 \cos\left((x_1 + x_2)\frac{\pi}{2}\right)\right) \begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix}.$$

The eigenvalues λ_k, $k = 1, \ldots, N_{\mathrm{dof}}$, of $(\mathsf{A}^{\mathrm{p}})^{-1}\mathsf{A}$ and the lower and upper bounds λ_k^{L} and λ_k^{U}, respectively, obtained by Algorithms 1 and 2, are shown in Figure 3.2.

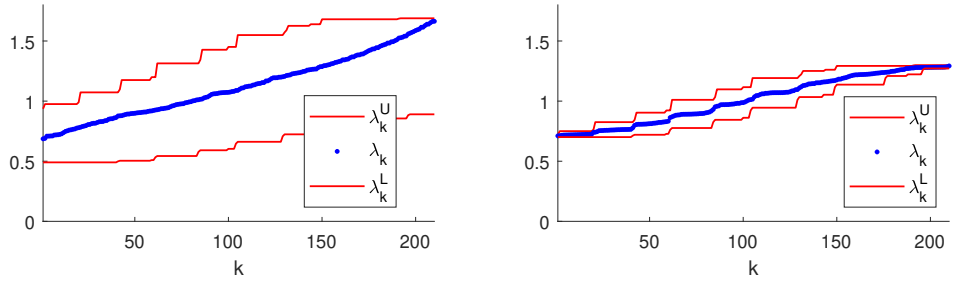


Figure 3.2: Eigenvalues of preconditioned stiffness matrices (Example 3.4) obtained by FEM (blue dots) and their lower and upper bounds (solid red lines) for preconditioners with data $\boldsymbol{a}^{\mathrm{p}1}$ (left) and $\boldsymbol{a}^{\mathrm{p}2}$ (right), respectively.

3.3.2 Finite element method and linear elasticity

The linear elasticity equation [21, 109] is a vector equation, i.e. the unknown is a vector function $\boldsymbol{u}$ defined in D. The weak form of the two-dimensional linear elasticity equation with homogeneous Dirichlet boundary conditions reads to find $\boldsymbol{u} = (u_1, u_2) \in (H_0^1(D))^2$ which fulfills

$$\int_D \boldsymbol{\partial v}^T \boldsymbol{C} \boldsymbol{\partial u} \,\mathrm{d}\boldsymbol{x} = \int_D \boldsymbol{v}^T \boldsymbol{F} \,\mathrm{d}\boldsymbol{x}, \quad \boldsymbol{\partial u} = \begin{pmatrix} \frac{\partial u_1}{\partial x_1} \\ \frac{\partial u_2}{\partial x_2} \\ \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} \end{pmatrix}, \quad \boldsymbol{C} = \begin{pmatrix} c_{11} & c_{12} & 0 \\ c_{12} & c_{22} & 0 \\ 0 & 0 & c_{33} \end{pmatrix}, \tag{3.20}$$

for all $\boldsymbol{v} \in (H_0^1(D))^2$. The stiffness tensor $\boldsymbol{C} : \overline{D} \to \mathbb{R}^{3\times 3}$ is positive definite, measurable and bounded uniformly in $\overline{D}$, and $\boldsymbol{F} \in (L^2(D))^2$ is the vector of external forces. The stiffness matrix A and the preconditioning matrix A^{p} have their entries

$$\mathsf{A}_{\boldsymbol{ij}} = \int_D \boldsymbol{\partial}(\phi_{i_1}, \phi_{i_2})^T \boldsymbol{C}\, \boldsymbol{\partial}(\phi_{j_1}, \phi_{j_2}) \,\mathrm{d}\boldsymbol{x} \quad \text{and} \quad \mathsf{A}^{\mathrm{p}}_{\boldsymbol{ij}} = \int_D \boldsymbol{\partial}(\phi_{i_1}, \phi_{i_2})^T \boldsymbol{C}^{\mathrm{p}}\, \boldsymbol{\partial}(\phi_{j_1}, \phi_{j_2}) \,\mathrm{d}\boldsymbol{x},$$

respectively, $i_1, i_2, j_1, j_2 = 1, \ldots, N$, where $\boldsymbol{i} = (i_1, i_2)$, $\boldsymbol{j} = (j_1, j_2)$ and $N_{\mathrm{dof}} = 2N$. The local matrices A_n and $\mathsf{A}^{\mathrm{p}}_n$, $n = 1, \ldots, N_{\mathrm{e}}$, have their entries

$$(\mathsf{A}_n)_{\boldsymbol{ij}} = \int_{D_n} \boldsymbol{\partial}(\phi_{i_1}, \phi_{i_2})^T \boldsymbol{C}\, \boldsymbol{\partial}(\phi_{j_1}, \phi_{j_2}) \,\mathrm{d}\boldsymbol{x} \quad \text{and} \quad (\mathsf{A}^{\mathrm{p}}_n)_{\boldsymbol{ij}} = \int_{D_n} \boldsymbol{\partial}(\phi_{i_1}, \phi_{i_2})^T \boldsymbol{C}^{\mathrm{p}}\, \boldsymbol{\partial}(\phi_{j_1}, \phi_{j_2}) \,\mathrm{d}\boldsymbol{x},$$

respectively, $i_1, i_2, j_1, j_2 = 1, \ldots, N$. The lower and upper bounds to eigenvalues are then obtained directly according to Algorithms 1 and 2, respectively. See also more examples in [99] and another approach in [46].

Example 3.5. Let us consider $D = (0,1) \times (0,1)$, linear elasticity problem (3.20) and the preconditioning problem with tensor data $\boldsymbol{C}$ and $\boldsymbol{C}^{\mathrm{p}}$, respectively, where

$$\boldsymbol{C} = \frac{E(\boldsymbol{x})}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & 0.5-\nu \end{pmatrix}, \quad \boldsymbol{C}^{\mathrm{p}} = \begin{pmatrix} 1-\nu^{\mathrm{p}} & \nu^{\mathrm{p}} & 0 \\ \nu^{\mathrm{p}} & 1-\nu^{\mathrm{p}} & 0 \\ 0 & 0 & 0.5-\nu^{\mathrm{p}} \end{pmatrix},$$

$E(\boldsymbol{x}) = 1 + 0.3\,\mathrm{sign}\,(x_1 + x_2 - 1)$ and $\nu = 0.2$. The preconditioning matrix uses either data $\boldsymbol{C}^{\mathrm{p1}}$ with $\nu^{\mathrm{p}} = 0$ or $\boldsymbol{C}^{\mathrm{p2}}$ with $\nu^{\mathrm{p}} = 0.2$. The FEM discretization with bilinear basis functions yielding $N_{\mathrm{e}} = 196$ and $N_{\mathrm{dof}} = 2 \cdot 13^2 = 338$ is used. The eigenvalues of $(\mathsf{A}^{\mathrm{p}})^{-1}\mathsf{A}$ as well as their lower and upper bounds are displayed in Figure 3.3. Note that if $\boldsymbol{C}^{\mathrm{p}}$ was a multiple of $\boldsymbol{C}$ in some parts of D, some eigenvalues can be determined exactly with sharp bounds; see Theorem 3.1.

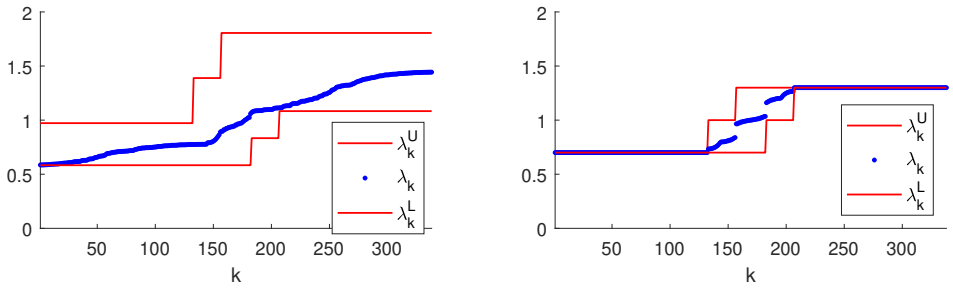


Figure 3.3: Eigenvalues of preconditioned stiffness matrices (Example 3.5) obtained by FEM for the elasticity equation (blue dots) and their lower and upper bounds (solid red lines) for a preconditioner with $\boldsymbol{C}^{\mathrm{p1}}$ (left) and $\boldsymbol{C}^{\mathrm{p2}}$ (right).

3.3.3 Algebraic multilevel preconditioning

In this part, we recall algebraic multilevel (AML) preconditioning and estimating the resulting spectrum. AML methods use nested meshes and associated hierarchical FE basis functions. In this paper, we use only two levels of hierarchy, which we call coarse and fine and denote by superscripts c and f, respectively. The details of this useful method can be found e.g. in [3, 15, 31, 57]. Here we show that the well known algorithm of estimating the spectrum also fits in the context of Algorithms 1 and 2.

We consider again second-order linear elliptic problem (3.12) defined in a polygonal domain D with a homogeneous Dirichlet boundary condition on ∂D. We use FEM defining a coarse triangulation with element-wise linear basis functions ϕ_j^{c}, $j = 1, \ldots, N_{\mathrm{dof}}^{\mathrm{c}}$. We assume that $\boldsymbol{a}$ is constant on every coarse element (triangle) D_n, $n = 1, \ldots, N_{\mathrm{e}}$. Let each coarse element D_n be split into four (fine) triangles of the same shape with vertices equal either to the vertices of the coarse triangles or to the centers of the edges of the coarse triangles. We consider only such fine basis functions attached to the centers of the edges of the coarse triangles: ϕ_j^{f}, $j = 1, \ldots, N_{\mathrm{dof}}^{\mathrm{f}}$. Thus the set of all fine and coarse basis functions

$$W = \{\phi_j^{\mathrm{c}};\, j = 1, \ldots, N_{\mathrm{dof}}^{\mathrm{c}}\} \cup \{\phi_j^{\mathrm{f}};\, j = 1, \ldots, N_{\mathrm{dof}}^{\mathrm{f}}\}$$

is linearly independent. The approximation and test spaces are obtained as a span of W; thus $N_{\mathrm{dof}} = N_{\mathrm{dof}}^{\mathrm{c}} + N_{\mathrm{dof}}^{\mathrm{f}}$. The stiffness matrix $\mathsf{A} \in \mathbb{R}^{N_{\mathrm{dof}} \times N_{\mathrm{dof}}}$ can be obtained in the block form

$$\mathsf{A} = \begin{pmatrix} \mathsf{A}^{\mathrm{cc}} & \mathsf{A}^{\mathrm{cf}} \\ (\mathsf{A}^{\mathrm{cf}})^T & \mathsf{A}^{\mathrm{ff}} \end{pmatrix},$$

where the superscripts c and f are related to coarse and fine basis functions (DOFs), respectively, and $\mathsf{A}^{\mathrm{cc}} \in \mathbb{R}^{N_{\mathrm{dof}}^{\mathrm{c}} \times N_{\mathrm{dof}}^{\mathrm{c}}}$ and $\mathsf{A}^{\mathrm{ff}} \in \mathbb{R}^{N_{\mathrm{dof}}^{\mathrm{f}} \times N_{\mathrm{dof}}^{\mathrm{f}}}$. In AML methods, one of possible preconditioners can be

$$\mathsf{A}^{\mathrm{p}} = \begin{pmatrix} \mathsf{A}^{\mathrm{cc}} & 0 \\ 0 & \mathsf{A}^{\mathrm{ff}} \end{pmatrix}.$$

The matrix A can be obtained as

$$\mathsf{A} = \sum_{n=1}^{N_{\mathrm{e}}} \mathsf{A}_n$$

where A_n is a sparse local matrix with only 6×6 non-zero entries,

$$\begin{aligned} (\mathsf{A}_n^{\mathrm{cc}})_{ij} &= \int_{D_n} \nabla \phi_j^{\mathrm{c}} \cdot \boldsymbol{a} \nabla \phi_i^{\mathrm{c}} \,\mathrm{d}\boldsymbol{x} \\ (\mathsf{A}_n^{\mathrm{ff}})_{lm} &= \int_{D_n} \nabla \phi_m^{\mathrm{f}} \cdot \boldsymbol{a} \nabla \phi_l^{\mathrm{f}} \,\mathrm{d}\boldsymbol{x} \\ (\mathsf{A}_n^{\mathrm{cf}})_{lj} &= \int_{D_n} \nabla \phi_j^{\mathrm{c}} \cdot \boldsymbol{a} \nabla \phi_l^{\mathrm{f}} \,\mathrm{d}\boldsymbol{x}, \end{aligned}$$

$i, j = 1, \ldots, N_{\mathrm{dof}}^{\mathrm{c}}$, $l, m = 1, \ldots, N_{\mathrm{dof}}^{\mathrm{f}}$, $n = 1, \ldots, N_{\mathrm{e}}$.

Example 3.6. We consider AML preconditioning for problem (3.12) with a homogeneous Dirichlet boundary condition on ∂D. We have $N_{\mathrm{e}} = 392$ coarse elements D_n and $N_{\mathrm{dof}} = N_{\mathrm{dof}}^{\mathrm{c}} + N_{\mathrm{dof}}^{\mathrm{f}} = 182 + 502 = 684$. The resulting eigenvalues of $(\mathsf{A}^{\mathrm{p}})^{-1}\mathsf{A}$ and their lower and upper bounds are displayed in Figure 3.4. As a rule, the bounds are not tight in AML. Instead of estimating the individual eigenvalues, only λ_1^{L} and $\lambda_{N_{\mathrm{dof}}}^{\mathrm{U}}$ can be computed to get the upper bound to the condition number

$$\kappa\left((\mathsf{A}^{\mathrm{p}})^{-1}\mathsf{A}\right) \leq \lambda_{N_{\mathrm{dof}}}^{\mathrm{U}} / \lambda_1^{\mathrm{L}}.$$

The results in Figure 3.4 illustrate that the upper bound is sharp.

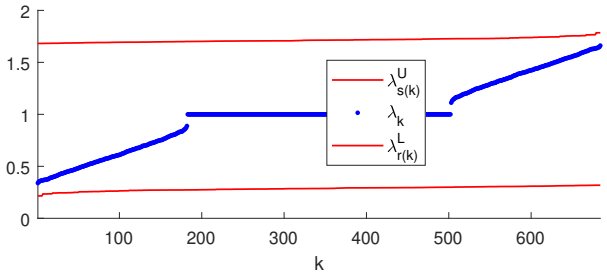


Figure 3.4: Eigenvalues of the preconditioned stiffness matrix (Example 3.6) obtained by FEM and AML for the heat equation (blue dots) and their lower and upper bounds (solid red lines).

3.3.4 Stochastic Galerkin finite element method

We consider a stochastic (or parameter) second order scalar elliptic differential equation

$$-\nabla \cdot (a(\boldsymbol{x},\boldsymbol{\xi})\nabla u(\boldsymbol{x},\boldsymbol{\xi})) = f(\boldsymbol{x}),$$

where $\boldsymbol{x} \in D$, $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_{N_{\rm par}}) \in \Omega \subset \mathbb{R}^{N_{\rm par}}$ with Dirichlet boundary conditions on ∂D defined for all $\boldsymbol{\xi} \in \Omega$. The scalar coefficient $\boldsymbol{a}$ is uniformly bounded in $\overline{D}$ for almost all $\boldsymbol{\xi} \in \Omega$. The gradient operator is applied with respect to the variable $\boldsymbol{x}$. The weak form reads to find $u \in V = H^1_0(D) \otimes L^2_\rho(\Omega)$ such that

$$\int_\Omega \int_D \nabla v(\boldsymbol{x},\boldsymbol{\xi}) \cdot (a(\boldsymbol{x},\boldsymbol{\xi})\nabla u(\boldsymbol{x},\boldsymbol{\xi}))\, \rho(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{\xi} = \int_\Omega \int_D f(\boldsymbol{x}) v(\boldsymbol{x},\boldsymbol{\xi})\, \rho(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{\xi}, \quad v \in V,$$

where $L^2_\rho(\Omega)$ is the Lebesgue space with the positive weight function ρ defined on $\Omega \subset \mathbb{R}^{N_{\rm par}}$. The discretization by SGFEM [5, 27, 47, 157] yields a space $V_{\rm dof}$ which is a span of the products of $N^{\rm fe}_{\rm dof}$ finite element basis functions $\phi_i(\boldsymbol{x})$, $i = 1, \ldots, N^{\rm fe}_{\rm dof}$, defined in D, with $N^{\rm pol}_{\rm dof}$ polynomials $\psi_j(\boldsymbol{\xi})$, $j = 1, \ldots, N^{\rm pol}_{\rm dof}$, defined in Ω. Thus $N_{\rm dof} = N^{\rm fe}_{\rm dof} N^{\rm pol}_{\rm dof}$. The matrix of the discretized problem reads

$$\mathsf{A}_{(i-1)N+r,\,(j-1)N+s} = \int_\Omega \psi_s(\boldsymbol{\xi})\psi_r(\boldsymbol{\xi}) \int_D \nabla\phi_j(\boldsymbol{x}) \cdot (a(\boldsymbol{x},\boldsymbol{\xi})\nabla\phi_i(\boldsymbol{x}))\, \rho(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{\xi},$$

$i, j = 1, \ldots, N^{\rm fe}_{\rm dof}$, $r, s = 1, \ldots, N^{\rm pol}_{\rm dof}$.

Preconditioning can be applied in various manners. Here we examine truncated based preconditioning [11, 23**?** , 116, 117, 139] where the matrix $\mathsf{A}^{\rm P}$ is built using fewer terms of expansion of $a(\boldsymbol{x},\boldsymbol{\xi})$ than in the original operator. Let us see the following example which is very simple, still showing all essential principles of estimating the spectra of the preconditioned matrices for SGFEM.

Example 3.7. Let $D = (0,1)$, $N_{\rm par} = 1$, $\Omega = \mathbb{R}$, $\rho = \mathrm{e}^{-x^2/2}$, thus the distribution of the parameter ξ can be considered as random with the Gaussian probability density function ρ (up to a scaling factor). Let

$$a(x,\xi) = a_0(x) + \xi a_1(x), \quad a_0(x) = 4 + \mathrm{sign}(x - 0.3), \quad a_1(x) = 0.2.$$

Let us consider homogeneous Dirichlet boundary conditions for any $\xi \in \Omega$. Let $N^{\rm fe}_{\rm dof} = 21$, and either $N^{\rm pol}_{\rm dof} = 5$, or $N^{\rm pol}_{\rm dof} = 15$, then either $N_{\rm dof} = 105$ or $N_{\rm dof} = 315$, respectively. Then

$$\mathsf{A} = \mathsf{A}^{(0)} + \mathsf{A}^{(1)},$$

where

$$\begin{aligned}
\mathsf{A}^{(0)}_{(i-1)N+r,\,(j-1)N+s} &= \int_\Omega \psi_s(\xi)\psi_r(\xi)\,\rho(\xi)\,\mathrm{d}\xi \int_D \nabla\phi_j(x)\cdot(a_0(x)\nabla\phi_i(x))\,\mathrm{d}x,\\
\mathsf{A}^{(1)}_{(i-1)N+r,\,(j-1)N+s} &= \int_\Omega \xi\psi_s(\xi)\psi_r(\xi)\,\rho(\xi)\,\mathrm{d}\xi \int_D \nabla\phi_j(x)\cdot(a_1(x)\nabla\phi_i(x))\,\mathrm{d}x.
\end{aligned}$$

Let the preconditioning matrix be $\mathsf{A}^{\mathrm{p}} = \mathsf{A}^{(0)}$. Then we can apply Algorithms 1 and 2 to get the bounds to the eigenvalues of $(\mathsf{A}^{\mathrm{p}})^{-1}\mathsf{A}$ in such a manner that the local matrices A_n and $\mathsf{A}^{\mathrm{p}}_n$ are

$$\begin{aligned}
(\mathsf{A}^{(0)}_n)_{(i-1)N+r,\,(j-1)N+s} &= \int_\Omega \psi_s(\xi)\psi_r(\xi)\,\rho(\xi)\,\mathrm{d}\xi \int_{D_n} \nabla\phi_j(x)\cdot(a_0(x)\nabla\phi_i(x))\,\mathrm{d}x,\\
(\mathsf{A}^{(1)}_n)_{(i-1)N+r,\,(j-1)N+s} &= \int_\Omega \xi\psi_s(\xi)\psi_r(\xi)\,\rho(\xi)\,\mathrm{d}\xi \int_{D_n} \nabla\phi_j(x)\cdot(a_1(x)\nabla\phi_i(x))\,\mathrm{d}x,
\end{aligned}$$

where $D_n = ((n-1)h, nh)$, $n = 1,\dots,N^{\mathrm{fe}}_{\mathrm{dof}}+1$, $h = 1/(N^{\mathrm{fe}}_{\mathrm{dof}}+1)$. Thus $N_{\mathrm{e}} = N^{\mathrm{fe}}_{\mathrm{dof}}+1$. The resulting bounds as well as the spectra of $(\mathsf{A}^{\mathrm{p}})^{-1}\mathsf{A}$ are displayed in Figure 3.5. We can notice that in both examples the maximal eigenvalues equal their upper bounds and the minimal eigenvalues equal their lower bounds. Therefore, a sharp upper bound to the condition number $\kappa \le \lambda^{\mathrm{U}}_{N_{\mathrm{dof}}}/\lambda^{\mathrm{L}}_1$ can be obtained from only the first steps of Algorithms 1 and 2; cf. also [**?**].

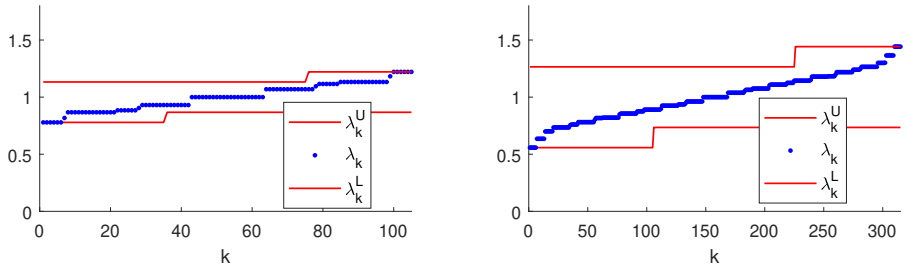


Figure 3.5: Eigenvalues of preconditioned stiffness matrices (Example 3.7) obtained by SGFEM (blue dots) and their lower and upper bounds (solid red lines) for $N^{\mathrm{pol}}_{\mathrm{dof}} = 5$ (left) and $N^{\mathrm{pol}}_{\mathrm{dof}} = 15$ (right).

3.3.5 Finite difference method

A lot of FD schemes can be found in literature; see e.g. [76] and the references therein. The formulae for substituting mixed derivatives in problems with variable and anisotropic data can be found e.g. in [12, 20, 63, 78, 81, 124, 144]. Special schemes yielding symmetric matrices are e.g. in [130]. In this part, we consider $D = (0,1)\times(0,1)$ and a uniform rectangular mesh with $N_{\mathrm{dof}} = N_1N_2$ inner nodes $\boldsymbol{x}_{ij}$, $i = 1,\dots,N_1$, $j = 1,\dots,N_2$, and second-order linear elliptic problem (3.12) with a homogeneous Dirichlet boundary condition. Thus we have N_{dof} unknown function values (DOFs)

$$u_{ij} = u(\boldsymbol{x}_{ij}), \quad \boldsymbol{x}_{ij} = (ih_1, jh_2), \quad i = 1,\dots,N_1,\ j = 1,\dots,N_2,$$

where $h_1 = 1/(N_1+1)$, $h_2 = 1/(N_2+1)$. We shall use such difference schemes which lead to a symmetric system matrix:

$$\begin{aligned}
\frac{\partial}{\partial x_1}\left(c\frac{\partial u}{\partial x_1}\right)(\boldsymbol{x}_{ij}) &\approx \frac{(c_{i-1,j}+c_{ij})u_{i-1,j} - (c_{i-1,j}+2c_{i,j}+c_{i+1,j})u_{i,j} + (c_{i,j}+c_{i+1,j})u_{i+1,j}}{2h_1^2}\\
\frac{\partial}{\partial x_2}\left(c\frac{\partial u}{\partial x_2}\right)(\boldsymbol{x}_{ij}) &\approx \frac{(c_{i,j-1}+c_{ij})u_{i,j-1} - (c_{i,j-1}+2c_{i,j}+c_{i,j+1})u_{i,j} + (c_{i,j}+c_{i,j+1})u_{i,j+1}}{2h_2^2}
\end{aligned}$$

where $c_{ij} = c(\boldsymbol{x}_{ij})$, $i = 1, \ldots, N_1$, $j = 1, \ldots, N_2$. Mixed derivatives are replaced by

$$\begin{aligned} & \left(\frac{\partial}{\partial x_1} \left(c \frac{\partial u}{\partial x_2} \right) + \frac{\partial}{\partial x_2} \left(c \frac{\partial u}{\partial x_1} \right) \right) (\boldsymbol{x}_{ij}) \\ \approx \quad & \frac{1}{4h_1h_2}((u_{i-1,j-1}(c_{i,j} + c_{i-1,j-1}) + u_{i+1,j+1}(c_{ij} + c_{i+1,j+1}) \\ & -u_{i-1,j+1}(c_{ij} + c_{i-1,j+1}) - u_{i+1,j-1}(c_{ij} + c_{i+1,j-1}) \\ & +u_{ij}(c_{i-1,j+1} + c_{i+1,j-1} - c_{i+1,j+1} - c_{i-1,j-1})). \end{aligned}$$

It can be shown that the system matrix A can be built as a sum of $N_\mathrm{e} = (N_1 + 1)(N_2 + 1)$ sparse local matrices $\mathsf{A}_n \in \mathbb{R}^{N_\mathrm{dof} \times N_\mathrm{dof}}$, each with at most 4×4 non-zero entries in positions attached to 2×2 neighboring unknowns (DOFs) in the nodes $\{\boldsymbol{x}_{ij}, \boldsymbol{x}_{i+1,j}, \boldsymbol{x}_{i,j+1}, \boldsymbol{x}_{i+1,j+1}\}$, $i = 0, 1, \ldots, N_1$, $j = 0, 1, \ldots, N_2$. There are $N_\mathrm{e} = N_1 N_2$ such 2×2 sets. Note that the function values are known at the boundary nodes and thus some matrices A_n contain fewer non-zero entries. The matrices A_n have in general the following non-zero submatrices (minors)

$$\begin{aligned} & (\mathsf{A}_n)_{\boldsymbol{r},\boldsymbol{r}} \\ & = \frac{(a_{11})_{ij} + (a_{11})_{i+1,j}}{2} \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \frac{(a_{11})_{i,j+1} + (a_{11})_{i+1,j+1}}{2} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \\ & + \frac{(a_{22})_{ij} + (a_{22})_{i,j+1}}{2} \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \frac{(a_{22})_{i+1,j} + (a_{22})_{i+1,j+1}}{2} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix} \\ & + \frac{(a_{12})_{ij} + (a_{12})_{i+1,j+1}}{2} \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix} - \frac{(a_{12})_{i+1,j} + (a_{12})_{i,j+1}}{2} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \end{aligned}$$

where the vector $\boldsymbol{r}$ contains four indices corresponding to the DOFs attached to nodes $\boldsymbol{x}_{ij}$, $\boldsymbol{x}_{i+1,j}$, $\boldsymbol{x}_{i,j+1}$, $\boldsymbol{x}_{i+1,j+1}$ in this order. If some of these sets of 2×2 nodes contains one or more boundary nodes, the corresponding matrix A_n has fewer non-zero entries: 2×2 or even only 1. The preconditioning matrix A^p and local matrices A_n^p are obtained in the same manner as A and A_n, respectively, but for different coefficient data.

Example 3.8. We consider $D = (0, 1) \times (0, 1)$, with $N_\mathrm{dof} = 13^2 = 169$ inner nodes uniformly distributed in D, homogeneous Dirichlet boundary conditions on ∂D, the coefficient function $\boldsymbol{a}(\boldsymbol{x})$ defined by (3.18), and two types of preconditioners with the coefficient functions $\boldsymbol{a}^{\mathrm{p}1}(\boldsymbol{x})$ and $\boldsymbol{a}^{\mathrm{p}2}(\boldsymbol{x})$ defined by (3.19). Note that the data of the problem and of the preconditioners are the same as in Example 3.3 up to the boundary condition. We use $N_\mathrm{e} = 14^2 = 196$. The resulting eigenvalues and their bounds are displayed in Figure 3.6.

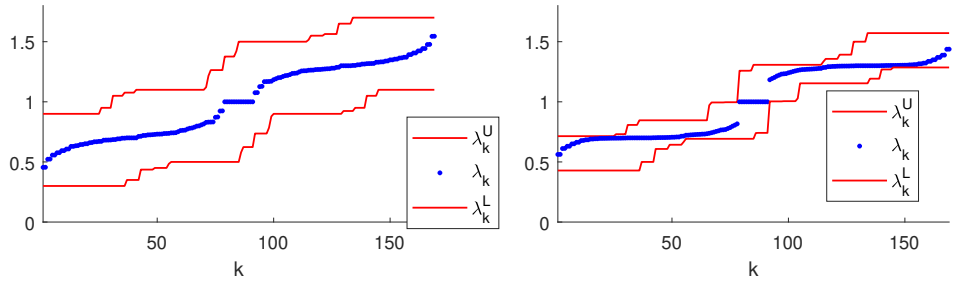


Figure 3.6: Eigenvalues of preconditioned stiffness matrices (Example 3.8) obtained by FDM (blue dots) and their lower and upper bounds (solid red lines) for preconditioners with data $\boldsymbol{a}^{\mathrm{p1}}$ (left) and $\boldsymbol{a}^{\mathrm{p2}}$ (right), respectively.

Example 3.9. Let us consider the same setting and preconditioners as in Example 3.8 but with data

$$\boldsymbol{a}(\boldsymbol{x}) = \left(1 + 0.3\cos\left((x_1 + x_2)\frac{\pi}{2}\right)\right)\begin{pmatrix} 1 & 0.3 \\ 0.3 & 1 \end{pmatrix}.$$

The resulting eigenvalues and their bounds are displayed in Figure 3.7.

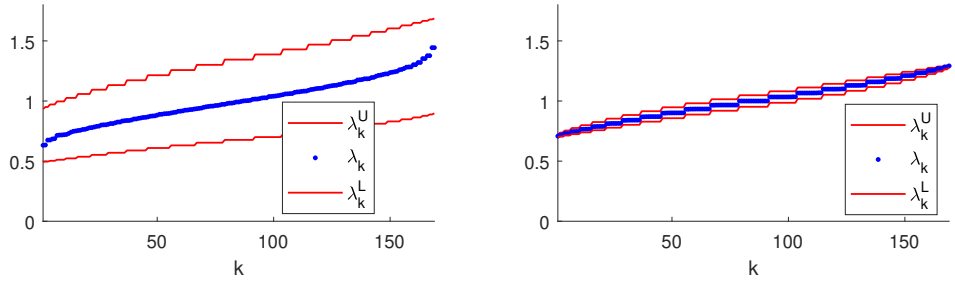


Figure 3.7: Eigenvalues of preconditioned stiffness matrices (Example 3.9) obtained by FDM (blue dots) and their lower and upper bounds (solid red lines) for preconditioners with data $\boldsymbol{a}^{\mathrm{p1}}$ (left) and $\boldsymbol{a}^{\mathrm{p2}}$ (right), respectively.

3.4 Conclusion

An efficient general algorithm providing two-sided guaranteed bounds to all individual eigenvalues of a preconditioned matrix of some discretized elliptic PDE is introduced in this paper. The assumption is that the matrices must be obtained as sums of certain locally built matrices such that the kernels of the corresponding pairs of local matrices are equal. Violating this assumption still allows using the introduced theory, however, the obtained bounds in practical examples could be trivial. The algorithm is based on comparing these pairs of local matrices, or even only on comparing local (material) properties of the operators and on a local connection among the DOFs defined by the discretization (see Example 3.3). We show that such local matrices can be naturally obtained in FEM, both for scalar and vector problems, in AML preconditioning, in SGFEM, and even in FDM. For these discretization methods, we present examples showing the construction of local matrices and we compute the resulting exact eigenvalues as well as the obtained bounds. To be able to compute all eigenvalues exactly, we introduce examples with relatively small N_{dof}, and we assume only two physical dimensions of the problems and one physical dimension in the SGFEM problem. However, the results can be naturally adapted to problems of any dimension.

In practical examples, some of the obtained bounds are tight and thus the eigenvalues are localized almost exactly. This is the case, for example, when the preconditioner was a multiple of the original operator in some parts of D; see Examples 3.4 and 3.9. In other cases, however, the bounds may not approximate the exact eigenvalues well. There is an open question whether the bounds can be improved under such conditions. Also, no a priori accuracy estimates for the bounds are available yet. In practice, we may sometimes need only bounds to condition numbers. Then only the first steps of Algorithms 1 and 2 can be performed to get a lower bound to the smallest eigenvalue and an upper bound to the largest eigenvalue, respectively, and their ratio as an upper bound to the condition number

$$\kappa((\mathsf{A}^{\mathrm{p}})^{-1}\mathsf{A}) \leq \lambda^{\mathrm{U}}_{N_{\mathrm{dof}}}/\lambda^{\mathrm{L}}_{1},$$

see Examples 3.6 and 3.7.

The method of characterizing the eigenvalues of a preconditioned matrix introduced in [45, 46] yields sets of intervals such that there exists a one-to-one mapping between the eigenvalues and the intervals such that each eigenvalue lies in an assigned interval. This method can yield different bounds (intervals) for the true eigenvalues than the method introduced in this paper. For instance, the method of [45, 46] can be able to identify a larger number of exact eigenvalues in some cases than our approach. In [99, Section 4.3] some further main differences are pointed out in the context of FEM. On the other hand, the approach introduced here can be applied to some more general problems and types of the discretization.

We recall that the main condition for the applicability of our algorithm is that the pairs of local matrices of the original operator and of the preconditioning operator must share their kernels. This can motivate readers to search for new choices of local matrices for the presented and also for further problems and discretization methods. We note that especially in general FDM, the obtained system matrix is usually not symmetric. Thus finding FD schemes leading to a symmetric matrix and splitting it into local matrices A_n is a challenging issue. There is also a space for new improvements in steps 2(b) and 2(c) in Algorithms 1 and 2, where more sophisticated choices of n and m instead of arbitrary could be suggested. If the original and preconditioning matrices arise from different operators, for example, as a stiffness matrix and a mass matrix, respectively, the guaranteed spectral bounds can still be obtained, but Friedrichs or Poincare constants and/or some sophisticated reconstruction techniques must be employed; see e.g. [39, 160, 145].

Chapter 4

Optimal FFT-accelerated finite element solver for homogenization

Abstract: *We provide a generalization and a linear algebra-based insight on an FFT-accelerated finite element (FE) homogenization scheme that was pioneered by Schneider at al. [131] and Leuschner and Fritzen [79]. The efficiency of the matrix-free scheme follows from a preconditioned well-scaled reformulation allowing for the use of the conjugate gradient or similar iterative solvers. The geometrically-optimal preconditioner — a discretized Green's function of a periodic homogeneous reference problem — has a block-diagonal structure in the Fourier space which permits its efficient inversion using the fast Fourier transform (FFT) techniques for generic regular meshes. This implies that the scheme scales as $\mathcal{O}(n \log(n))$ like FFT, rendering it equivalent to spectral solvers in terms of computational efficiency. However, in contrast to classical spectral solvers, the proposed scheme works with FE shape functions with local supports and is free of the Fourier ringing phenomenon. We showcase that the scheme achieves the number of iterations that are almost independent of spatial discretiszation and scales mildly with the phase contrast. Additionally, we discuss the equivalence between our displacement-based scheme and the recently proposed strain-based homogenization technique with finite-element projection.*

Reproduced from:

[101] **M. Ladecký**, J. R. Leute, A. Falsafi, I. Pultarová, L. Pastewka, T. Junge, and J. Zeman. Optimal FFT-accelerated finite element solver for homogenisation. 2022. DOI: 10.48550/arXiv.2203.02962

My contribution:

I was one of two main software developers of displacement-based finite elements solver in the open-source C++ library muSpectre [60]. I provided investigation of numerical behaviour, implementation of examples, creation of all resuts used in the publication, writing of the first draft and editing of the manuscript.

CRediT: Writing - Original Draft, Writing - Review & Editing, Conceptualization, Methodology, Software, Investigation, Visualization

4.1 Introduction

Complex macroscopic phenomena such as plastic yielding or damage in materials are governed by the nonlinear behavior of materials at meso-, micro-, or nanoscales. This intrinsic multiscale aspect of materials behavior creates the demand for the development of specialized scale-bridging techniques [83, 90, 38]. We focus here on an image-based homogenization technique [141] that combines the characterization of materials microstructures by high-resolution images (originating, e.g., from micro-computed tomography [87] or geometry-based models [138]) and a numerical solution of an underlying partial differential equation (PDE) with coefficient defined on a regular grid and typically involving periodic boundary conditions.

The solution of such PDEs discretized with the conventional finite element (FE) then becomes challenging even in the simplest scalar elliptic case, because it results in a system of equations with millions to billions of unknowns [59, Section 7.6]. In this regard, matrix-free iterative solvers are clearly preferential to direct solvers because of their lower memory footprint and speed, with the conjugate gradient (CG) method [56] being the most prominent candidate. However, the convergence behavior of the CG method depends on the spectral properties of the linear system matrix and deteriorates with decreasing FE mesh size [59, Section 7.7].

More than two decades ago, Moulinec and Suquet in their foundational works [102, 103] proposed a method that resolved these issues. According to its original interpretation, the method employed fixed-point iterations involving convolution with the Green's function of an auxiliary homogeneous problem with data and unknowns defined directly on the input grid. The method is suitable for high resolution homogenization problems thanks to the efficient implementation of the convolution step using the fast Fourier transform (FFT) algorithm [50] and mesh-size independent number of iterations.

These features attracted great interest in the community of computational mechanics of materials, as documented in two recent surveys by Schneider [129] and Lucarini et al. [84]. In what follows, we outline the developments most relevant to our work and refer an interested reader to [129, 84] for the full story of FFT-based methods.

Conjugate gradient solvers. As reported independently by Brisard and Dormieux [18] and Zeman et al. [158], the original spectral scheme [102, 103] can be further accelerated when replacing the fixed-point algorithm with the CG method. Later on, these computational observations were justified by Brisard and Dormieux [19], who showed that the computational scheme of Brisard and Dormieux [18] follows from the Ritz discretization of the Hashin-Shtrikman variational principles and by Vondřejc et al. [150], who showed that the computational scheme of Zeman et al. [158] follows from the Fourier-Galerkin discretization of the underlying PDE. These results directly extend to nonlinear problems linearized by the Newton's method, as first reported by Gélébart and Mondon-Cancel [43] and Kabel et al. [61] for the Green's function framework and by Zeman et al. [159] and de Geus et al. [25] for the Fourier-Galerkin framework.

Oscillations. Because the stress or strain fields may exhibit discontinuities at interphases between different material phases, discretizing the problem by Fourier trigonometric polynomials results in spurious numerical oscillations (also referred to as Fourier ringing artifacts in Section 2.5 of [129]) that pollute the approximate results. To reduce these oscillations, Kaßbohm et al. [64] smoothed the material data and Shanthraj et al. [135] filtered out high Fourier frequencies from the solution fields. A different approach was used by Willot et al. [155], who considered a modified Green's function obtained from a finite difference discretization. Schneider et al. [130] extended this approach by proposing a staggered grid finite difference approximation

to the underlying PDE, with a follow-up study [131] on FE discretization employing linear hexahedral elements. A related approach building on bi/trilinear FE basis functions instead of the Fourier basis was proposed by Leuschner and Fritzen [79]. Most recently, Leute et al. [80] developed a compatibility projection-based method in the spirit of Refs. [159, 25] while considering several finite difference- and finite element-based discretization stencils. Further discussion on mitigating the oscillation phenomena can be found in a dedicated comparative study of Ma et al. [86] or in Section 2.5 and 2.6 of Schneider [129].

Our work. We develop an alternative FFT-accelerated, oscillation-free computational homogenization scheme based purely on FE discretization that scales quasilinearly with the mesh size. We consider a nonlinear small-strain elasticity micromechanical problem discretized on a regular periodic grid with FE method in Sections 4.2 and linearize it with the Newton's method in Sections 4.3. Note that the localized support of the FE basis functions directly resolves the oscillation issue, see e.g. [80]. Thus no additional artificial adjustments of the data or the solution are needed.

In Section 4.4, we overcome the main drawback of the FE discretization — deteriorating conditioning of a linear system with the increasing size of the discretization grid — using a suitable preconditioner. Similarly to [131, 79], we construct the preconditioner from a stiffness matrix of a reference problem with generally anisotropic spatially uniform material data discretized on the same regular grid as the original problem. Using classical results, see e.g. [4, Section 5.1.2], we can guarantee that the condition number of the preconditioned linear system becomes almost independent on the mesh size. Moreover, employing local ratios of the problem material data and the reference problem material data, we can localize all individual eigenvalues [118, 45, 99]. This may help to better predict the convergence of the CG method, see e.g. [45, Section 2]. Therefore, the iterative CG solver is an optimal choice for the solution of problems with highly resolved microstructures. The application of the preconditioner is presented in detail in Section 4.5, with emphasis on reducing its computational complexity using the FFT algorithm [22].

We demonstrate the main features of the proposed algorithm by examples collected in Section 4.6 that covers 2-dimensional linear thermal conduction (with the necessary adjustments outlined in 4.A), 3–dimensional linear small-strain elasticity, and 2–dimensional nonlinear finite-strain elasto-plasticity. Section 4.7 is devoted to a comparison of our scheme with related developments by Schneider et al. [131] and Leuschner and Fritzen [79], and Section 4.8 concludes our work.

Notation. We denote d-dimensional vectors and matrices by boldface letters: $\boldsymbol{a} = (a_\alpha)_{\alpha=1}^d \in \mathbb{R}^d$ or $\boldsymbol{A} = (A_{\alpha\beta})_{\alpha,\beta=1}^d \in \mathbb{R}^{d\times d}$. Matrix-matrix and matrix-vector multiplications are denoted as $\boldsymbol{C} = \boldsymbol{B}\boldsymbol{A}$ and $\boldsymbol{c} = \boldsymbol{B}\boldsymbol{a}$. Vectors and matrices arising from the discretization will be denoted by $\boldsymbol{a}$ and $\boldsymbol{A}$, to highlight their special structure. The (I)-th component of $\boldsymbol{a}$ will be denoted as $\boldsymbol{a}[I]$ and (I,J)-th component of $\boldsymbol{A}$ will be denoted as $\boldsymbol{A}[I,J]$. We consider a general d-dimensional setting throughout the paper. However, for the sake of readability, we use $d = 2$ in the expanded form of matrices, such as in equation (4.2).

4.2 Nonlinear small-strain elasticity

We consider a d-dimensional rectangular periodic cell $\mathcal{Y} = \prod_{\alpha=1}^d \left[-\frac{l_\alpha}{2}, \frac{l_\alpha}{2}\right]$, of volume $|\mathcal{Y}| = \prod_{\alpha=1}^d l_\alpha$, to be a representative volume element, i.e., a typical material microstructure; see Fig. 4.1 for an illustration. The symmetries of small-strain elasticity allow us to employ the Mandel notation and reduce the dimension of the second-order strain tensor $\nabla_{\mathrm{s}}\boldsymbol{u} =$

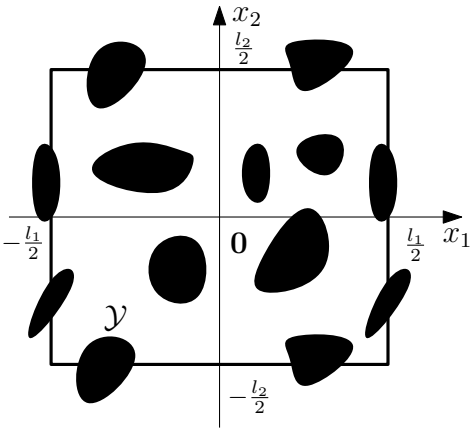


Figure 4.1: A rectangular two-dimensional cell $\mathcal{Y} = \left[-\frac{l_1}{2}, \frac{l_1}{2}\right] \times \left[-\frac{l_2}{2}, \frac{l_2}{2}\right]$ with outlined periodic microstructure.

$\frac{1}{2}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^\mathsf{T}) : \mathcal{Y} \to \mathbb{R}^{d\times d}_{\text{sym}}$ to a vector $\boldsymbol{\partial u} : \mathcal{Y} \to \mathbb{R}^{d_\text{m}}$, where $\boldsymbol{\partial}$ is the symmetrized gradient operator such that, for $d = 2$,

$$\boldsymbol{\partial u} = \begin{pmatrix} (\nabla_\text{s}\boldsymbol{u})_{11} \\ (\nabla_\text{s}\boldsymbol{u})_{22} \\ \sqrt{2}(\nabla_\text{s}\boldsymbol{u})_{12} \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial x_1} & 0 \\ 0 & \frac{\partial}{\partial x_2} \\ \frac{\sqrt{2}}{2}\frac{\partial}{\partial x_2} & \frac{\sqrt{2}}{2}\frac{\partial}{\partial x_1} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}.$$

Similarly, a fourth-order tensor $\mathbb{C} : \mathcal{Y} \to \mathbb{R}^{d\times d\times d\times d}_{\text{sym}}$ is represented by a matrix $\mathbf{C} : \mathcal{Y} \to \mathbb{R}^{d_\text{m}\times d_\text{m}}$,

$$\mathbf{C} = \begin{pmatrix} \mathbb{C}_{1111} & \mathbb{C}_{1122} & \sqrt{2}\mathbb{C}_{1112} \\ \mathbb{C}_{2211} & \mathbb{C}_{2222} & \sqrt{2}\mathbb{C}_{2212} \\ \sqrt{2}\mathbb{C}_{1211} & \sqrt{2}\mathbb{C}_{1222} & 2\mathbb{C}_{1212} \end{pmatrix},$$

where the number of components of the symmetrized gradient in the Mandel notation is $d_\text{m} = \dfrac{(d+1)d}{2}$, and indices $\alpha_\text{m}, \beta_\text{m}, \gamma_\text{m} \in \{1, \ldots, d_\text{m}\}$.

In the small-strain micromechanical problem, we split the overall strain $\boldsymbol{\varepsilon} : \mathcal{Y} \to \mathbb{R}^{d_\text{m}}$ into an average strain $\boldsymbol{e} = \frac{1}{|\mathcal{Y}|}\int_\mathcal{Y} \boldsymbol{\varepsilon}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \in \mathbb{R}^{d_\text{m}}$ and a periodically fluctuating field $\boldsymbol{\partial}\tilde{\boldsymbol{u}} : \mathcal{Y} \to \mathbb{R}^{d_\text{m}}$,

$$\boldsymbol{\varepsilon}(\boldsymbol{x}) = \boldsymbol{e} + \boldsymbol{\partial}\tilde{\boldsymbol{u}}(\boldsymbol{x}) \quad \text{for all } \boldsymbol{x} \in \mathcal{Y}.$$

Here, $\boldsymbol{\partial}\tilde{\boldsymbol{u}}$ denotes the symmetrized gradient in the Mandel notation, and the fluctuating displacement field $\tilde{\boldsymbol{u}}$ belongs to the space of admissible functions $\mathcal{V} = \left\{\tilde{\boldsymbol{v}} : \mathcal{Y} \to \mathbb{R}^d,\ \tilde{\boldsymbol{v}} \text{ is } \mathcal{Y}\text{-periodic}\right\}$. The governing equations for $\boldsymbol{\partial}\tilde{\boldsymbol{u}}$ are the mechanical equilibrium conditions

$$-\boldsymbol{\partial}^\mathsf{T}\boldsymbol{\sigma}(\boldsymbol{x}, \boldsymbol{e} + \boldsymbol{\partial}\tilde{\boldsymbol{u}}(\boldsymbol{x}), \boldsymbol{g}(\boldsymbol{x})) = \mathbf{0} \quad \text{for all } \boldsymbol{x} \in \mathcal{Y},$$

in which $\boldsymbol{\sigma} : \mathcal{Y} \times \mathbb{R}^{d_\text{m}} \times \mathbb{R}^g \to \mathbb{R}^{d_\text{m}}$ is the stress field and $\boldsymbol{g} : \mathcal{Y} \to \mathbb{R}^g$ designates the vector of internal parameters. The equilibrium equations are converted to the weak form

$$\int_\mathcal{Y} \boldsymbol{\partial}\tilde{\boldsymbol{v}}(\boldsymbol{x})^\mathsf{T}\boldsymbol{\sigma}(\boldsymbol{x}, \boldsymbol{e} + \boldsymbol{\partial}\tilde{\boldsymbol{u}}(\boldsymbol{x}), \boldsymbol{g}(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x} = 0 \qquad \text{for all } \tilde{\boldsymbol{v}} \in \mathcal{V},$$

where $\tilde{\boldsymbol{v}}$ is the test displacement field. The weak form (4.2) serves as a starting point for the FE method.

4.3 Finite element discretization

For the discretization of the weak form (4.2), we use a uniform mesh and conforming FE basis functions. In our setting, the discretization mesh does not necessarily follow the regular pixel/voxel structure, but can correspond to a space-filling pattern of finite elements; see the first row in Fig. 4.2. The discretization mesh is generated by a periodic repetition of a discretization stencil in the cell $\mathcal{Y}$; see the second row in Fig. 4.2. Such flexibility in discretization is useful, e.g., for damage or plasticity material models that exhibit sensitivity to mesh-grid anisotropy.

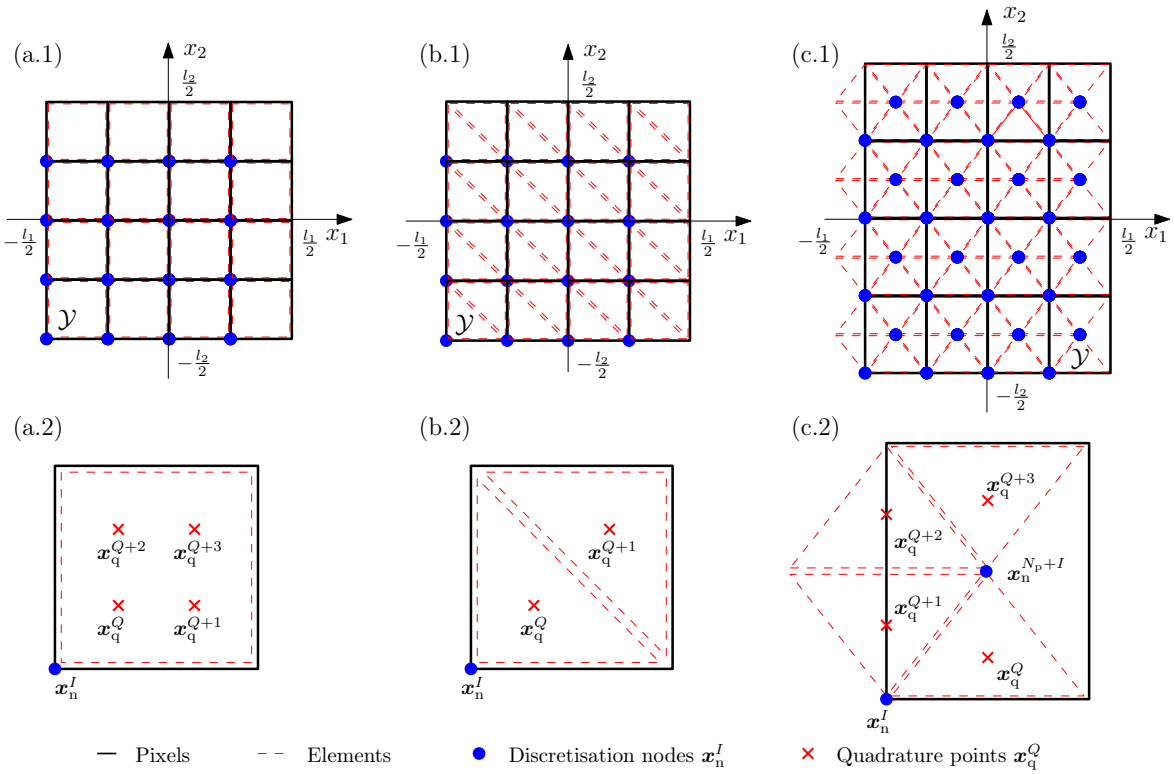


Figure 4.2: Example of regular periodic FE grids with associated discretization stencils for a two-dimensional cell $\mathcal{Y}$. All grids consists of 16 pixels ($N_{\mathrm{p}} = 16$). The row **(1)** shows: **(a.1)** grid with 16 discretization nodes ($N_{\mathrm{I}} = 16$) and quadrature points ($N_{\mathrm{Q}} = 64$), **(b.1)** grid with 16 discretization nodes ($N_{\mathrm{I}} = 16$) and 32 quadrature points ($N_{\mathrm{Q}} = 32$), **(c.1)** grid with 32 discretization nodes ($N_{\mathrm{I}} = 32$) and 64 quadrature points ($N_{\mathrm{Q}} = 64$). The row **(2)** shows: **(a.2)** one-node stencil ($N_{\mathrm{n}} = 1$) with one bilinear rectangular element and four quadrature points with the quadrature weights $w^{Q} = \frac{1}{4}V_{\mathrm{p}}$, **(b.2)** one-node stencil ($N_{\mathrm{n}} = 1$) with two linear triangular elements and two quadrature points with the quadrature weights $w^{Q} = \frac{1}{2}V_{\mathrm{p}}$, **(c.2)** two-node stencil ($N_{\mathrm{n}} = 2$) with four linear triangular elements and four quadrature points with the quadrature weights $w^{Q} = \frac{1}{4}V_{\mathrm{p}}$, Here, V_{p} denotes pixel volume, such that $V_{\mathrm{p}}N_{\mathrm{p}} = |\mathcal{Y}|$.

Strain and stress fields are evaluated at quadrature points $\boldsymbol{x}_{\mathrm{q}}^{Q}$, $Q \in \{1, 2, \ldots, N_{\mathrm{Q}}\}$, cf. Fig. 4.2, and the displacement fields are sampled at discretization nodes $\boldsymbol{x}_{\mathrm{n}}^{I}$, $I \in \{1, 2, \ldots, N_{\mathrm{I}}\}$. The number of discretization nodes $N_{\mathrm{I}} = N_{\mathrm{p}}N_{\mathrm{n}}$ is given by the number of pixel/voxel-associated discretization stencils N_{p} and the number of nodes per stencil N_{n}, as explained in Fig. 4.2. The number of degrees of freedom per stencil is thus dN_{n} and the total number of degrees of freedom per domain is dN_{I}.

Following the standard FE theory, $\tilde{\boldsymbol{v}}$ and $\tilde{\boldsymbol{u}}$ are approximated by continuous element-wise

polynomials $\mathcal{P}_k$ of the degree k; their symmetrized gradients $\boldsymbol{\partial}\tilde{\boldsymbol{v}}$ and $\boldsymbol{\partial}\tilde{\boldsymbol{u}}$ then become element-wise polynomials of the degree up to k. Furthermore, the integral (4.2) can be approximated with a suitable quadrature rule,

$$\begin{aligned}&\int_{\mathcal{Y}} \boldsymbol{\partial}\tilde{\boldsymbol{v}}(\boldsymbol{x})^{\mathsf{T}}\boldsymbol{\sigma}(\boldsymbol{x}, \boldsymbol{e}+\boldsymbol{\partial}\tilde{\boldsymbol{u}}(\boldsymbol{x}), \boldsymbol{g}(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x}\\ \approx &\sum_{Q=1}^{N_{\mathrm{Q}}} \boldsymbol{\partial}\tilde{\boldsymbol{v}}(\boldsymbol{x}_{\mathrm{q}}^{Q})^{\mathsf{T}}\boldsymbol{\sigma}(\boldsymbol{x}_{\mathrm{q}}^{Q}, \boldsymbol{e}+\boldsymbol{\partial}\tilde{\boldsymbol{u}}(\boldsymbol{x}_{\mathrm{q}}^{Q}), \boldsymbol{g}(\boldsymbol{x}_{\mathrm{q}}^{Q}))\,w^{Q},\end{aligned}$$

where the positions of the quadrature points $\boldsymbol{x}_{\mathrm{q}}^{Q}$ and the quadrature weights w^{Q} depend on the choice of the quadrature rule[1] ; recall Fig. 4.2.

Every component $\tilde{u}_\alpha$ of the unknown vector $\tilde{\boldsymbol{u}}$ is approximated by a linear combination

$$\tilde{u}_\alpha(\boldsymbol{x}) \approx \tilde{u}_\alpha^N(\boldsymbol{x}) = \sum_{I=1}^{N_{\mathrm{I}}} \tilde{u}_\alpha^N(\boldsymbol{x}_{\mathrm{n}}^I)\phi^I(\boldsymbol{x}) \quad \text{for all } \boldsymbol{x}\in\mathcal{Y},$$

where the coefficients $\tilde{u}_\alpha^N(\boldsymbol{x}_{\mathrm{n}}^I)$ are the nodal values of $\tilde{u}_\alpha^N$ at discretization nodes $\boldsymbol{x}_{\mathrm{n}}^I$ and ϕ^I are FE basis functions. A partial derivative of this approximation

$$\frac{\partial \tilde{u}_\alpha^N(\boldsymbol{x})}{\partial x_\beta} = \sum_{I=1}^{N_{\mathrm{I}}} \tilde{u}_\alpha^N(\boldsymbol{x}_{\mathrm{n}}^I)\frac{\partial \phi^I(\boldsymbol{x})}{\partial x_\beta} \quad \text{for all } \boldsymbol{x}\in\mathcal{Y},$$

evaluated in the quadrature points is given by

$$\frac{\partial \tilde{u}_\alpha^N(\boldsymbol{x}_{\mathrm{q}}^Q)}{\partial x_\beta} = \sum_{I=1}^{N_{\mathrm{I}}} \tilde{u}_\alpha^N(\boldsymbol{x}_{\mathrm{n}}^I)\frac{\partial \phi^I(\boldsymbol{x}_{\mathrm{q}}^Q)}{\partial x_\beta} \quad \text{for } Q=1,\ldots,N_{\mathrm{Q}}.$$

Therefore, if we store the nodal values of displacement $\tilde{\boldsymbol{u}}(\boldsymbol{x}_{\mathrm{n}}^I)$ into a vector $\tilde{\boldsymbol{u}}\in\mathbb{R}^{dN_{\mathrm{I}}}$, the gradient vector $\boldsymbol{\partial}\tilde{\boldsymbol{u}}\in\mathbb{R}^{d_{\mathrm{m}}N_{\mathrm{Q}}}$ at all quadrature points is given with

$$\boldsymbol{\partial}\tilde{\boldsymbol{u}} = \boldsymbol{D}\tilde{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{D}_1 & \mathsf{o} \\ \mathsf{o} & \boldsymbol{D}_2 \\ \frac{\sqrt{2}}{2}\boldsymbol{D}_2 & \frac{\sqrt{2}}{2}\boldsymbol{D}_1 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{u}}_1 \\ \tilde{\boldsymbol{u}}_2 \end{bmatrix},$$

where the matrix $\boldsymbol{D}\in\mathbb{R}^{d_{\mathrm{m}}N_{\mathrm{Q}}\times dN_{\mathrm{I}}}$ consists of sub-matrices of the partial derivatives

$$\boldsymbol{D}_\beta[Q,I] = \frac{\partial \phi^I(\boldsymbol{x}_{\mathrm{q}}^Q)}{\partial x_\beta} \quad \text{for } Q=1,\ldots,N_{\mathrm{Q}} \text{ and } I=1,\ldots,N_{\mathrm{I}},$$

and $\tilde{\boldsymbol{u}}_\alpha$ stores values of the displacement in the direction α. Due to the local supports of the basis functions ϕ^I, these sub-matrices exhibit significant sparsity, e.g., for the element-wise linear approximation, shown in the middle of Fig. 4.2, each row of $\boldsymbol{D}_\beta$ contains only two nonzero entries. Since both the interpolating and quadrature points are periodically distributed in $\mathcal{Y}$, the matrix $\boldsymbol{D}_\beta$ has a block circulant structure.

Now, the discretized weak form (4.2) using quadrature (4.3) can be rewritten in the matrix notation as

$$\tilde{\boldsymbol{v}}^{\mathsf{T}}\boldsymbol{D}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{\sigma}(\boldsymbol{e}+\boldsymbol{D}\tilde{\boldsymbol{u}}, \boldsymbol{g}) = 0 \quad \text{for all } \tilde{\boldsymbol{v}}\in\mathbb{R}^{dN_{\mathrm{I}}},$$

[1] Note, that under-integrated quadrature rule can be used to reduce memory footprint. However, the quality of the solution field can deteriorate, see Section 4.6.

where $\tilde{\boldsymbol{v}}$ stores the nodal values of test displacements, $\boldsymbol{e} \in \mathbb{R}^{d_\mathrm{m} N_\mathrm{Q}}$ stands for the discretized average strain, $\boldsymbol{\sigma} : \mathbb{R}^{d_\mathrm{m} N_\mathrm{Q}} \times \mathbb{R}^{g N_\mathrm{Q}} \to \mathbb{R}^{d_\mathrm{m} N_\mathrm{Q}}$ is a nonlinear map transforming, locally at quadrature points, a vector of discrete strains and internal parameters $\boldsymbol{g} \in \mathbb{R}^{g N_\mathrm{Q}}$ to discrete stresses, and the diagonal matrix $\boldsymbol{W} \in \mathbb{R}^{d_\mathrm{m} N_\mathrm{Q} \times d_\mathrm{m} N_\mathrm{Q}}$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_\mathrm{m} & \mathbf{o} & \mathbf{o} \\ \mathbf{o} & \boldsymbol{W}_\mathrm{m} & \mathbf{o} \\ \mathbf{o} & \mathbf{o} & \boldsymbol{W}_\mathrm{m} \end{bmatrix}$$

consists of d_m identical diagonal matrices $\boldsymbol{W}_\mathrm{m} \in \mathbb{R}^{N_\mathrm{Q} \times N_\mathrm{Q}}$ storing quadrature weights, $\boldsymbol{W}_\mathrm{m}[Q, Q] = w^Q$.

As the vector $\tilde{\boldsymbol{v}}$ is arbitrary, discretized weak form (4.3) is equivalent to the system of discrete nonlinear equilibrium conditions

$$\boldsymbol{D}^\mathsf{T} \boldsymbol{W} \boldsymbol{\sigma}(\boldsymbol{e} + \boldsymbol{D}\tilde{\boldsymbol{u}}, \boldsymbol{g}) = \mathbf{o}.$$

4.3.1 Linearisation

We employ the Newton's method to solve the nonlinear system (4.3) iteratively. For this purpose, the $(i+1)$-th approximation of the nodal displacement $\tilde{\boldsymbol{u}}_{(i+1)} \in \mathbb{R}^{N_\mathrm{I}}$ is given by the previous approximation $\tilde{\boldsymbol{u}}_{(i)} \in \mathbb{R}^{N_\mathrm{I}}$ adjusted by a finite displacement increment $\delta\tilde{\boldsymbol{u}}_{(i+1)} \in \mathbb{R}^{N_\mathrm{I}}$,

$$\tilde{\boldsymbol{u}}_{(i+1)} = \tilde{\boldsymbol{u}}_{(i)} + \delta\tilde{\boldsymbol{u}}_{(i+1)},$$

with an initial approximation $\tilde{\boldsymbol{u}}_{(0)} \in \mathbb{R}^{N_\mathrm{I}}$. The displacement increment $\delta\tilde{\boldsymbol{u}}_{(i+1)}$ follows from the solution of the linear system

$$\underbrace{\boldsymbol{D}^\mathsf{T} \boldsymbol{W} \boldsymbol{C}_{(i)} \boldsymbol{D}}_{\boldsymbol{K}_{(i)}} \delta\tilde{\boldsymbol{u}}_{(i+1)} = \underbrace{-\boldsymbol{D}^\mathsf{T} \boldsymbol{W} \boldsymbol{\sigma}(\boldsymbol{e} + \boldsymbol{D}\tilde{\boldsymbol{u}}_{(i)}, \boldsymbol{g}_{(i)})}_{\boldsymbol{b}_{(i)}},$$

where the algorithmic tangent matrix $\boldsymbol{C}_{(i)} = \dfrac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}(\boldsymbol{e} + \boldsymbol{D}\tilde{\boldsymbol{u}}_{(i)}, \boldsymbol{g}_{(i)}) \in \mathbb{R}^{d_\mathrm{m} N_\mathrm{Q} \times d_\mathrm{m} N_\mathrm{Q}}$,

$$\boldsymbol{C}_{(i)} = \begin{bmatrix} \boldsymbol{C}_{(i)11} & \boldsymbol{C}_{(i)12} & \boldsymbol{C}_{(i)13} \\ \boldsymbol{C}_{(i)21} & \boldsymbol{C}_{(i)22} & \boldsymbol{C}_{(i)23} \\ \boldsymbol{C}_{(i)31} & \boldsymbol{C}_{(i)32} & \boldsymbol{C}_{(i)33} \end{bmatrix},$$

is obtained from the constitutive tangent $\mathbf{C}_{(i)}(\boldsymbol{x}) = \dfrac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}(\boldsymbol{x}, \boldsymbol{e} + \boldsymbol{\partial}\tilde{\boldsymbol{u}}_{(i)}(\boldsymbol{x}), \boldsymbol{g}_{(i)}(\boldsymbol{x}))$, evaluated at quadrature points. Therefore, the sub-matrices $\boldsymbol{C}_{(i)\alpha_\mathrm{m}\beta_\mathrm{m}} \in \mathbb{R}^{N_\mathrm{Q} \times N_\mathrm{Q}}$ are diagonal with entries $\boldsymbol{C}_{(i)\alpha_\mathrm{m}\beta_\mathrm{m}}[Q, Q] = C_{(i)\alpha_\mathrm{m}\beta_\mathrm{m}}(\boldsymbol{x}_\mathrm{q}^Q)$. Traditionally, $\boldsymbol{K}_{(i)} \in \mathbb{R}^{dN_\mathrm{I} \times dN_\mathrm{I}}$ denotes the matrix of the linear system (4.3.1), and $\boldsymbol{b}_{(i)} \in \mathbb{R}^{dN_\mathrm{I}}$ stands for the right-hand side of (4.3.1).

4.4 Preconditioning

Recall that we focus on micromechanical problems with a finely described microstructure that involves a large number of degrees of freedom dN_I. We aim to use a memory-efficient matrix-free iterative method to find the solution of the linear system (4.3.1). The system matrix $\boldsymbol{K}_{(i)}$ is symmetric and positive definite for the symmetric algorithmic tangent $\boldsymbol{C}_{(i)}$, which renders the CG method as the method of choice, when combined with an appropriate preconditioner. This section discusses how to construct such a preconditioner in an optimal manner.

4.4.1 Reference material-based preconditioner

The idea of preconditioning, see, e.g., [50, Section 10.3] or [123, Chapters 9 and 10], is based on assumptions that the matrix of the preconditioned linear system

$$\boldsymbol{M}_{(i)}^{-1}\boldsymbol{K}_{(i)}\delta\tilde{\boldsymbol{u}}_{(i+1)} = \boldsymbol{M}_{(i)}^{-1}\boldsymbol{b}_{(i)},$$

has more favourable spectral properties than the original system $\boldsymbol{K}_{(i)}\delta\tilde{\boldsymbol{u}}_{(i+1)} = \boldsymbol{b}_{(i)}$. At the same time, the preconditioning matrix $\boldsymbol{M}_{(i)} \in \mathbb{R}^{dN_\mathrm{I}\times dN_\mathrm{I}}$ should be relatively easy to invert, such that the faster convergence of the iterative method compensates the computational overhead of the preconditioning. Please note that system matrix $\boldsymbol{M}_{(i)}^{-1}\boldsymbol{K}_{(i)}$ is no longer symmetric. However, for symmetric $\boldsymbol{M}_{(i)}$ and $\boldsymbol{K}_{(i)}$, system (4.4.1) is equivalent with the system preconditioned in the symmetric form $\boldsymbol{M}_{(i)}^{-1/2}\boldsymbol{K}_{(i)}\boldsymbol{M}_{(i)}^{-1/2}\delta\boldsymbol{z}_{(i+1)} = \boldsymbol{M}_{(i)}^{-1/2}\boldsymbol{b}_{(i)}$, where $\delta\boldsymbol{z}_{(i+1)} = \boldsymbol{M}_{(i)}^{1/2}\delta\tilde{\boldsymbol{u}}_{(i+1)}$. The latter form is in fact solved by the PCG method; see [123, Section 9.2.1] for more details. Nonetheless, we prefer the notation with the left preconditioning (4.4.1) for brevity.

Our approach is based on a preconditioner constructed in the same manner as the original matrix of the linear system (4.3.1),

$$\boldsymbol{M}_{(i)} = \boldsymbol{K}_{(i)}^\mathrm{ref} = \boldsymbol{D}^\mathsf{T}\boldsymbol{W}\boldsymbol{C}_{(i)}^\mathrm{ref}\boldsymbol{D} \in \mathbb{R}^{dN_\mathrm{I}\times dN_\mathrm{I}},$$

where the reference algorithmic tangent matrix $\boldsymbol{C}_{(i)}^\mathrm{ref} \in \mathbb{R}^{d_\mathrm{m}N_\mathrm{Q}\times d_\mathrm{m}N_\mathrm{Q}}$ corresponds to spatially uniform material data $\mathbf{C}_{(i)}^\mathrm{ref} \in \mathbb{R}^{d_\mathrm{m}\times d_\mathrm{m}}$. Finally, substituting (4.4.1) into (4.4.1) leads to the preconditioned linear system

$$\left(\boldsymbol{K}_{(i)}^\mathrm{ref}\right)^{-1}\boldsymbol{K}_{(i)}\delta\tilde{\boldsymbol{u}}_{(i+1)} = \left(\boldsymbol{K}_{(i)}^\mathrm{ref}\right)^{-1}\boldsymbol{b}_{(i)},$$

referred to as the reference material-based preconditioned problem in what follows. Notice that the spectrum of $\boldsymbol{K}_{(i)}^\mathrm{ref}$ contains null eigenvalue(s), associated with the infinitesimal rigid body modes, thus instead of the inverse of $\boldsymbol{K}_{(i)}^\mathrm{ref}$, we consider its (Moore-Penrose) pseudo-inverse[2] but still denote it by $\left(\boldsymbol{K}_{(i)}^\mathrm{ref}\right)^{-1}$ for notation simplicity.

In the following, we advocate this choice of the preconditioner. First, we derive a computationally efficient pseudo-inverse of $\boldsymbol{K}_{(i)}^\mathrm{ref}$ and second, we explain how the preconditioning impacts the spectral properties of the matrix of the system (4.4.1).

4.4.2 Fourier pseudo-inversion

Regular FE discretization of the problem with periodic boundary conditions leads to the same stencil for every pixel. Thus, for the uniform $\mathbf{C}_{(i)}^\mathrm{ref}$ in the whole $\mathcal{Y}$ (at every quadrature point $\boldsymbol{x}_\mathrm{q}^Q$), the resulting preconditioning matrix $\boldsymbol{K}_{(i)}^\mathrm{ref} \in \mathbb{R}^{dN_\mathrm{n}N_\mathrm{p}\times dN_\mathrm{n}N_\mathrm{p}}$,

$$\boldsymbol{K}_{(i)}^\mathrm{ref} = \begin{bmatrix} \boldsymbol{K}_{(i)11}^\mathrm{ref} & \boldsymbol{K}_{(i)12}^\mathrm{ref} \\ \boldsymbol{K}_{(i)21}^\mathrm{ref} & \boldsymbol{K}_{(i)22}^\mathrm{ref} \end{bmatrix} \in \mathbb{R}^{2N_\mathrm{p}\times 2N_\mathrm{p}}, \quad (\text{for } dN_\mathrm{n} = 2)$$

consists of $(dN_\mathrm{n})^2$ block-circulant blocks $\boldsymbol{K}_{(i)\bar{\alpha}\bar{\beta}}^\mathrm{ref} \in \mathbb{R}^{N_\mathrm{p}\times N_\mathrm{p}}$, where $\bar{\alpha}, \bar{\beta} \in \{1, \ldots, dN_\mathrm{n}\}$. All row vectors of a block-circulant block $\boldsymbol{K}_{(i)\bar{\alpha}\bar{\beta}}^\mathrm{ref}$ contain the same information and each row is block-periodically shifted with respect to the preceding one. This directly reflects the periodically repeated discretization pattern; recall Fig. 4.2, and that the action of $\boldsymbol{K}_{(i)\bar{\alpha}\bar{\beta}}^\mathrm{ref}$ is a discrete convolution of the displacement $\delta\tilde{\mathbf{u}}_{\bar{\beta}}$ with the discretization kernel, as schematically

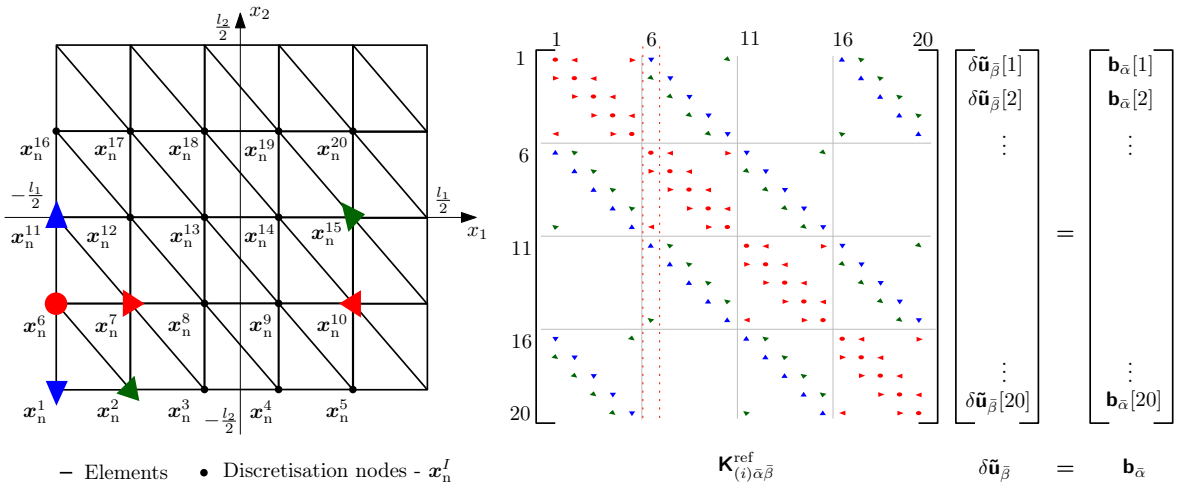


Figure 4.3: The block-circulant structure of block $\boldsymbol{K}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}}$ from the preconditioner $\boldsymbol{K}^{\mathrm{ref}}_{(i)}$ for spatially uniform material data $\mathbf{C}^{\mathrm{ref}}_{(i)}$ and periodic boundary condition. The two-dimensional ($d = 2$) discretization grid consisting of 20 pixels ($N_{\mathrm{p}} = 20$) with one-node stencil ($N_{\mathrm{n}} = 1$), and 20 discretization nodes ($N_{\mathrm{I}} = 20$) is shown left. Contributions of unit nodal displacement $\delta\tilde{\mathbf{u}}_{\bar{\beta}}[I] = 1$ to nodal components of right-hand side vector, graphically shown in the node $\boldsymbol{x}^{6}_{\mathrm{n}}$, are given as follows: (●) self contribution, contributions (▶) to the right node, (◀) to the left node, (◥) to the upper left node, (▲) to the upper node, (▼) to the bottom node, and (◢) to the bottom right node.

shown in Fig. 4.3. Note that in the one-dimensional ($d = 1$) case with one node per interval ($N_{\mathrm{n}} = 1$), $\boldsymbol{K}^{\mathrm{ref}}_{(i)}$ has only one circulant block, $\boldsymbol{K}^{\mathrm{ref}}_{(i)} = \boldsymbol{K}^{\mathrm{ref}}_{(i)11}$. The block structure of $\boldsymbol{K}^{\mathrm{ref}}_{(i)}$ appears whenever more than one type of degree of freedom is involved, i.e., $d > 1$, or $N_{\mathrm{n}} > 1$.

To make the inversion of $\boldsymbol{K}^{\mathrm{ref}}_{(i)}$ efficient, let us define the discrete d-dimensional Fourier transform matrix $\boldsymbol{F} \in \mathbb{R}^{N_{\mathrm{p}} \times N_{\mathrm{p}}}$ such that $\boldsymbol{F}^{\mathsf{H}} = \boldsymbol{F}^{-1}$, where $\boldsymbol{F}^{\mathsf{H}}$ is the conjugate transpose of $\boldsymbol{F}$. Then the Fourier counterpart

$$\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}} = \boldsymbol{F}\boldsymbol{K}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}}\boldsymbol{F}^{\mathsf{H}}$$

to any block-circulant $\boldsymbol{K}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}}$ is diagonal, and has the same spectrum (eigenvalues) as $\boldsymbol{K}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}}$. Therefore, $\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)}$ is block-diagonal and cheaply (pseudo) invertible

$$\left(\boldsymbol{K}^{\mathrm{ref}}_{(i)}\right)^{-1} = \boldsymbol{F}_d^{\mathsf{H}}\left(\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)}\right)^{-1}\boldsymbol{F}_d = \begin{bmatrix} \boldsymbol{F}^{\mathsf{H}} & \mathbf{o} \\ \mathbf{o} & \boldsymbol{F}^{\mathsf{H}} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)11} & \widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)12} \\ \widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)21} & \widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)22} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{F} & \mathbf{o} \\ \mathbf{o} & \boldsymbol{F} \end{bmatrix},$$

where $\boldsymbol{F}_d = I_{dN_{\mathrm{n}}} \otimes \boldsymbol{F}$ and $I_{dN_{\mathrm{n}}} \in \mathbb{R}^{dN_{\mathrm{n}} \times dN_{\mathrm{n}}}$ is the identity matrix. The expanded form in (4.4.2) apply for $dN_{\mathrm{n}} = 2$.

Finally, inserting (4.4.2) as the preconditioner in (4.4.1) leads to

$$\underbrace{\boldsymbol{F}_d^{\mathsf{H}}\left(\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)}\right)^{-1}\boldsymbol{F}_d}_{\left(\boldsymbol{K}^{\mathrm{ref}}_{(i)}\right)^{-1}}\boldsymbol{K}_{(i)}\delta\tilde{\boldsymbol{u}}_{(i+1)} = \underbrace{\boldsymbol{F}_d^{\mathsf{H}}\left(\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)}\right)^{-1}\boldsymbol{F}_d}_{\left(\boldsymbol{K}^{\mathrm{ref}}_{(i)}\right)^{-1}}\boldsymbol{b}_{(i)},$$

which reads in the expanded form as

[2] For details of Moore-Penrose pseudo-inverse refer to [50]

$$\underbrace{\boldsymbol{F}_d^{\mathsf{H}}(\boldsymbol{F}_d\boldsymbol{D}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{C}_{(i)}^{\mathrm{ref}}\boldsymbol{D}\boldsymbol{F}_d^{\mathsf{H}})^{-1}\boldsymbol{F}_d}_{(\boldsymbol{K}_{(i)}^{\mathrm{ref}})^{-1}}\underbrace{\boldsymbol{D}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{C}_{(i)}\boldsymbol{D}}_{\boldsymbol{K}_{(i)}}\delta\tilde{\boldsymbol{u}}_{(i+1)}$$
$$= -\underbrace{\boldsymbol{F}_d^{\mathsf{H}}(\boldsymbol{F}_d\boldsymbol{D}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{C}_{(i)}^{\mathrm{ref}}\boldsymbol{D}\boldsymbol{F}_d^{\mathsf{H}})^{-1}\boldsymbol{F}_d}_{(\boldsymbol{K}_{(i)}^{\mathrm{ref}})^{-1}}\underbrace{\boldsymbol{D}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{\sigma}(\boldsymbol{e}+\boldsymbol{D}\tilde{\boldsymbol{u}}_{(i)},\boldsymbol{g}_{(i)})}_{-\boldsymbol{b}_{(i)}}.$$

4.4.3 Spectrum of the preconditioned problem

To support the claim that the system matrix of the linear system (4.4.2) is well conditioned, we rely on the results published recently in [118, 45, 99] that provide simple algorithms for obtaining guaranteed two-sided bounds for all individual eigenvalues of the preconditioned operator by using element-by-element estimates. Note that extremal eigenvalue bounds obtained by such an element-by-element algorithm were introduced first in [4, 31] and found use, e.g., in algebraic multilevel methods [3]. Recently, motivated by Nielsen et al. [110], Gergelits et al. [45] published a new method yielding the bounds to all individual eigenvalues. This allows not only estimating the condition number of the preconditioned system but also to characterize its spectrum, which can provide more specific insights into the convergence of the CG method; see e.g. [45, Section 2] for more details. In [118, 99] an alternative algorithm is presented that can be applied to a variety of problems and discretization methods.

Let us recall the approach of **M. Ladecký** et al. [99]. Thanks to the local supports of FE basis functions ϕ^I it is possible to estimate all eigenvalues of the preconditioned linear system matrix (4.4.2). For each ϕ^I, we calculate

$$\lambda_I^{\mathrm{L}} = \min_{\boldsymbol{x}_{\mathrm{q}}^Q \in \operatorname{supp}\phi^I} \lambda_{\min}\left((\mathbf{C}_{(i)}^{\mathrm{ref}}(\boldsymbol{x}_{\mathrm{q}}^Q))^{-1}\mathbf{C}_{(i)}(\boldsymbol{x}_{\mathrm{q}}^Q)\right), \quad I = 1,\dots,N_{\mathrm{I}},$$
$$\lambda_I^{\mathrm{U}} = \max_{\boldsymbol{x}_{\mathrm{q}}^Q \in \operatorname{supp}\phi^I} \lambda_{\max}\left((\mathbf{C}_{(i)}^{\mathrm{ref}}(\boldsymbol{x}_{\mathrm{q}}^Q))^{-1}\mathbf{C}_{(i)}(\boldsymbol{x}_{\mathrm{q}}^Q)\right), \quad I = 1,\dots,N_{\mathrm{I}},$$

where $\operatorname{supp}\phi^I$ denotes the support of ϕ^I, and $\lambda_{\min}, \lambda_{\max}$ are the minimal and maximal generalized eigenvalues, respectively. For element-wise constant materials $\mathbf{C}_{(i)}$ and $\mathbf{C}_{(i)}^{\mathrm{ref}}$, any quadrature point $\boldsymbol{x}_{\mathrm{q}}^Q$ can be used to evaluate $\lambda_{\min}$ and $\lambda_{\max}$ on element. Therefore, only one pair $\lambda_{\min}, \lambda_{\max}$ has to be calculated for each element. Considering every λ_I^{L} and λ_I^{U} d-times and sorting these two sets into nondecreasing sequences gives the desired lower and upper eigenvalue bounds.

The resulting eigenvalue bounds are therefore independent of the characteristic element diameter h, which suggests that the condition number[3] $\kappa((\boldsymbol{K}_{(i)}^{\mathrm{ref}})^{-1}\boldsymbol{K}_{(i)})$ of the preconditioned linear system (4.4.2) will be independent of the problem size. In contrast, $\kappa(\boldsymbol{K}_{(i)}) = \mathcal{O}(h^{-2})$ for the unpreconditioned problem, e.g. [59, Section 7.7]. The ratio between the maximum and minimum eigenvalues of the preconditioned problem (4.4.2) will increase with an increasing ratio between extreme eigenvalues of $\mathbf{C}_{(i)}$ (so-called material contrast) and decrease as the reference material data $\mathbf{C}_{(i)}^{\mathrm{ref}}$ approach the material data $\mathbf{C}_{(i)}$ of the problem. Therefore, we can call our preconditioner as optimal, or more precisely, as *geometrically optimal*, which emphasizes that by keeping the discretization and changing only the data of the preconditioner

[3] Please note that by the condition number $\kappa((\boldsymbol{K}_{(i)}^{\mathrm{ref}})^{-1}\boldsymbol{K}_{(i)})$ we mean the ratio of the largest and the smallest eigenvalues of $(\boldsymbol{K}_{(i)}^{\mathrm{ref}})^{-1}\boldsymbol{K}_{(i)}$.

can lead to the matrix where all eigenvalues are the same, i.e., the condition number is 1. However, in such a case, the inversion of the preconditioner would become more expensive. The effects of phase contrast and the choice of $\mathbf{C}^{\text{ref}}_{(i)}$ on the CG performance are further illustrated with examples presented later in Section 4.6.2.

4.5 Implementation

The pseudo-algorithm of the incremental Newton-PCG solver for FE discretization on a regular grid is outlined in Algorithm 4.1. In the first part, we detail a matrix-free implementation. The second part deals with the assembly of the preconditioner via matrix-free operators and the third part focuses on the efficient pseudo-inversion of the preconditioner.

4.5.1 Matrix-free implementation

As mentioned in the previous sections, the explicit matrix structure is useful for explanation, but the computations can be performed more efficiently in a matrix-free manner.

The Gradient. Computational efficiency of our method relies on the fast evaluation of the gradient vector $\boldsymbol{\partial\tilde{u}} = \boldsymbol{D\tilde{u}}$. For regular periodic discretizations, the multiplication $\boldsymbol{D\tilde{u}}$ can be implemented as a convolution of $\boldsymbol{\tilde{u}}$ with a short kernel, namely the gradient stencil. To emphasize this, we replace matrix notation $\boldsymbol{D}$ and $\boldsymbol{D}^{\mathsf{T}}$ with the (matrix-free) operator notation $\mathcal{D} : \mathbb{R}^{dN_{\text{I}}} \to \mathbb{R}^{d_{\text{m}}N_{\text{Q}}}$ and $\mathcal{D}^{\mathsf{T}} : \mathbb{R}^{d_{\text{m}}N_{\text{Q}}} \to \mathbb{R}^{dN_{\text{I}}}$, such that

$$\mathcal{D}\delta\tilde{\boldsymbol{u}}_{(i+1)} = \boldsymbol{D}\delta\tilde{\boldsymbol{u}}_{(i+1)}, \quad \text{and} \quad \mathcal{D}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{C}_{(i)}\mathcal{D}\delta\tilde{\boldsymbol{u}}_{(i+1)} = \boldsymbol{D}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{C}_{(i)}\boldsymbol{D}\delta\tilde{\boldsymbol{u}}_{(i+1)}.$$

These operations are equal from the viewpoint of linear algebra, but algorithmically $\mathcal{D}$ is of linear $\mathcal{O}(N_{\text{I}})$ cost.

The fast Fourier transform. In the same manner, the multiplication with the discrete Fourier transform matrix can be replaced with the forward and the inverse fast Fourier transform algorithm

$$\mathcal{F}\delta\tilde{\boldsymbol{u}}_{(i+1)} = \boldsymbol{F}\delta\tilde{\boldsymbol{u}}_{(i+1)} \quad \text{and} \quad \mathcal{F}^{-1}\delta\tilde{\boldsymbol{u}}_{(i+1)} = \boldsymbol{F}^{H}\delta\tilde{\boldsymbol{u}}_{(i+1)},$$

of $\mathcal{O}(N_{\text{I}} \log N_{\text{I}})$ complexity.

Quadrature weights. Quadrature weights do not change through the process, so we fuse them with the transpose of the gradient operator

$$\mathcal{D}^{\mathsf{T}}_{\boldsymbol{W}} = \boldsymbol{D}^{\mathsf{T}}\boldsymbol{W},$$

where $\mathcal{D}^{\mathsf{T}}_{\boldsymbol{W}} : \mathbb{R}^{d_{\text{m}}N_{\text{Q}}} \to \mathbb{R}^{dN_{\text{I}}}$ can be interpreted as a weighted discrete divergence operator.

4.5.2 Assembly of the preconditioner

It may be useful to reassemble the preconditioner with updated $\mathbf{C}^{\text{ref}}_{(i)}$, whenever $\mathbf{C}_{(i)}$ significantly changes with respect to the previous Newton step, with $\mathbf{C}_{(i-1)}$. However, the use of matrix-free operators $\mathcal{D}, \mathcal{D}^{\mathsf{T}}_{\boldsymbol{W}}, \mathcal{F}$ and $\mathcal{F}^{-1}$ prohibits the direct assembly of $\widehat{\boldsymbol{K}}^{\text{ref}}_{(i)}$ through matrices, like in (4.4.2). Thus, we suggest an efficient algorithm for the assembly of $\widehat{\boldsymbol{K}}^{\text{ref}}_{(i)}$, that is outlined in Algorithm 4.2.

Algorithm 4.1 Pseudo-algorithm of the displacement-based Newton-PCG solver

1: **Initialize:**
2: $\tilde{\boldsymbol{u}}_{(0)}$, $\boldsymbol{e}$ ▷ initial displacement, macroscopic strain
3: η^{NW}, η^{CG} ▷ Newton- and CG-tolerance
4: $it^{\mathrm{NW}}_{\max}$, $it^{\mathrm{CG}}_{\max}$ ▷ max. iterations Newton and CG
5:
6: **for** $i = 0, 1, 2, \ldots, it^{\mathrm{NW}}_{\max}$ **do** ▷ Newton iteration
7: $\boldsymbol{g}_{(i)} = \ldots$ ▷ update internal parameters
8: $\boldsymbol{b}_{(i)} = -\mathcal{D}^{\mathsf{T}}_{\boldsymbol{W}}\boldsymbol{\sigma}(\boldsymbol{e} + \mathcal{D}\tilde{\boldsymbol{u}}_{(i)}, \boldsymbol{g}_{(i)})$ ▷ right-hand side
9: $\boldsymbol{C}_{(i)} = \dfrac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}}(\boldsymbol{e} + \mathcal{D}\tilde{\boldsymbol{u}}_{(i)})$ ▷ material tangent
10: Assembly $\left(\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)}\right)^{-1}$ ▷ Preconditioner assembly - Algorithm 4.2
11: Solve for $\delta\tilde{\boldsymbol{u}}_{(i+1)}$ with **PCG**:
12: $\boldsymbol{K}_{(i)}\delta\tilde{\boldsymbol{u}}_{(i+1)} = \boldsymbol{b}_{(i)}$ with preconditioner $\left(\boldsymbol{K}^{\mathrm{ref}}_{(i)}\right)^{-1}$ in $it^{\mathrm{CG}}_{\max}$ steps or until the termination criteria (4.6) is reached.
13: $\tilde{\boldsymbol{u}}_{(i+1)} = \tilde{\boldsymbol{u}}_{(i)} + \delta\tilde{\boldsymbol{u}}_{(i+1)}$ ▷ iterative update
14: **if** $\left\|\delta\tilde{\boldsymbol{u}}_{(i+1)}\right\| \leq \eta^{\mathrm{NW}}\left\|\tilde{\boldsymbol{u}}_{(i+1)}\right\|$ **then**
15: Proceed to line 18 ▷ Newton's method converged
16: **end if**
17: **end for**
18: **return** $\tilde{\boldsymbol{u}}_{(i+1)}$

First, take a look at (block-periodic) $\bar{\alpha}\bar{\beta}$-block $\boldsymbol{K}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}} \in \mathbb{R}^{N_{\mathrm{p}}\times N_{\mathrm{p}}}$ of $\boldsymbol{K}^{\mathrm{ref}}_{(i)} \in \mathbb{R}^{dN_{\mathrm{n}}N_{\mathrm{p}}\times dN_{\mathrm{n}}N_{\mathrm{p}}}$. Thanks to the convolution theorem, the whole diagonal $\mathrm{diag}(\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}}) \in \mathbb{R}^{N_{\mathrm{p}}}$ can be obtained by the FFT of any, say the first, row or, because of the symmetry, column of $\boldsymbol{K}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}}$,

$$\mathrm{diag}(\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}}) = \mathcal{F}(\boldsymbol{K}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}}[1,:])^{\mathsf{T}} = \mathcal{F}(\boldsymbol{K}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}}[:,1])$$

where a colon indicates a complete column or row. Before the FFTs, we have to compute one column $\boldsymbol{K}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}}[1,:]$ for each of $(dN_{\mathrm{n}})^2$ blocks $\boldsymbol{K}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}}$ of $\boldsymbol{K}^{\mathrm{ref}}_{(i)}$. Consider a unit impulse vector $\boldsymbol{i}^p \in \mathbb{R}^{dN_{\mathrm{n}}N_{\mathrm{p}}}$ that has only one non-zero element equal to 1 on the p-th position. When we apply $\boldsymbol{K}^{\mathrm{ref}}_{(i)}$ to vector $\boldsymbol{i}^1$, we obtain the first columns of dN_{n} blocks $\boldsymbol{K}^{\mathrm{ref}}_{(i)\bar{\alpha}1}$. From the structure of $\boldsymbol{K}^{\mathrm{ref}}_{(i)}$ visible in (4.4.2) it is obvious that we need dN_{n} vectors $\boldsymbol{i}^p$ to obtain all $(dN_{\mathrm{n}})^2$ columns $\boldsymbol{K}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}}[1,:]$, where $p = (\bar{\beta}-1)N_{\mathrm{p}} + 1$ and $\bar{\beta} \in \{1, \ldots, dN_{\mathrm{n}}\}$. The whole procedure is schematically shown in Fig. 4.4.

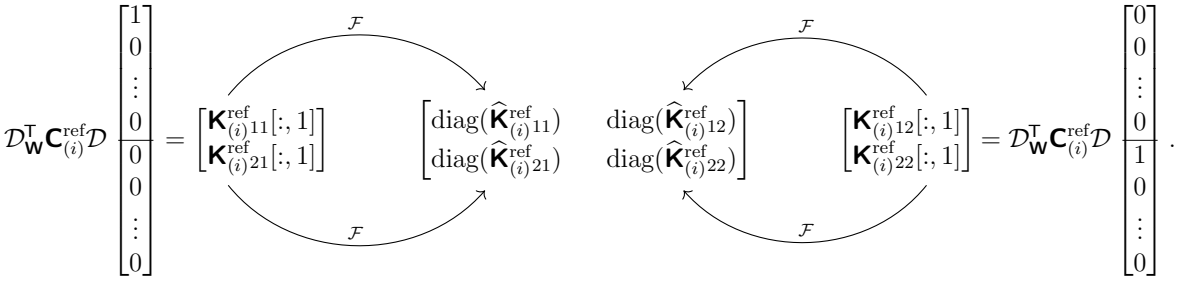


Figure 4.4: The schematic procedure of matrix-free assembly of $\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)}$ for $dN_{\mathrm{n}} = 2$. First columns of blocks $\boldsymbol{K}^{\mathrm{ref}}_{(i)11}$ and $\boldsymbol{K}^{\mathrm{ref}}_{(i)21}$ are obtained as a result of the matrix-free action of $\boldsymbol{K}^{\mathrm{ref}}_{(i)}$ on the unit impulse vector $\boldsymbol{i}^1$. Diagonals $\mathrm{diag}(\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)11})$ and $\mathrm{diag}(\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)21})$ are then computed through d-dimensional FFT of $\boldsymbol{K}^{\mathrm{ref}}_{(i)11}[:,1]$ and $\boldsymbol{K}^{\mathrm{ref}}_{(i)21}[:,1]$, respectively. By analogy, columns of blocks $\boldsymbol{K}^{\mathrm{ref}}_{(i)12}$ and $\boldsymbol{K}^{\mathrm{ref}}_{(i)22}$ are obtained by the matrix-free action of $\boldsymbol{K}^{\mathrm{ref}}_{(i)}$ on the unit impulse vector $\boldsymbol{i}^p$ where $p = (2-1)N_{\mathrm{p}} + 1$.

4.5.3 Pseudo-inverse of the preconditioner

Once we have all diagonal blocks $\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}}$, we may proceed to the computation of the pseudo-inverse of $\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)}$. By a proper row and column reordering, it can be seen that the pseudo-inverse of the block diagonal matrix $\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)}$ is equivalent to the pseudo-inverse of N_{p} (number of pixels/stencils) submatrices

$$\begin{bmatrix} \widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)11}[J,J] & \dots & \widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)1\bar{\beta}}[J,J] \\ \vdots & \ddots & \vdots \\ \widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)\bar{\alpha}1}[J,J] & \dots & \widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}}[J,J] \end{bmatrix}^{-1} \in \mathbb{R}^{dN_{\mathrm{n}} \times dN_{\mathrm{n}}}, \quad \text{where} \quad J \in \{1,\dots,N_{\mathrm{p}}\}.$$

The $(N_{\mathrm{p}} - 1)$ submatrices are of full rank and thus directly invertible. Only one submatrix, corresponding to the zero frequency Fourier mode, is singular and has to be treated separately. This block has exactly d null eigenvalues corresponding to d rigid-body modes. We compute the (Moore-Penrose) pseudo-inverse of this block instead of its inversion[4]. The pseudo-inverse can be computed exactly by restriction onto the space orthogonal to the kernel of the singular block. For any specific type of FE and the corresponding discretization stencil, the kernel can be exactly identified.

[4]Please note that the Moore-Penrose pseudo-inverse is depicted by † in Algorithm 4.2.

Algorithm 4.2 Pseudo-algorithm of reference material based preconditioner assembly

Initialize:

$\boldsymbol{C}^{\mathrm{ref}}_{(i)}$ ▷ spatially uniform reference material

for $\bar{\beta} = 1, \ldots, dN_{\mathrm{n}}$ **do** ▷ loop over d vectors

  $p = (\bar{\beta} - 1)N_{\mathrm{p}} + 1$ ▷ column index

  $\boldsymbol{c}_{\bar{\beta}} = \mathcal{D}_{\boldsymbol{W}}^{\mathsf{T}} \boldsymbol{C}^{\mathrm{ref}}_{(i)} \mathcal{D} \boldsymbol{i}^{p}$ ▷ p-th column of $\boldsymbol{K}^{\mathrm{ref}}_{(i)}$

  for $\bar{\alpha} = 1, \ldots, dN_{\mathrm{n}}$ **do**

    $\mathrm{diag}(\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}}) = \mathcal{F}(\boldsymbol{c}_{\bar{\beta}}[(\bar{\alpha} - 1)N_{\mathrm{p}} + 1 : \bar{\alpha}N_{\mathrm{p}}])$ ▷ assign to $\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}}$ diagonals

  end for

end for

▷ pseudo-inverse of singular submatrix of $\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)}$

$$\left(\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}}\right)^{-1}[1,1] = \begin{bmatrix} \widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)11}[1,1] & \ldots & \widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)1\bar{\delta}}[1,1] \\ \vdots & \ddots & \vdots \\ \widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)\bar{\gamma}1}[1,1] & \ldots & \widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)\bar{\gamma}\bar{\delta}}[1,1] \end{bmatrix}^{\dagger}_{\bar{\alpha}\bar{\beta}}$$ ▷ 1-th block

for $J = 2, \ldots, N_{\mathrm{p}}$ **do** ▷ inverse of remaining submatrices of $\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)}$

$$\left(\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)\bar{\alpha}\bar{\beta}}\right)^{-1}[J,J] = \begin{bmatrix} \widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)11}[J,J] & \ldots & \widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)1\bar{\delta}}[J,J] \\ \vdots & \ddots & \vdots \\ \widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)\bar{\gamma}1}[J,J] & \ldots & \widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)\bar{\gamma}\bar{\delta}}[J,J] \end{bmatrix}^{-1}_{\bar{\alpha}\bar{\beta}}$$ ▷ J-th block

end for

4.6 Numerical experiments

We demonstrate the numerical behavior of the proposed approach on several examples. In general, we compare our displacement-based (DB) FE scheme, described in the previous sections, with the (P)CG accelerated strain-based (SB) Fourier-Galerkin method with numerical integration taken from [150, 159, 25]. All results were obtained with the μSpectre software, an open-source platform for efficient FFT-based continuum mesoscale modelling, which is freely available at `https://gitlab.com/muspectre/muspectre`. The software package includes the examples, which are described in the following sections.

Termination criteria. To obtain comparable results, we have to choose the corresponding termination criteria for both SB and DB schemes. The Newton's method stops when the relative norm of strain increment drops below the tolerance η^{NW}, $\left\|\delta\boldsymbol{\partial}\tilde{\boldsymbol{u}}_{(i+1)}\right\| \leq \eta^{\mathrm{NW}}\left\|\boldsymbol{\partial}\tilde{\boldsymbol{u}}_{(i+1)}\right\|$. The (P)CG solver is stopped when the relative $\left(\boldsymbol{K}^{\mathrm{ref}}_{(i)}\right)^{-1}$-norm of the residual drops below the

tolerance η^{CG},

$$\left\|\boldsymbol{r}^k_{(i+1)}\right\|_{(\boldsymbol{K}^{\mathrm{ref}}_{(i)})^{-1}} \leq \eta^{\mathrm{CG}} \left\|\boldsymbol{r}^0_{(i+1)}\right\|_{(\boldsymbol{K}^{\mathrm{ref}}_{(i)})^{-1}}.$$

This choice is motivated by the optimal property of PCG to minimize the error energy norm

$$\left\|\boldsymbol{e}^k_{(i+1)}\right\|_{\boldsymbol{K}_{(i)}} = \left\|\delta\tilde{\boldsymbol{u}}_{(i+1)} - \delta\tilde{\boldsymbol{u}}^k_{(i+1)}\right\|_{\boldsymbol{K}_{(i)}},$$

where $\delta\tilde{\boldsymbol{u}}^k_{(i+1)}$ is the approximation of the solution $\delta\tilde{\boldsymbol{u}}_{(i+1)}$ in k-th PCG step. If $(\boldsymbol{K}^{\mathrm{ref}}_{(i)})^{-1}\boldsymbol{K}_{(i)} \approx \boldsymbol{I}$, then $(\boldsymbol{K}^{\mathrm{ref}}_{(i)})^{-1}$-norm of the residual $\boldsymbol{r}^k_{(i+1)} = \boldsymbol{b}_{(i)} - \boldsymbol{K}_{(i)}\delta\tilde{\boldsymbol{u}}^k_{(i+1)}$ approximate the error energy norm,

$$\left\|\boldsymbol{r}^k_{(i+1)}\right\|_{(\boldsymbol{K}^{\mathrm{ref}}_{(i)})^{-1}} = \boldsymbol{e}^k_{(i+1)}{}^{\mathsf{T}}\boldsymbol{K}^{\mathsf{T}}_{(i)}(\boldsymbol{K}^{\mathrm{ref}}_{(i)})^{-1}\boldsymbol{K}_{(i)}\boldsymbol{e}^k_{(i+1)}{}^{\mathsf{T}} = \left\|\boldsymbol{e}^k_{(i+1)}\right\|_{\boldsymbol{K}^{\mathsf{T}}_{(i)}(\boldsymbol{K}^{\mathrm{ref}}_{(i)})^{-1}\boldsymbol{K}_{(i)}}.$$

Additionally, the $(\boldsymbol{K}^{\mathrm{ref}}_{(i)})^{-1}$-norm of residual naturally appears in the PCG algorithm, therefore is free to obtain.

4.6.1 Linear steady-state thermal conduction problem

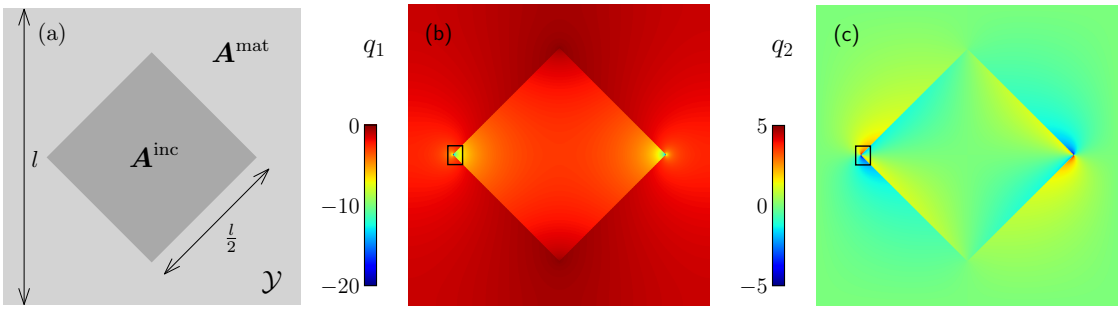


Figure 4.5: A linear heat transfer problem from Section 4.6.1. The square periodic unit cell $\mathcal{Y}$ with a square inclusion **(a)**. The flux field component q_1 **(b)** and q_2 **(c)** arising from average temperature gradient $\boldsymbol{e} = [0.01, 0.0]^{\mathsf{T}}$. Results are obtained with one-node FE stencil ($N_{\mathrm{n}} = 1$) with two linear triangular elements discretization and 815 nodes in both directions ($N_{\mathrm{I}} = 815^2$).

In the first example, we demonstrate the oscillation-free character of gradient fields arising from the FE discretization. For this purpose, we reconstruct the benchmark problem from [79, Section 3.7.1] or [18, Section 3.2], where the Fourier-Galerkin methods exhibit significant discretization artifacts.

We consider a scalar problem of linear heat transfer, where we look for the flux field $\boldsymbol{q}$ satisfying the weak balance condition (4.A); see 4.A for more details. The microstructure is defined by the square periodic unit cell $\mathcal{Y}$, as sketched on the left-hand side of Fig. 4.5. The composite microstructure consists of an insulating matrix with the conductivity $\mathbf{A}^{\mathrm{mat}} = 100\,\mathbf{I}$, and a conducting inclusion with the conductivity $\mathbf{A}^{\mathrm{inc}} = 100\,\mathbf{A}^{\mathrm{mat}}$. An average temperature gradient $\boldsymbol{e} = [0.01, 0.0]^{\mathsf{T}}$ is applied. The number of pixels is 815^2, and the material coefficients are constant per pixel. The choice of reference material $\mathbf{A}^{\mathrm{ref}}$ has no effect on discretization artifacts, thus we set $\mathbf{A}^{\mathrm{ref}} = \mathbf{I}$ for simplicity. Components of the global flux field $\boldsymbol{q}$ are shown in Fig. 4.5; q_1 in the middle and q_2 on the right-hand side. The regions of details depicted in Fig. 4.6 and Fig. 4.7 are highlighted by the black rectangles in Fig. 4.5.

In Fig. 4.6, we show the details of heat fluxes for various discretizations: the Fourier-Galerkin method in the column (a), the one-node FE stencil ($N_\mathrm{n} = 1$) with two linear triangular elements and two quadrature points (Fig. 4.2 (b)) in the column (b), the one-node FE stencil ($N_\mathrm{n} = 1$) with one bilinear rectangular element and four quadrature points (Fig. 4.2 (a)) in the column (c) and the one-node FE stencil ($N_\mathrm{n} = 1$) with one bilinear rectangular element and one quadrature point in the column (d).

The Fourier-Galerkin method exhibits strong oscillations through the region. The under-integrated FE scheme shows checkerboard patterns, while FE solutions of fully-integrated schemes are devoid of oscillations in the interior of the domains occupied by a single phase as discretization discrepancies remain confined to the vicinity of the phase boundaries. For instance, triangular discretizations reduce the phase boundary discretization artifacts to the two pixel-wide layer around the phase boundary.

The zigzag patterns on the phase boundary arise from the pixel-based geometry. If the elements can capture the interface of the two phases exactly, we do not get any discretization artifacts, as can be seen in Fig. 4.6 and Fig. 4.7 in the column (b). This speaks in favour of using FE over the Fourier-Galerkin discretization.

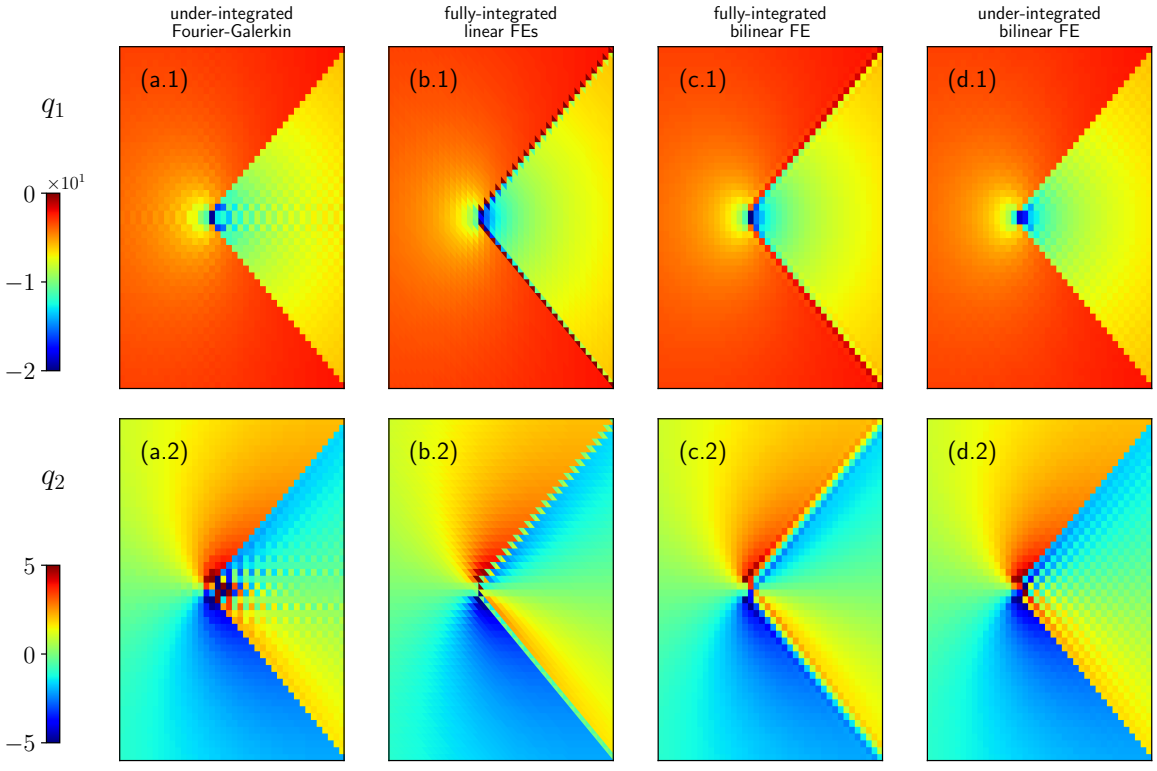


Figure 4.6: Local heat flux field components q_1 **(1)** and q_2 **(2)** from experiment in Section 4.6.1, obtained with the under-integrated Fourier-Galerkin method is shown in the column **(a)**, one-node FE stencil ($N_\mathrm{n} = 1$) with two linear triangular elements and two quadrature points is shown in the column **(b)**, , one-node FE stencil ($N_\mathrm{n} = 1$) with one bilinear rectangular element and four quadrature points is shown in the column **(c)** and one-node FE stencil ($N_\mathrm{n} = 1$) with one bilinear rectangular element and one quadrature points is shown in the column **(d)**.

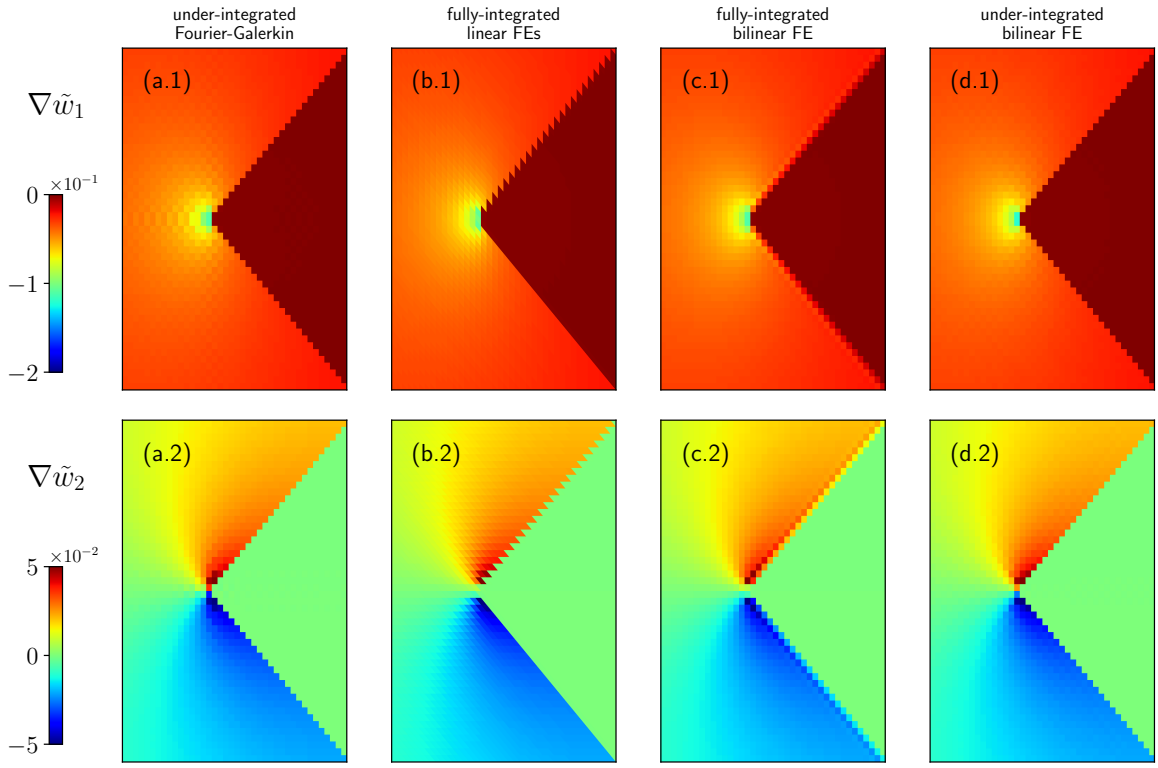


Figure 4.7: Local temperature gradient field components $\nabla\tilde{w}_1$ **(1)** and $\nabla\tilde{w}_2$ **(2)** from experiment 4.6.1, obtained with the under-integrated Fourier-Galerkin method is shown in the column **(a)**„ one-node FE stencil ($N_\mathrm{n} = 1$) with two linear triangular elements and two quadrature points is shown in the column **(b)**, one-node FE stencil ($N_\mathrm{n} = 1$) with one bilinear rectangular element and four quadrature points is shown in the column **(c)** and one-node FE stencil ($N_\mathrm{n} = 1$) with one bilinear rectangular element and one quadrature points is shown in the column **(d)**.

4.6.2 Small-strain elasticity problem

The second example focuses on the effect of the preconditioner on the number of PCG iterations with respect to the number of discretization nodes N_I and phase contrast ρ. For this purpose, we use Hashin's coated sphere construction adapted from [130, Section 4.1] and the references therein.

We choose a linear small-strain elastic problem described in Section 4.2. The three-phase microstructure representing a coated sphere in the matrix with effective material properties is depicted in Fig. 4.8, with the core radius $r_1 = 0.2$, annulus-shaped coating outer radius $r_2 = 0.4$ and the cubic domain edge length $l = 1$. An average macroscopic strain $\boldsymbol{e} = [1, 0, 0, 0, 0, 0]^\mathsf{T}$ is applied. We assume isotropic phases with bulk and shear moduli $\mathrm{K}_1, \mathrm{G}_1$ in the core, $\mathrm{K}_2, \mathrm{G}_2$ in the coating and $\mathrm{K}_\mathrm{eff}, \mathrm{G}_\mathrm{eff}$ in the surrounding matrix. The bulk moduli $\mathrm{K}_1, \mathrm{K}_2$ are chosen in a way that the resulting response of the unit cell is equivalent to the response of a homogeneous material with K_eff. As a consequence, the bulk moduli $\mathrm{K}_1, \mathrm{K}_2$ have to be balanced for particular phase contrast $\rho = \mathrm{K}_2/\mathrm{K}_1$.

First, in accordance with Schneider et al. [130, Section 4.1.3], we set $\rho = 10^3$ and the remaining parameters to

$$\begin{aligned} \mathrm{K}_1 &\doteq 0.00132060, \quad & \mathrm{K}_2 &\doteq 1.3206033, \quad & \mathrm{K}_\mathrm{eff} &\doteq 1.0, \\ \mathrm{G}_1 &\doteq 0.00079236, \quad & \mathrm{G}_2 &\doteq 0.7923620, \quad & \mathrm{G}_\mathrm{eff} &\doteq 0.6. \end{aligned}$$

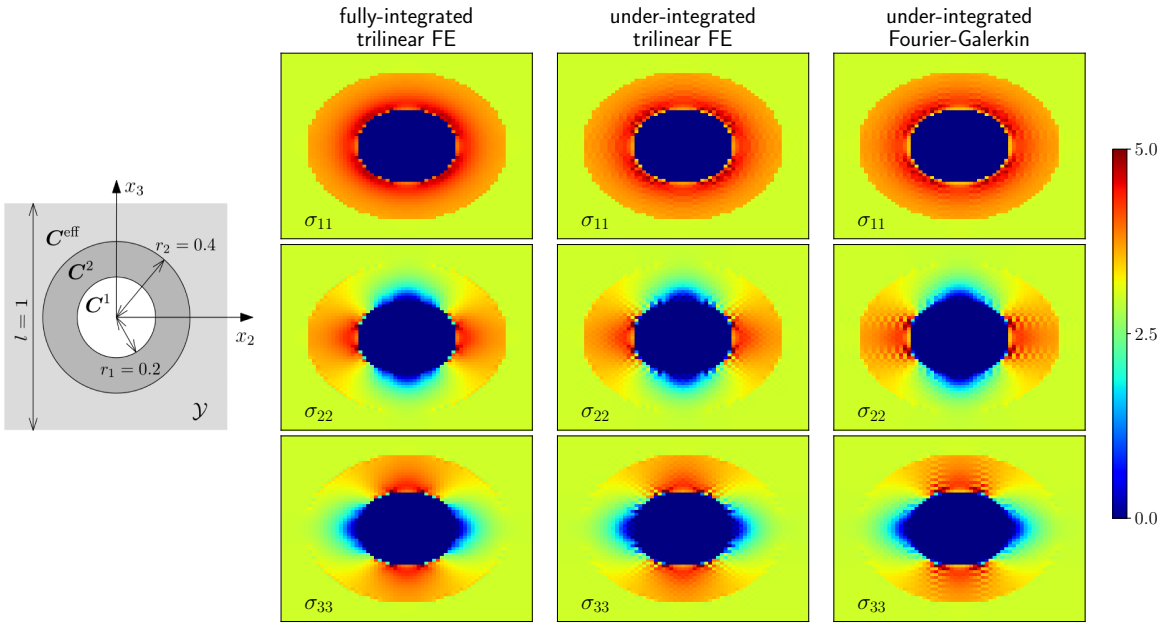


Figure 4.8: Two-dimensional sections at $x_1 = 0.5$ of the 3-dimensional cubic periodic unit cell $\mathcal{Y}$ with a coated sphere inclusion. Radii $r_1 = 0.2$, $r_2 = 0.4$ and the domain size $l = 1$. Components of the local stress fields $\sigma_{\alpha\alpha}$ for trilinear FE discretization with eight quadrature points (left column), trilinear FE discretization with one quadrature point (middle column) and under-integrated Fourier-Galerkin discretization (right column) with the number discretization nodes $N_{\text{I}} = 65^3$.

Two-dimensional sections at $x_1 = 0.5$ of global stress field components are shown in Fig. 4.8 right. Fully-integrated trilinear FE discretization (the left column in Fig. 4.8) generates oscillation free results compared to under-integrated trilinear FE discretization (the middle column in Fig. 4.8) the under-integrated Fourier-Galerkin discretization (the left column in Fig. 4.8).

Second, we are interested in how our preconditioned scheme behaves with respect to the number of discretization nodes N_{I} and varying phase contrast ρ. The convergence of PCG depends on the choice of reference material $\mathbf{C}^{\text{ref}}$. We compare two cases: the first $\mathbf{C}^{\text{ref}}_{\mathbf{I}_s} = \mathbf{I}_s$, with $\mathbf{I}_s \in \mathbb{R}^{d_{\text{m}} \times d_{\text{m}}}$ being the symmetrized identity tensor $(I_{\text{s}})_{\alpha\beta\iota\kappa} = \frac{1}{2}(\delta_{\alpha\iota}\delta_{\beta\kappa} + \delta_{\alpha\kappa}\delta_{\beta\iota})$ in the Mandel notation, and secondly $\mathbf{C}^{\text{ref}}_{\text{mean}} = \frac{1}{|\mathcal{Y}|}\sum_{Q=1}^{N_{\text{Q}}} \mathbf{C}(\boldsymbol{x}^Q_{\text{q}})w^Q$, where $\mathbf{C}^{\text{ref}}_{\text{mean}}$ is the mean stiffness matrices over $\mathcal{Y}$.

The preconditioner with mean reference material $\mathbf{C}^{\text{ref}}_{\text{mean}}$ exhibits better performance in all studied cases, see Fig. 4.9. The numbers of iterations slowly increases with the growing N_{I} until it stabilizes for sufficiently fine discretizations. In addition, $\mathbf{C}^{\text{ref}}_{\text{mean}}$ significantly reduced the phase contrast sensibility, especially for $\rho > 1$ (softer sphere core).

4.6.3 Finite-strain elasto-plastic problem

The purpose of the last example is twofold. First, we demonstrate the applicability of the approach to real-world problems in the finite-strain setting, and the effect of nonphysical oscillations on the results. Second, we point out the equivalence of our DB FE scheme with SB scheme with FE projection operator recently proposed by Leute et al. [80]. The equivalence of these two approaches is briefly explained later in Section 4.7.1.

For this purpose, we adapt the example from Section 5.5 of de Geus et al. [25]. The example

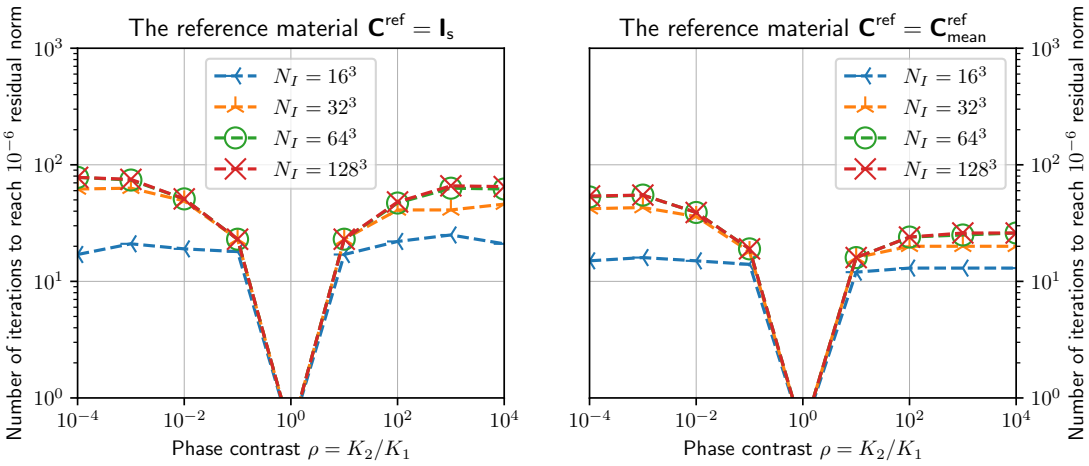


Figure 4.9: The number of PCG iterations for trilinear FE discretization for different phase contrasts ρ and number of discretization nodes N_I. Termination parameter for linear solver $\eta^{\text{CG}} = 10^{-6}$.

studies a sample of a dual-phase steel obtained by a scanning electron microscope. Responses of the material phases are elastic and homogeneous in the elastic part of deformation with Young's moduli $E = 200\,\text{GPa}$ and Poisson's ratios $\nu = 0.3$, and differ in the parameters of linear hardening in the plastic regions, see [25, Section 5] for more details on the material model.

The yield stress τ_y evolves with respect to plastic strain ε_p, initial yield stresses τ_{y0}^{hard}, and hardening moduli H_0^{hard} such that $\tau_y = \tau_{y0} + H\varepsilon_\text{p}$. We set these parameters to

$$\tau_{y0}^{\text{hard}} = 2\tau_{y0}^{\text{soft}} = 0.003E, \quad \text{and} \quad H_0^{\text{hard}} = 2H_0^{\text{soft}} = 0.01E.$$

Total macroscopic deformation gradient

$$\mathbf{F} = \frac{\sqrt{3}}{2}\begin{bmatrix} 0.995 & 0 \\ 0 & -0.995 \end{bmatrix}$$

is applied in 5 load increments.

We solved this problem with the following schemes: the under-integrated SB Fourier-Galerkin scheme with Fourier projection operator from [159, 25], SB scheme with two linear triangular FEs and the FE projection operator from [80], the DB FE scheme with two linear triangular FEs, and the DB FE scheme with one bilinear rectangular FEs. We set the Newton tolerance to $\eta^{\text{NW}} = 10^{-5}$ and (P)CG tolerance to $\eta^{\text{CG}} = 10^{-5}$. We solve three cases with identity $\mathbf{C}^{\text{ref}} = \mathbf{I}$, symmetrized identity $\mathbf{C}^{\text{ref}} = \mathbf{I}_s$ and mean value $\mathbf{C}^{\text{ref}} = \mathbf{C}^{\text{ref}}_{\text{mean}}$ reference materials, in with analogy to Section 4.6.2.

First, the distributions of global plastic strain ε_p obtained for these four approaches are shown in the first row of Fig. 4.10. The regions of details (the second row) uncover the checkerboard patterns in the plastic strain fields of the under-integrated SB Fourier-Galerkin solution (a.2), that are a direct consequence of the oscillating stress field (a.3). The other three schemes, columns (b) to (d), produce solutions without oscillations.

Second, the number of Newton's method steps and the total number of (P)CG iterations needed to solve the problem with these four approaches are shown in Table 4.1. The table highlights the equivalence of our DB scheme and the SB scheme presented by Leute et al. [80], if equivalent discretizations are used.

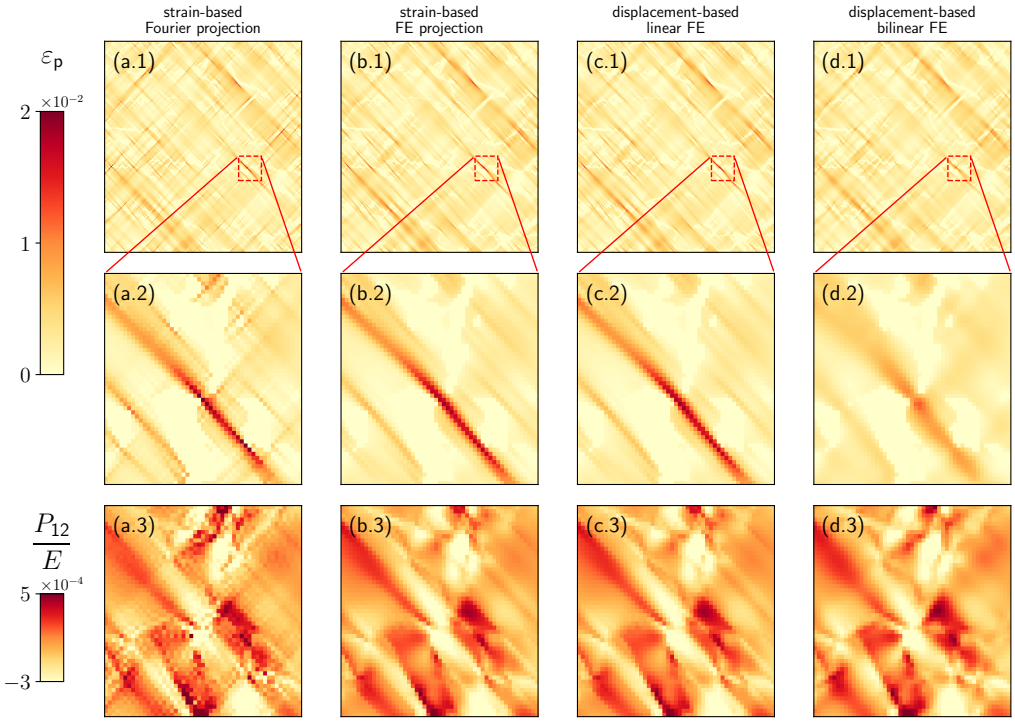


Figure 4.10: Global plastic strains ε_p in dual-phase steel with applied deformation gradient (4.6.3) in row **(1)** with local detials in row **(2)**. Row **(3)** shows accompanying normalized shear stresses P_{12} in detailed area. Discretization schemes in columns: **(a)** the standard SB scheme with Fourier projection operator, **(b)** the SB scheme with FE projection operator with two linear triangular elements, **(c)** the DB FE scheme with two linear triangular elements, and **(d)** the DB FE scheme with one bilinear rectangular elements. All quantities are averaged per pixel.

		strain-based (SB)		displacement-based (DB)	
	$\mathbf{C}^{\text{ref}}$	Fourier projection	FE projection	linear FE	bilinear FE
Newton steps		11	**9**	**9**	10
(P)CG steps	$\mathbf{I}$	1012	**861**	**861**	761
	$\mathbf{I}_s$	781	**609**	**609**	540
	$\mathbf{C}^{\text{ref}}_{\text{mean}}$	585	**457**	**457**	407

Table 4.1: The number of Newton's method steps and the total number of (P)CG steps required to solve the finite-strain elasto-plastic problem of Section 4.6.3 for a three choices of reference material, with Newton tolerance $\eta^{\text{NW}} = 10^{-5}$ and (P)CG tolerance $\eta^{\text{CG}} = 10^{-5}$. Discretization approaches from left to right: the standard SB Fourier-Galerkin scheme with Fourier projection operator, SB scheme with FE projection operator with two linear triangular elements, the DB FE scheme with two linear triangular elements, and the DB FE scheme with one bilinear rectangular element per pixel. Numbers in boldface highlight the equivalence of our DB FE scheme and SB FE scheme presented by Leute et al. [80].

4.7 Comparison with related FFT-based schemes

Several FFT-based computational homogenization schemes exist [129, 84]. An interested reader may therefore find a comparison and placement of our approach in the context of contemporary literature useful.

Recall that our approach is derived from the weak form of the mechanical equilibrium condition (4.2) with an unknown displacement field. The equilibrium (4.2) is discretized in the standard Galerkin manner with the FE basis functions. The nonlinear nodal equilibrium (4.3) is linearized by the Newton's method, and the system of linear equations (4.3.1) is solved by the PCG method. Favourable convergence property of the PCG method is guaranteed by the reference material based preconditioner (4.4.1), which fast application builds on FFT.

4.7.1 The Connection with strain-based approaches

Unlike the DB FE, most spectral methods use strains (gradients) as unknown. SB approaches, like those in [80, 159, 130], typically use the projection operator to enforce the compatibility of strain fields. To reveal a link between the DB and SB approaches, recall the preconditioned scheme (4.4.2),

$$\underbrace{(\boldsymbol{D}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{C}^{\mathrm{ref}}_{(i)}\boldsymbol{D})^{-1}}_{(\boldsymbol{K}^{\mathrm{ref}}_{(i)})^{-1}}\underbrace{\boldsymbol{D}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{C}_{(i)}\boldsymbol{D}}_{\boldsymbol{K}_{(i)}}\delta\tilde{\boldsymbol{u}}_{(i+1)} = -\underbrace{(\boldsymbol{D}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{C}^{\mathrm{ref}}_{(i)}\boldsymbol{D})^{-1}}_{(\boldsymbol{K}^{\mathrm{ref}}_{(i)})^{-1}}\underbrace{\boldsymbol{D}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{\sigma}(\boldsymbol{e}+\boldsymbol{D}\tilde{\boldsymbol{u}}_{(i)},\boldsymbol{g}_{(i)})}_{-\boldsymbol{b}_{(i)}},$$

where we omit the FFTs for simplicity. In the case of linear triangles or tetrahedral elements with a single quadrature point per element, all quadrature weights w^{Q} are equal. Then the multiplication by quadrature weights $\boldsymbol{W}$ can be left out in (4.7.1), leading to

$$\underbrace{(\boldsymbol{D}^{\mathsf{T}}\boldsymbol{C}^{\mathrm{ref}}_{(i)}\boldsymbol{D})^{-1}}_{(\boldsymbol{K}^{\mathrm{ref}}_{(i)})^{-1}}\underbrace{\boldsymbol{D}^{\mathsf{T}}\boldsymbol{C}_{(i)}\boldsymbol{D}}_{\boldsymbol{K}_{(i)}}\delta\tilde{\boldsymbol{u}}_{(i+1)} = -\underbrace{(\boldsymbol{D}^{\mathsf{T}}\boldsymbol{C}^{\mathrm{ref}}_{(i)}\boldsymbol{D})^{-1}}_{(\boldsymbol{K}^{\mathrm{ref}}_{(i)})^{-1}}\underbrace{\boldsymbol{D}^{\mathsf{T}}\boldsymbol{\sigma}(\boldsymbol{e}+\boldsymbol{D}\tilde{\boldsymbol{u}}_{(i)},\boldsymbol{g}_{(i)})}_{-\boldsymbol{b}_{(i)}}.$$

Next, we replace the iterated unknown $\tilde{\boldsymbol{u}}_{(i)}$ with its gradient $\boldsymbol{\partial}\tilde{\boldsymbol{u}}_{(i)}$, recognizing that $\boldsymbol{\partial}\tilde{\boldsymbol{u}}_{(i)} = \boldsymbol{D}\tilde{\boldsymbol{u}}_{(i)}$. After the multiplication with $\boldsymbol{D}$ from the left-hand side, we finally obtain

$$\underbrace{\boldsymbol{D}(\boldsymbol{D}^{\mathsf{T}}\boldsymbol{C}^{\mathrm{ref}}_{(i)}\boldsymbol{D})^{-1}\boldsymbol{D}^{\mathsf{T}}}_{\boldsymbol{\Gamma}^{0}_{(i)}}\boldsymbol{C}_{(i)}\delta\boldsymbol{\partial}\tilde{\boldsymbol{u}}_{(i+1)} = -\underbrace{\boldsymbol{D}(\boldsymbol{D}^{\mathsf{T}}\boldsymbol{C}^{\mathrm{ref}}_{(i)}\boldsymbol{D})^{-1}\boldsymbol{D}^{\mathsf{T}}}_{\boldsymbol{\Gamma}^{0}_{(i)}}\boldsymbol{\sigma}(\boldsymbol{e}+\boldsymbol{\partial}\tilde{\boldsymbol{u}}_{(i)},\boldsymbol{g}_{(i)}),$$

where $\boldsymbol{\Gamma}^{0}_{(i)} : \mathbb{R}^{d_{\mathrm{m}}N_{\mathrm{Q}}} \to \mathbb{R}^{d_{\mathrm{m}}N_{\mathrm{Q}}}$ stands for the discretized periodic Green's operator. Leute et al. [80] described that by setting $\boldsymbol{C}^{\mathrm{ref}}_{(i)} = \boldsymbol{I}_{\boldsymbol{s}}$, $\boldsymbol{\Gamma}^{0}_{(i)}$ projects an arbitrary field from $\mathbb{R}^{d_{\mathrm{m}}N_{\mathrm{Q}}}$ to its closest compatible part in the least square sense with respect to the L^2-norm.

Therefore, this section demonstrates that the schemes (4.7.1) and (4.7.1) are equivalent and generate equivalent solutions in every step of the CG in exact arithmetic. If corresponding stopping criteria are used, CG yields the same approximate solutions. Thus, the only decision-making argument is the possibility of efficient implementation.

4.7.2 The Connection with FEM-FFT approaches

To the best of our knowledge, our method shares the most similarities with the linear hexahedral elements (FFT-Q_1 Hex) formulation by Schneider at al. [131] and Fourier-Accelerated Nodal Solver (FANS) by Leuschner and Fritzen [79]. The novelty of our approach lies in the following:

- *The gradient operator.* Similarly to FFT-Q_1 Hex and FANS, the gradient field is derived with respect to the FE approximation. However, we do not express the discrete gradient operator $\boldsymbol{D}$ in the Fourier space, but keep it in the real space. The direct convolution with a short gradient kernel is cheaper than the Fourier convolution via forward and inverse FFTs. We use the Fourier representation only for the efficient inverse of the preconditioner $\boldsymbol{K}^{\mathrm{ref}}_{(i)}$ as discussed in Section 4.4.

- *Preconditioner and reference material.* Our preconditioner (4.4.1) has the same form as the fundamental solution $\boldsymbol{G}^0$ contained in the discretized periodic Green's operator $\boldsymbol{\Gamma}^0_{(i)}$ of FFT-Q_1 Hex scheme (equation (16) of [131]), and the fundamental solution $\widehat{\phi}$ in FANS (equation (49) of [79]), therefore we expect similar conditioning of all three schemes. However, we provide detailed insight from a linear algebra viewpoint. Direct correspondence between the reference material $\mathbf{C}^{\mathrm{ref}}_{(i)}$, material $\mathbf{C}_{(i)}(\boldsymbol{x})$ and the resulting eigenvalues renders the optimization of $\mathbf{C}^{\mathrm{ref}}_{(i)}$ more accessible. The closer the reference material is to the real material of the sample, the better conditioning the discretized problem has. Therefore, in contrast to [159, 25], we recommend reassembling the preconditioner $\widehat{\boldsymbol{K}}^{\mathrm{ref}}_{(i)}$ when the material tangent significantly changes in Newton's method.

- *Discretization grid.* Both FFT-Q_1 Hex and FANS are developed for bi/trilinear FE basis and quadrilateral/hexahedral elements. Their authors mentioned a possible extension for more complex elements which we present in this paper. In addition, the discretization grid of our method does not have to follow the pixel/voxel structure. We allow for an arbitrary space-filling pattern of elements to be used, recall the patterns in Fig. 4.2. Further extension of our formulation to FE with higher-order polynomial basis functions is therefore straightforward.

- *Computational complexity.* Computational complexity of FFT-based methods is governed by $\mathcal{O}(n \log n)$ complexity of the FFT. However, in our scheme, we compute two FFTs on dN_{n} displacement fields of size N_{I}, instead of d_{m} strain fields of size N_{Q} in FFT-Q_1. Because the number of strain components d_{m} exceeds the number of displacement components dN_{n} per stencil and the number of quadrature points N_{Q} exceeds the number of discretization nodes N_{I}, our method has smaller computational overhead than the DB methods that evaluate the gradient in the Fourier space and perform FFT on the strain-sized fields. For instance, in the case of trilinear hexahedral FEs with 8 quadrature points per element, the saving factor of our method is 24.

4.8 Conclusions

In this paper, we present a novel and *optimal* approach for computational homogenization of nonlinear micromechanical and thermal problems in periodic media. The efficiency is achieved due to a clever interplay between the PCG solver and the geometry and physical properties of the problem [106]. Standard FE discretization on a regular grid is coupled with the Newton's method to handle the nonlinear system iteratively. The linearized system is solved by the PCG method, which is enhanced with a preconditioner based on a discretized inverse (Green's) operator for a problem with spatially uniform reference material data. The proposed matrix-free method exhibits excellent convergence properties as the number of linear solver iterations is bounded independently of the number of discretization nodes and shows mild phase-contrast sensitivity. Our main findings are summarized as follows:

- The condition number associated with the preconditioned linear system decreases as the reference material data approaches the material data. Two-sided bounds for all eigenvalues of the preconditioned linear system are easily accessible and thus provide valuable insight into the choice of the reference material.
- The computational complexity is governed by the FFT algorithm applied to the displacement field. The preconditioning operator is cheaply inverted and applied in Fourier space, while the gradient is evaluated through the convolution with a short kernel in the real space.
- The FE bases produce oscillation-free stress and strain solution fields with marginal discretization artifacts at the phase interfaces. Additional variability of discretization patterns allows the reduction of mesh anisotropy and a more accurate representation of the geometry and the solution.

In addition, the Galerkin nature of the FE method connected with the minimization of the related energy functional allows us to use a well-built theory on the FE method for error estimation, convergence analysis, and other useful tools. In the future, the extension of the equivalence of DB and SB schemes to a general reference material and the fusion of low-rank tensor approximation technique of Vondřejc et al. [152] with our FE scheme are of primal interest.

4.A Thermal conduction

The proposed preconditioned FE method can be used also for potential problems such as thermal conduction or electrostatics. From a mathematical viewpoint, these problems are described by a scalar elliptic partial differential equation.

For the scalar thermal conduction problem, we split the overall temperature gradient $\nabla w : \mathcal{Y} \to \mathbb{R}^d$ into an average temperature gradient $\boldsymbol{e} = \frac{1}{|\mathcal{Y}|}\int_{\mathcal{Y}} \nabla w(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \in \mathbb{R}^d$ and a periodically fluctuating field $\nabla\tilde{w} : \mathcal{Y} \to \mathbb{R}^d$

$$\nabla w = \boldsymbol{e} + \nabla\tilde{w} \quad \text{for all } \boldsymbol{x} \in \mathcal{Y}.$$

Here, $\nabla\tilde{w}$ denotes the temperature gradient, and the fluctuating temperature field $\tilde{w}$ belongs to the space of admissible functions $\mathcal{V} = \{\tilde{v} : \mathcal{Y} \to \mathbb{R},\ \tilde{v} \text{ is } \mathcal{Y}\text{-periodic }\}$. The governing equation for $\nabla\tilde{w}$ follows from the thermal equilibrium condition

$$-\nabla \cdot \boldsymbol{q}(\boldsymbol{x}, \boldsymbol{e} + \nabla\tilde{w}(\boldsymbol{x})) = 0 \quad \text{for all } \boldsymbol{x} \in \mathcal{Y},$$

in which $\boldsymbol{q} : \mathcal{Y} \times \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}^d$ is the flux field. As usual, the equilibrium equation is converted to the weak form

$$\int_{\mathcal{Y}} \nabla\tilde{v}(\boldsymbol{x})^{\mathsf{T}} \boldsymbol{q}(\boldsymbol{x}, \boldsymbol{e} + \nabla\tilde{w}(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x} = 0 \quad \text{for all } \tilde{v} \in \mathcal{V}$$

that serves as the starting point for the FE method. Following the discretization scheme described in Section 4.3, the linearisation in Section 4.3.1 and preconditioning in Section 4.4 leads to a well-conditioned linear system

$$\underbrace{\boldsymbol{F}^H(\boldsymbol{F}\boldsymbol{D}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{A}^{\mathrm{ref}}_{(i)}\boldsymbol{D}\boldsymbol{F}^H)^{-1}\boldsymbol{F}}_{(\boldsymbol{K}^{\mathrm{ref}}_{(i)})^{-1}}\underbrace{\boldsymbol{D}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{A}_{(i)}\boldsymbol{D}}_{\boldsymbol{K}_{(i)}}\delta\tilde{\boldsymbol{w}}_{(i+1)} = \underbrace{\boldsymbol{F}^H(\boldsymbol{F}\boldsymbol{D}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{A}^{\mathrm{ref}}_{(i)}\boldsymbol{D}\boldsymbol{F}^H)^{-1}\boldsymbol{F}}_{(\boldsymbol{K}^{\mathrm{ref}}_{(i)})^{-1}}\underbrace{\boldsymbol{D}^{\mathsf{T}}\boldsymbol{W}\boldsymbol{q}(\boldsymbol{e} + \boldsymbol{D}\tilde{\boldsymbol{w}}_{(i)})}_{-\boldsymbol{b}_{(i)}},$$

for a finite Newton's method increment $\delta\tilde{\boldsymbol{w}}_{(i+1)}$. Material data matrix $\boldsymbol{A}_{(i)} \in \mathbb{R}^{dN_\mathrm{Q}\times dN_\mathrm{Q}}$ stores values of conductivity tangent matrix $\mathbf{A}_{(i)}(\boldsymbol{x}) = \dfrac{\partial \boldsymbol{q}}{\partial \nabla\tilde{w}}(\boldsymbol{x}, \boldsymbol{e} + \nabla\tilde{w}(\boldsymbol{x})) \in \mathbb{R}^{d\times d}$ in (i)-th Newton's method step, and $\boldsymbol{A}^{\mathrm{ref}}_{(i)} \in \mathbb{R}^{dN_\mathrm{Q}\times dN_\mathrm{Q}}$ comes from spatially uniform material data $\mathbf{A}^{\mathrm{ref}}_{(i)} \in \mathbb{R}^{d\times d}$. Another small difference lies in the form of the gradient matrix $\boldsymbol{D}$,

$$\boldsymbol{\nabla}\tilde{\boldsymbol{w}} = \boldsymbol{D}\tilde{\boldsymbol{w}} = \begin{bmatrix}\boldsymbol{D}_1\\ \boldsymbol{D}_2\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{w}}\end{bmatrix}.$$

Here, the entries are the same as in the elasticity problem, recall equation (4.3).

Chapter 5

Elimination of ringing artifacts by finite-element projection in FFT-based homogenization

Abstract: *Micromechanical homogenization is often carried out with Fourier-accelerated methods that are prone to ringing artifacts. We here generalize the compatibility projection introduced by Vondřejc et al. [150] beyond the Fourier basis. In particular, we formulate the compatibility projection for linear finite elements while maintaining Fourier-acceleration and the fast convergence properties of the original method. We demonstrate that this eliminates ringing artifacts and yields an efficient computational homogenization scheme that is equivalent to canonical finite-element formulations on fully structured grids.*

Reproduced from:

[80] R. J. Leute, **M. Ladecký**, A. Falsafi, I. Jödicke, I. Pultarová, J. Zeman, T. Junge, and L. Pastewka. Elimination of ringing artifacts by finite-element projection in FFT-based homogenization. *Journal of Computational Physics*, 453:110931, 2022. DOI: 10.1016/j.jcp.2021.110931

My contribution:

I was involved in software implementation in the open-source C++ library muSpectre, review and editing of the manuscript.

CRediT: Methodology, Software, Writing - Review & Editing

5.1 Introduction

Mechanical homogenization seeks the computation of effective mechanical properties, such as homogenized elastic constants or the complete (potentially nonlinear) stress-strain response, given the microstructure of material and constitutive laws for the individual components. Homogenization often operates on periodic "representative" volume elements that are characteristic for the material under investigation [95, 90]. The intrinsic periodicity of such volume elements suggests the use of a spectral Fourier-basis [17] for the efficient numerical solution of the homogenization problem.

Since the seminal works by Moulinec and Suquet in 1994 [102] and 1998 [103], (fast) Fourier transform (FFT) based homogenization methods were developed rapidly; see, e.g. Refs. [129, 84] for authoritative reviews. The spectral method is here used to solve the partial differential equation (PDE) for a static mechanical equilibrium of a microstructure. The availability of highly optimized FFT implementations (like FFTW [40]) enabled the implementation of efficient and highly parallel spectral solvers which can beat standard implementations of the finite element method (FEM) in speed and accessible system sizes [122].

A major advantage of spectral methods is that it is straightforward to increase the basis set and by this systematically improve the accuracy of the solution. However, because of their global support trigonometric polynomials are not well suited for the solution of PDEs with discontinuous material coefficients or discontinuities in their solution. This leads to a phenomenon known as Gibbs ringing. Gibbs ringing in Fourier spectral methods is well documented, and there are several approaches to suppress the problem [52].

Gibbs ringing persists in FFT-based homogenization methods and was investigated by several authors (see e.g. Refs. [107, 64, 155, 19, 154, 130, 69, 86]). A typically mitigation strategy is the introduction of discrete derivatives: for example, Müller [107] used a finite difference discretization or Willot [154] and Schneider et al. [130] describe a central-difference scheme on a staggered grid. Other FFT-accelerated solution schemes using finite differences [77, 146] or finite elements [131, 79] can be found in the literature. However, published works have mainly focused on natural extensions of the Moulinec-Suquet scheme that is formulated in terms of a reference problem in the displacement space, i.e. for small-strain elasticity in form of the Navier-Lamé equations, that explicitly contain the constitutive law. A solution in Fourier space is then only possible for a homogeneous reference medium. Heterogeneity is superimposed onto this reference solution through Lippman-Schwinger-like approaches for capturing it. Convergence properties of such schemes deteriorate with the strength of the heterogeneity.

In this article we discuss the problem of ringing in FFT-based homogenization methods and present a general solution to it for which we combine the advantages of an FEM-type Galerkin approach with the speed-up of a FFT method. We work in the framework of compatibility projection developed by Lahellec, Vondrejc, Zeman, de Geus and coworkers [75, 150, 25, 25] that considers the deformation gradient as the primary degree of freedom and eliminates the need for an explicit reference medium. Different to previous works, we derive a general expression for a discretized operator for the projection on compatible fields and we propose an elimination rather than a mitigation of the problem for certain operator choices. We show that in addition to Gibbs ringing, the solution with projection techniques is prone to ringing artifacts arising from missing degrees of freedom in the formulation of the deformation gradient.

Our suggested improvements to the method are supported by numerical simulations and comparison with analytical results. We suggest a linear finite-element-based projection

operator that eliminates all ringing artifacts and discuss the analogy of the modified FFT-based homogenization method to standard FEM. All methods are implemented in the open source code μSpectre [60] and the numerical examples shown in Section 5.3 can easily be reproduced by running the corresponding examples of the software.

5.2 Methods

5.2.1 Compatibility projection

We investigate microstructures by a representative volume element (RVE) in a periodic simulation cell Ω_0. The microstructure can consist of different phases which are described by arbitrary small- or finite-strain material models. Here and in the following we will denote first-order tensors (vectors) by arrows and second-order tensors by bold symbols.

Any deformation of the simulation cell can be described by a function (the placement map) $\vec{\chi} : \Omega_0 \to \Omega$, mapping the undeformed grid positions $\vec{r} \in \Omega_0$ into a deformed configuration $\vec{\chi}(\vec{r}) \in \Omega$ where Ω is the deformed periodic simulation domain. The overall goal is to solve for the static mechanical equilibrium of the periodic cell for a given deformation. The static mechanic equilibrium is given by [8]

$$\nabla \cdot \mathbf{P}^T(\mathbf{F}(\vec{r})) = \vec{0} \quad \text{or} \quad \partial_\alpha P_{i\alpha}(\mathbf{F}(\vec{r})) = 0, \tag{5.0}$$

with implicit summation over repeated indices (the Einstein summation convention) and a dot product $(\cdot)$ as defined by the second part of Eq. (5.2.1). Greek indices indicate a tensor dimension representing a derivative, whereas Latin indices are used for all other tensor dimensions. $\mathbf{P}$ is the first Piola-Kirchhoff stress tensor, in general a non-linear function of the deformation gradient

$$\mathbf{F}(\vec{r}) = \nabla\vec{\chi}(\vec{r}) \quad \text{or} \quad F_{i\alpha}(\vec{r}) = \partial_\alpha \chi_i(\vec{r}) \tag{5.0}$$

Note that the partial derivatives $\partial_\alpha \equiv \partial/\partial r_\alpha$ in Eqs. (5.2.1) and (5.2.1) are with respect to the undeformed configuration Ω_0 of the cell.

The most widely used homogeneization schemes combine Eqs. (5.2.1) and (5.2.1), leading to a set of second-order differential equations in $\vec{\chi}$. For small strains, those are the well known Navier-Lamé equations that contain spatial derivatives of the elastic constants. The Green's function of this second order differential equation therefore explicitly contains the constitutive law. Solution with Fourier-techniques then requires the introduction of a homogeneous reference material.

In contrast, the formulation employed here [75, 150, 25, 25] solves the set of first-order differential equations given by Eqs. (5.2.1) and (5.2.1). This leaves the deformation gradient $\mathbf{F}$ as a degree of freedom. Equation (5.2.1) can be interpreted as the constraint that $\mathbf{F}$ needs to be *compatible*, i.e. given by the gradient of a respective placement map. We now solve Eqs. (5.2.1) and (5.2.1) in the subspace of compatible second-order tensors such that the pair of first-order differential equations reduces to the single first-order differential Eq. (5.2.1). This is formulated mathematically by a projection operator $\mathbb{G}$ that maps any second-order tensor onto its compatible part and thereby into the subspace of compatible tensors.

Following Refs. [150, 25, 25], we reformulate Eq. (5.2.1) in the weak (weighted residual) form,

$$\int_{\Omega_0} \mathrm{d}^D r\, \vec{t}(\vec{r}) \cdot \left(\nabla \cdot \mathbf{P}^T\left(\mathbf{F}(\vec{r})\right)\right) = -\int_{\Omega_0} \mathrm{d}^D r\, \left(\nabla \otimes \vec{t}(\vec{r})\right) : \mathbf{P}^T\left(\mathbf{F}(\vec{r})\right) = 0, \tag{5.0}$$

where $\vec{t}(\vec{r})$ is an arbitrary periodic (vector-valued) test function and $\otimes$ the outer product. The symbol D is the dimension of the space, i.e. vectors $\vec{r} \in \mathbb{R}^D$ and second-order tensors $\mathbf{F} \in \mathbb{R}^{D\times D}$. The colon : is the double dot product, a tensor contraction over two indices, $\mathbf{A} : \mathbf{B} = A_{ij}B_{ji}$ with implicit sums over D terms. Surface terms that should appear in Eq. (5.2.1) vanish due to periodicity. If we interpret the test function $\vec{t}(\vec{r})$ as a displacement, then $\delta\mathbf{F}(\vec{r}) = \nabla \otimes \vec{t}(\vec{r})$ is a suitable set of compatible test gradients. We can therefore write the equilibrium condition as

$$\begin{aligned}
&\int_{\Omega_0} \mathrm{d}^D r\, \delta\mathbf{F}(\vec{r}) : \mathbf{P}^T\left(\mathbf{F}(\vec{r})\right) \\
&\quad = \int_{\Omega_0} \mathrm{d}^D r\, \left(\mathbb{G} \star \delta\tilde{\mathbf{F}}\right)(\vec{r}) : \mathbf{P}^T\left(\mathbf{F}(\vec{r})\right) \\
&\quad = \int_{\Omega_0} \mathrm{d}^D r\, \delta\tilde{\mathbf{F}}^T(\vec{r}) : \left(\mathbb{G} \star \mathbf{P}\left(\mathbf{F}\right)\right)(\vec{r}) = 0
\end{aligned}$$

where now $\delta\tilde{\mathbf{F}}$ is an arbitrary (no longer necessarily compatible) test function and $\mathbb{G} \star \mathbf{A}$ denotes the application of the self-adjoint *operator* $\mathbb{G}$ to a right-hand side object $\mathbf{A}$. We now discretize the gradients rather than the displacement field, i.e. in the spirit of the Galerkin method, we express the test gradient $\delta\tilde{\mathbf{F}}$ and the deformation gradient $\mathbf{F}$ within the same basis set.

Equation (5.2.1) no longer contains gradients of the constitutive law, given by $\mathbf{P}(\mathbf{F})$. The compatibility operator $\mathbb{G}$ is clearly independent of it since it just ensures fulfillment of the compatibility condition, i.e. $\nabla \times (\mathbb{G} : \delta\tilde{\mathbf{F}}) = 0$ for finite strain formulations.

The compatibility operator $\mathbb{G}$ is block-diagonal in the Fourier basis, where the blocks are second-order tensors. This leads to the expression [25]

$$\hat{\mathbb{G}}(\vec{k}) : \mathcal{F}\{\mathbf{P}(\mathbf{F})\}(\vec{k}) = \mathbf{0} \quad \text{or} \quad \hat{G}_{i\alpha\beta j}(\vec{k})\mathcal{F}\{P_{j\beta}(\mathbf{F})\}(\vec{k}) = 0, \tag{5.0}$$

where $\hat{\mathbb{G}} \in \mathbb{C}^{D\times D\times D\times D}$ is a fourth-order tensor and $\vec{k}$ the wavevector. We use the hat symbol to denote a Fourier-transformed quantity and $\mathcal{F}\{\cdot\}(\vec{k})$ for the explicit Fourier transform of a quantity as given in 5.A by Eq. (5.A). The numerical solution of Eq. (5.2.1) benefits from the fact that for gradient-based optimizers, the steps of the optimization procedure automatically lead to compatible $\mathbf{F}$, see [25].

The operator $\mathbb{G}$ projects arbitrary tensor fields onto *compatible* fields, i.e. fields that can be expressed as the gradient of a lower order tensor. For enforcing the compatibility for second-order tensor fields, the operator is given by [150]

$$\hat{G}_{i\alpha\beta j}(\vec{k}) = \delta_{ij}\hat{g}_{\alpha\beta}(\vec{k}), \text{ with } \hat{g}_{\alpha\beta}(\vec{k}) = \begin{cases} 0 & \text{if } \vec{k} = \vec{0}, \\ \frac{k_\alpha k_\beta}{k^2} & \text{else}, \end{cases} \tag{5.0}$$

where k_α is the component α of the wavevector $\vec{k}$ and $k = |\vec{k}|$.

5.2.2 Interpreting the projection operator

The projection operator lends itself to a simple interpretation. Let us assume we have an arbitrary vector field $\vec{v}(\vec{r})$. This field is only a gradient $\vec{v}(\vec{r}) = \nabla\phi(\vec{r})$ of a scalar field $\phi(\vec{r})$ if its curl vanishes, $\nabla \times \vec{v}(\vec{r}) = \vec{0}$. If the field is also periodic, we call it a *periodic compatible* field. In the context of homogenization, $\vec{v}$ is a row of the deformation gradient $\mathbf{F}$ and ϕ a component of the placement map $\vec{\chi}$. We are interested in the special case where $\vec{v}(\vec{r})$ is

periodic but $\phi(\vec{r})$ is not. The periodicity of $\vec{v}$ will be later intrinsically fulfilled through the Fourier transform and we only investigate the gradient property here. For non-compatible fields, we search for the scalar field $\phi(\vec{r})$ that minimizes the residual vector

$$\vec{R}(\vec{r}) = \nabla\phi(\vec{r}) - \vec{v}(\vec{r}) \tag{5.0}$$

in a suitable sense (for compatible fields, $\vec{R} \equiv \vec{0}$). For a minimal residual vector, Eq. (5.2.2) is an equivalent formulation of Eq. (5.2.1) and the operator $\mathcal{D}^{-1}$ introduced below is the Green's function of that equation.

The canonical requirement is minimization in the least-squares sense, i.e. minimization of

$$\mathcal{R} = \int_{\Omega_0} d^3r\, \vec{R}(\vec{r}) \cdot \vec{R}(\vec{r}). \tag{5.0}$$

We now need to choose a specific basis set for a series expansion of $\phi(\vec{r})$. In a Fourier basis,

$$\phi(\vec{r}) = \vec{v}_0 \cdot \vec{r} + \frac{1}{N} \sum_{\vec{k}\neq\vec{0}} \hat{\phi}(\vec{k}) \exp\left(\mathrm{i}\vec{k} \cdot \vec{r}\right) \tag{5.0}$$

and an equivalent expansion holds for $\vec{v}$,

$$\vec{v}(\vec{r}) = \frac{1}{N} \sum_{\vec{k}} \hat{\vec{v}}(\vec{k}) \exp\left(\mathrm{i}\vec{k} \cdot \vec{r}\right) \tag{5.0}$$

with $\hat{\vec{v}}(\vec{0}) = \vec{v}_0$ and N the total number of voxels, see 5.A. Note that in Eq. (5.2.2) we have set the (arbitrary) mean value of $\phi(\vec{r})$ to zero but added a linear function that cannot be represented in the Fourier basis whose derivative gives the mean value of Eq. (5.2.2). In terms of homogenization, this mean value describes the affine deformation of the whole RVE and plays the role of the boundary condition of the constituting differential equation.

In the Fourier space the residual becomes

$$\vec{R}(\vec{k}) = \hat{\vec{\mathcal{D}}}(\vec{k})\hat{\phi}(\vec{k}) - \hat{\vec{v}}(\vec{k}) \tag{5.0}$$

for $\vec{k} \neq \vec{0}$ where $\hat{\vec{\mathcal{D}}}(\vec{k}) = \mathrm{i}\vec{k}$ is the Fourier representation of the gradient ∇. For $\vec{k} = \vec{0}$ we obtain $\vec{v}_0 = \hat{\vec{v}}(0)$. Parseval's theorem yields $\mathcal{R} = \sum_{\vec{k}} \vec{R}^*(\vec{k}) \cdot \vec{R}(\vec{k})$, or

$$\mathcal{R} = \sum_{\vec{k}\neq 0} \left(\hat{\vec{v}}^* \cdot \hat{\vec{v}} - \hat{\vec{\mathcal{D}}} \cdot \hat{\vec{v}}^* \hat{\phi} - \hat{\vec{\mathcal{D}}}^* \cdot \hat{\vec{v}} \hat{\phi}^* + \hat{\vec{\mathcal{D}}} \cdot \hat{\vec{\mathcal{D}}}^* \hat{\phi}^* \hat{\phi} \right) \tag{5.0}$$

where the star is the complex conjugate. Note that all symbols in Eq. (5.2.2) – $\hat{\vec{v}}$, $\hat{\vec{\mathcal{D}}}$ and $\hat{\phi}$ – are functions that depend explicitly on $\vec{k}$ but that dependence has been omitted here for brevity. Minimization gives the secular equation

$$\hat{\vec{\mathcal{D}}}(\vec{k}) \cdot \hat{\vec{\mathcal{D}}}^*(\vec{k})\hat{\phi}(\vec{k}) = \hat{\vec{\mathcal{D}}}^*(\vec{k}) \cdot \hat{\vec{v}}(\vec{k}). \tag{5.0}$$

We can solve this for (note that $\vec{k} \neq \vec{0}$)

$$\hat{\phi}(\vec{k}) = \frac{\hat{\vec{\mathcal{D}}}^*(\vec{k})}{\hat{\vec{\mathcal{D}}}(\vec{k}) \cdot \hat{\vec{\mathcal{D}}}^*(\vec{k})} \cdot \hat{\vec{v}}(\vec{k}) \equiv \hat{\vec{\mathcal{D}}}^{-1}(\vec{k}) \cdot \hat{\vec{v}}(\vec{k}) \tag{5.0}$$

where we interpret the term

$$\hat{\vec{\mathcal{D}}}^{-1}(\vec{k}) = \frac{\hat{\vec{\mathcal{D}}}^{*}(\vec{k})}{\hat{\vec{\mathcal{D}}}(\vec{k}) \cdot \hat{\vec{\mathcal{D}}}^{*}(\vec{k})} \tag{5.0}$$

as the inverse of the derivative, i.e. as some form of "integration".

In a next step, we compute the gradient of $\hat{\phi}(\vec{k})$. This yields

$$\hat{\vec{w}}(\vec{k}) = \hat{\vec{\mathcal{D}}}(\vec{k})\hat{\phi}(\vec{k}) = \hat{\vec{\mathcal{D}}}(\vec{k}) \left(\hat{\vec{\mathcal{D}}}^{-1}(\vec{k}) \cdot \hat{\vec{v}}(\vec{k}) \right) = \hat{\mathbf{g}}(\vec{k}) \cdot \hat{\vec{v}}(\vec{k}) \tag{5.0}$$

with

$$\hat{\mathbf{g}}(\vec{k}) = \hat{\vec{\mathcal{D}}}(\vec{k}) \otimes \hat{\vec{\mathcal{D}}}^{-1}(\vec{k}). \tag{5.0}$$

The operator $\hat{\mathbf{g}}(\vec{k})$ hence projects an arbitrary field on its compatible part in the least squares sense with respect to the integral inner product (L^2-norm). The full projection operator for the deformation gradient is given by the first part of Eq. (5.2.1). Using the Fourier derivative $\hat{\vec{\mathcal{D}}}(\vec{k}) = \mathrm{i}\vec{k}$ yields the specific form of the projection operator given in Eq. (5.2.1). Since $\mathbb{G}$ contains the Green's function of Eq. (5.2.1), but *not* of Eq. (5.2.1), it is independent of the constitutive law. The formulation of the projection operator for small-strain elasticity is described in 5.B.

5.2.3 Discrete projection

Rather than using Eq. (5.2.2), we can expand $\phi(\vec{r})$ in other bases of choice. For example, we will below employ linear finite elements. Other discretizations of the gradient operator ∇ with less suitable properties can be obtained through finite-differences methods. We here assume that the simulation cell is structured in a regular (equally spaced) grid with node positions $\{\vec{r}^{IJ}\}$, where I and J are node indices. We will call the individual grid cell a *voxel* and develop the theory in two dimensions, but generalization to three dimensions is straightforward. In two dimensions the position $\vec{r}^{IJ}$ is the lower left corner of voxel IJ. The placement map $\vec{\chi}$ is only known at the nodes which are the corners of the voxels.

The discrete derivative (in some "direction" α) in voxel I, J can generally be written as the convolution

$$\mathcal{D}_\alpha \chi(\vec{r}^{IJ}) = \frac{1}{\Delta_{(\alpha)}} \sum_{ij} s^{ij}_{(\alpha)} \chi(\vec{r}^{I+i,J+j}) \tag{5.0}$$

with $\vec{r}^{I+i,J+j} = \vec{r}^{IJ} + \vec{r}^{ij}$ and $\Delta_{(\alpha)}$ the voxel size in direction α where the round brackets indicate that there is no Einstein sum convention applied for the index (α) and periodicity of $\vec{r}$ in a natural sense is considered. The collection of coefficients s_{ij} is called the *stencil* of the operation. We will introduce different stencils below and note that derivatives in different directions α require different stencils. In the following we will assume that the deformation gradient is described by a set of d derivatives, but that d is not necessarily equal to the dimension D of the system. This allows the subdivision of voxels into multiple evaluation points, i.e., d may be an integer multiple of D. Using n_q evaluation points per voxel leads to $d = n_q D$ derivatives per voxel within the framework that we now develop. The subscript $()_q$ was chosen because these evaluation points correspond to Gaussian quadrature points in a classic finite-element discretization.

By expanding the discrete placement map into a Fourier series of the form given by Eq. (5.A), we can write the derivative operation in Fourier space as

$$\hat{\mathcal{D}}_\alpha(\vec{k}) = \frac{1}{\Delta_{(\alpha)}} \sum_{ij} s^{ij}_{(\alpha)} \exp\left(\mathrm{i}\vec{k} \cdot \vec{r}^{ij}\right). \tag{5.0}$$

Again the case $\vec{k} = \vec{0}$ is special since $\sum_{ij} s^{ij}_{(\alpha)} = 0$. For $\vec{k} \neq \vec{0}$, we can use the same argument as above: A general vector field $\hat{\vec{v}}(\vec{k})$ can be projected (in the least squares sense) onto its compatible part $\hat{\vec{w}}(\vec{k})$ using

$$\hat{w}_\alpha(\vec{k}) = \hat{g}_{\alpha\beta}(\vec{k})\hat{v}_\beta(\vec{k}) \tag{5.0}$$

with

$$\hat{g}_{\alpha\beta}(\vec{k}) = \frac{\hat{\mathcal{D}}_\alpha(\vec{k})\hat{\mathcal{D}}^*_\beta(\vec{k})}{\hat{\mathcal{D}}_\gamma(\vec{k})\hat{\mathcal{D}}^*_\gamma(\vec{k})} = \hat{\mathcal{D}}_\alpha(\vec{k})\hat{\mathcal{D}}^{-1}_\beta(\vec{k}), \tag{5.0}$$

cf. Eq. (5.2.2). We want to mention that by the Helmholtz decomposition $\delta_{\alpha\beta} - g_{\alpha\beta}(\vec{k})$ is a projection to divergence free fields, with applications to error estimation. The projection operator for the deformation gradient is then given by $\hat{G}_{i\alpha\beta j}(\vec{k}) = \delta_{ij}\hat{g}_{\alpha\beta}(\vec{k})$. Note that $\hat{\mathbf{g}} \in \mathbb{C}^{d\times d}$ and $\hat{\mathbb{G}} \in \mathbb{C}^{D\times d\times d\times D}$. The projection operator is idempotent (i.e. a *projection*), since,

$$\begin{aligned}
&\hat{G}_{l\gamma\alpha i}(\vec{k})\hat{G}_{i\alpha\beta j}(\vec{k})\hat{f}_{j\beta}(\vec{k}) \\
&\quad= \delta_{li}\hat{g}_{\gamma\alpha}(\vec{k})\left(\delta_{ij}\hat{g}_{\alpha\beta}(\vec{k})\hat{f}_{j\beta}(\vec{k})\right) \\
&\quad= \delta_{li}\delta_{ij}\hat{\mathcal{D}}_\gamma(\vec{k})\hat{\mathcal{D}}^{-1}_\alpha(\vec{k})\hat{\mathcal{D}}_\alpha(\vec{k})\hat{\mathcal{D}}^{-1}_\beta(\vec{k})\hat{f}_{j\beta}(\vec{k}) \\
&\quad= \delta_{lj}\hat{g}_{\gamma\beta}(\vec{k})\hat{f}_{j\beta}(\vec{k}) \\
&\quad= \hat{G}_{l\gamma\beta j}(\vec{k})\hat{f}_{j\beta}(\vec{k})
\end{aligned}$$

This compatibility operator $\hat{\mathbb{G}}$ is the projection generalized for arbitrary derivative operators $\hat{\vec{\mathcal{D}}}$.

The case $\vec{k} = \vec{0}$ contains the boundary condition and is treated as for the Fourier derivative. For multiple elements, each element needs to hold the same gradient for $\vec{k} = \vec{0}$. We want to emphasize that in this discrete basis, the discrete Fourier transformation has the role of accelerating the convolution rather than providing the basis set for the underlying discretization.

5.2.4 Finite differences and Lanczos-σ correction

Canonical discrete derivative operators are given by finite difference schemes. We here consider the first-order central-differences,

$$\frac{\partial\chi_i}{\partial r_\alpha} = \frac{\chi_i\left(r_j + \delta_{j\alpha}\Delta_{(\alpha)}\right) - \chi_i\left(r_j - \delta_{j\alpha}\Delta_{(\alpha)}\right)}{2\Delta_{(\alpha)}} + \mathcal{O}\left((\Delta_{(\alpha)})^2\right), \tag{5.0}$$

which can be expressed in Fourier space as follows (see e.g. [146]):

$$\hat{\mathcal{D}}^{\mathrm{cd}}_\alpha(\vec{k}) = \frac{\mathrm{i}\sin(k_\alpha\Delta_{(\alpha)})}{\Delta_{(\alpha)}}. \tag{5.0}$$

Remember that $\Delta_{(\alpha)}$ is as before the grid spacing in direction α and there is no summation over indices in parenthesis. We note that this central-differences scheme is related to the Lanczos-σ correction for Gibbs ringing in direction α [55], $\sigma_{(\alpha)}(\vec{k}) = \sin(k_\alpha\Delta_{(\alpha)})/k_\alpha\Delta_{(\alpha)}$. Expressed using the σ-factor, the derivative operator is given by $\mathcal{D}^{\mathrm{cd}}_\alpha(\vec{k}) = \mathrm{i}k_\alpha\sigma_{(\alpha)}(\vec{k})$.

Another common first-order finite difference scheme is forward-differences,

$$\frac{\partial\chi_i}{\partial r_\alpha} = \frac{\chi_i(r_j + \delta_{j\alpha}\Delta_{(\alpha)}) - \chi_i(r_j)}{\Delta_{(\alpha)}} + \mathcal{O}(\Delta_{(\alpha)}) \tag{5.0}$$

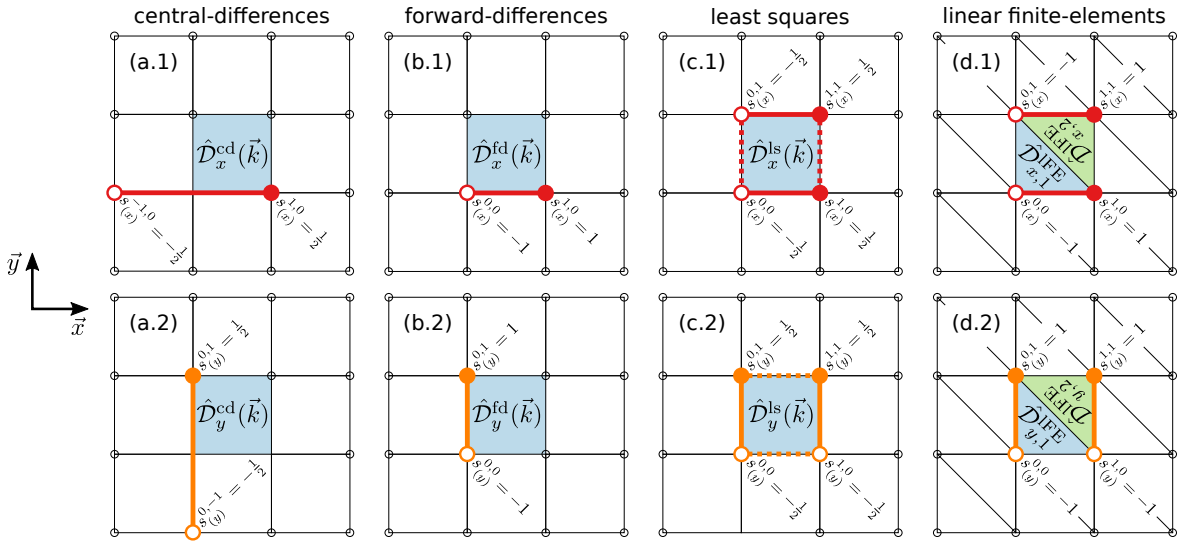


Figure 5.2.1: Graphical representation of the different stencils $s^{ij}_{(x)}$ and $s^{ij}_{(y)}$ for the derivatives in x and y-direction of the discrete derivative operators $\hat{\mathcal{D}}_\alpha(\vec{k})$. Column **(a)** shows central-differences, **(b)** forward-differences, **(c)** least squares and **(d)** linear finite-element stencils. Row **(1)** shows the derivatives in x-direction and **(2)** in y-direction. Computed derivatives are assigned to the voxel marked in blue. For linear finite elements in **(d)**, the voxel is subdivided into two triangles. Full dots indicate positive stencil values and open dots negative ones. Thick lines indicate the direction of the derivative and dotted lines in column **(c)** indicate connected stencils in the non derivative direction. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

with the Fourier-space representation

$$\hat{\mathcal{D}}^{\mathrm{fd}}_\alpha(\vec{k}) = \frac{\exp\left(\mathrm{i}k_\alpha \Delta_{(\alpha)}\right) - 1}{\Delta_{(\alpha)}}. \tag{5.0}$$

The full stencil coefficients for these two schemes as applied to two-dimensional problems are shown in Fig. 5.2.1a and 5.2.1b. Inserting these coefficients into the generic expression Eq. (5.2.3) yields the specific Fourier representations given in Eqs. (5.2.4) and (5.2.4). Figure 5.2.1 also has a graphical representation of these discrete derivatives.

5.2.5 Least squares

We now turn to a purely geometric interpretation of the deformation gradient to derive alternative discrete stencils. The deformation maps an infinitesimal fiber vector $\delta\vec{r}$ into $\delta\vec{r}' = \mathbf{F}(\vec{r}) \cdot \delta\vec{r}$. Assuming constant $\mathbf{F}$ over a given voxel (light blue rectangle in Fig. 5.2.2a), the voxel is affinely deformed by the deformation gradient $\mathbf{F}$ (green parallelogram in Fig. 5.2.2b) and cannot represent arbitrary displacements of the corners (dark blue trapezoid in Fig. 5.2.2b) as long as the deformation gradient $\mathbf{F}$ is uniform on that voxel. In order to represent the corner displacements exactly, we can for instance subdivide the voxel, e.g. in 2D into two triangles (see Fig. 5.2.2c and Fig. 5.2.2d), introducing multiple elements per voxel, with their own uniform deformation gradient per element. We will discuss this decomposition in the next section and will for now continue to work with a uniform deformation gradient per voxel and require matching of the corner displacements in a least-squares sense.

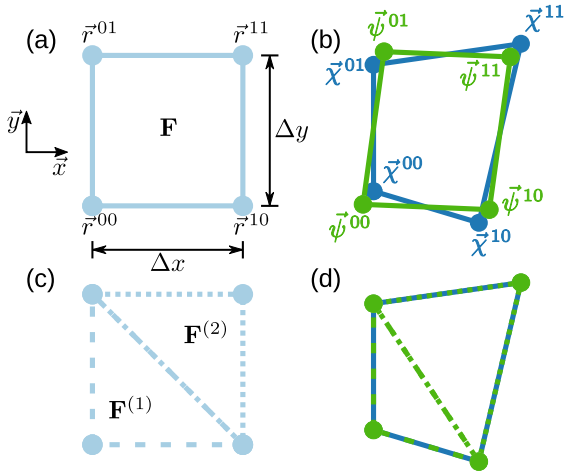


Figure 5.2.2: Non-affine deformation of the single 2D voxel at I=0, J=0. **(a)** The undeformed voxel (light blue) is described by its four corners $\vec{r}_{ij}$ in a grid with grid spacing Δx and Δy. **(b)** A non-affine deformation of this voxel is shown in dark blue. The green parallelogram shows the least-squares approximation of the dark blue trapezoid using a uniform gradient $\mathbf{F}$ throughout the voxel. **(c)** The single voxel of **(a)** is subdivided into two triangles, the deformation of each is described by a uniform deformation gradients $\mathbf{F}^{(1)}$ for the dashed triangle and $\mathbf{F}^{(2)}$ for the dotted triangle. **(d)** Individual affine deformation of these two triangles (green) can perfectly match the prescribed corner displacements (dark blue).

We now regard the displaced corner positions only within a single voxel (see Fig. 5.2.2a). Without loss of generality, we choose the voxel I=0, J=0 with undeformed corner coordinates $\{\vec{r}^{00}, \vec{r}^{01}, \vec{r}^{10}, \vec{r}^{11}\}$. To distinguish this subset of four nodes from the whole grid we name it $\vec{r}^{kl}$ with $k, l \in \{0, 1\}$. The true deformed coordinates are $\vec{\chi}^{kl}$ and the affinely deformed coordinates produced by the deformation gradient $\mathbf{F}$ are $\vec{\psi}^{kl}$. We require that the affinely deformed rectangle matches the true deformation in a least square sense. This means we are looking for the deformation gradient $\mathbf{F}$ that minimizes the residual

$$R = \sum_{k,l} \left| \vec{\psi}^{kl} - \vec{\chi}^{kl} \right|^2 . \tag{5.0}$$

Since the affinely deformed voxel is spanned by the two vectors $\mathbf{F} \cdot \Delta\vec{r}^{10}$ and $\mathbf{F} \cdot \Delta\vec{r}^{01}$ with $\Delta\vec{r}^{kl} = \left(\vec{r}^{kl} - \vec{r}^{00}\right)$, we can write the affine deformed coordinates as

$$\begin{aligned}
\vec{\psi}^{10} &= \vec{\psi}^{00} + \mathbf{F} \cdot \Delta\vec{r}^{10} \\
\vec{\psi}^{01} &= \vec{\psi}^{00} + \mathbf{F} \cdot \Delta\vec{r}^{01} \\
\vec{\psi}^{11} &= \vec{\psi}^{00} + \mathbf{F} \cdot \Delta\vec{r}^{11}.
\end{aligned}$$

We now insert these into Eq. (5.2.5) and minimize with respect to $\mathbf{F}$ and $\vec{\psi}^{00}$. This yields the secular equations

$$\begin{aligned}
\frac{\partial R}{\partial \vec{\psi}^{00}} &= 2 \sum_{k,l} \left(\vec{\psi}^{kl} - \vec{\chi}^{kl} \right) = \vec{0} \\
\frac{\partial R}{\partial \mathbf{F}} &= 2 \sum_{k,l} \left(\vec{\psi}^{kl} - \vec{\chi}^{kl} \right) \otimes \Delta\vec{r}^{kl} = \mathbf{0}
\end{aligned}$$

or

$$\vec{\psi}^{00} = \frac{1}{4}\sum_{k,l}\left(\vec{\chi}^{kl} - \mathbf{F}\cdot\Delta\vec{r}^{kl}\right) \tag{5.0}$$

and

$$\mathbf{F}\cdot\sum_{k,l}\left(\Delta\vec{r}^{kl} - \frac{1}{4}\sum_{m,n}\Delta\vec{r}^{mn}\right)\otimes\Delta\vec{r}^{kl} = \sum_{k,l}\left(\vec{\chi}^{kl} - \frac{1}{4}\sum_{m,n}\vec{\chi}^{mn}\right)\otimes\Delta\vec{r}^{kl}. \tag{5.0}$$

For rectangular lattices with lattice spacing Δx and Δy, this can be solved to give

$$\mathbf{F} = \frac{1}{2}\begin{pmatrix}\frac{\vec{\chi}^{10}-\vec{\chi}^{00}+\vec{\chi}^{11}-\vec{\chi}^{01}}{\Delta x} & \frac{\vec{\chi}^{01}-\vec{\chi}^{00}+\vec{\chi}^{11}-\vec{\chi}^{10}}{\Delta y}\end{pmatrix}. \tag{5.0}$$

The corresponding stencil coefficients are shown in Fig. 5.2.1.

5.2.6 Linear finite elements

The previous section argued that a uniform deformation gradient per voxel is insufficient to represent the voxel's deformation as measured by the displacement of each corner. This becomes evident from simply counting the degrees of freedom: In two dimensions, the voxel's deformation (and rotation) is given by three vectors (6 degrees of freedom) while the deformation gradient has 4 independent components. In this section, the problem is solved by splitting the voxel into two triangles that are each described by a uniform deformation gradient (see Fig. 5.2.2c). The voxel still has 6 degrees of freedom, but now we have 2 deformation gradients with a total of 8 independent components and the constraint that the diagonal boundary between the two triangles remain of the same length and direction (two blocked degrees of freedom). From a simple geometric argument, this triangular decomposition can hence describe arbitrary corner displacements (see Fig. 5.2.2d).

This geometric point of view is fully equivalent to a formulation using linear finite elements. Within our rectangular lattice, we use the usual linear shape functions

$$\begin{aligned}N_{00}^{(1)}(x,y) &= 1 - x/\Delta x - y/\Delta y\\ N_{10}^{(1)}(x,y) &= x/\Delta x\\ N_{01}^{(1)}(x,y) &= y/\Delta y\end{aligned}$$

where the origin of the coordinate system is at the bottom left of the voxel and an equivalent set of shape functions exists for element (2) in Fig. 5.2.2c. The shape function gradient of $\chi(x,y) = \chi_{00}N_{00}^{(e)}(x,y) + \chi_{10}N_{10}^{(e)}(x,y) + \chi_{01}N_{01}^{(e)}(x,y)$ is constant on element (e) and given by the stencil coefficients shown in Fig. 5.2.1d for the two deformation gradients. The stencil for element (1) is identical to the forward-differences scheme. The stencil for element (2) is a forward-differences scheme evaluated on a different set of nodes turning it into a backward differences scheme. Generalizations of this scheme to non-orthogonal voxels and three dimensions are straightforward. This formulation is identical to traditional linear finite elements, but unlike classical finite-elements, the projection formulation yields a condition number that is independent of system size, leading to scale-independent convergence properties [99]. (One can also get a full characterization of the spectrum which can lead to a better estimate of the performance of the conjugate gradient method [45].) Note that the least square approach described in the previous section simply yields the average of the deformation gradients on the two triangles.

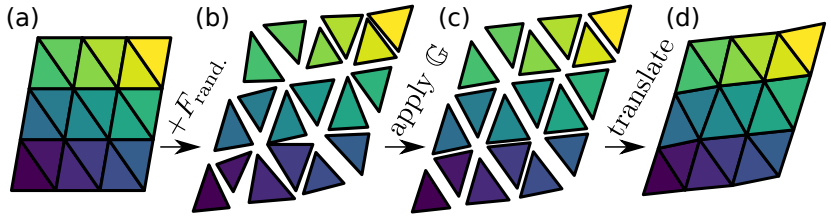


Figure 5.2.3: The suggested projection operator for the finite-element formulation is acting on a random, non-compatible deformation gradient field and projects it onto the respective compatible field. The figure shows **(a)** an undeformed grid **(b)** in which triangular elements are independently randomly deformed and presented in an exploded view. The random deformation is corrected by **(c)** applying the projection operator $\mathbb{G}$ and **(d)** translating from the exploded view into a compact cell where all triangles fit together.

It is important to emphasize that this formulation requires storing two deformation gradients per voxel. In the discrete projection developed above, this means we have $n_q = 2$ evaluation points and $d = 4$ derivatives for the two-dimensional formulation. The deformation gradient $\mathbf{F}$ and Piola-Kirchhoff stress $\mathbf{P}$ are then both elements of $\mathbb{R}^{D\times D\times n_q}$ within the projection scheme. For the evaluation of the constitutive law, both $\mathbf{F}$ and $\mathbf{P}$ need to be decomposed in their element-wise contributions $\mathbf{F}^{(e)}$ and $\mathbf{P}^{(e)}$ that are elements of $\mathbb{R}^{D\times D}$. We can formally write

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}^{(1)} \\ \mathbf{F}^{(2)} \end{pmatrix} \tag{5.0}$$

for this decomposition. We note that finite-element discretizations with multiple quadrature points can be described in this discrete projection using similar decompositions.

The interpretation put forward at the beginning of this section is that the deformation gradient describes the geometry of the respective triangle (see Fig. 5.2.2d). This allows an intuitive interpretation of the projection operator $\mathbb{G}$. We start with a decomposition of some domain into triangles (Fig. 5.2.3a). The rotation and shape of each of these triangles is described by a deformation gradient $\mathbf{F}^{(e)} \in \mathbb{R}^{D\times D}$. We now randomly disturb these triangles by adding a random number to the components of their deformation gradients. The resulting structure (see exploded view in Fig. 5.2.3b) is clearly no longer compatible. Application of $\mathbb{G}$ (Fig. 5.2.3c) adjusts the shape of each triangle such that they are again compatible and can be assembled into a continuous deformed structure (Fig. 5.2.3d).

5.2.7 Even vs. odd number of grid points

The original formulation of the compatibility projection that employs the Fourier derivative only works exactly for odd-sized grids, which is a result of the structure of the projection operator. The Fourier derivative is ambiguous at the Nyquist frequency (or the edge of the first Brillouin zone). This ambiguity originates in the freedom of choosing one of two possible equivalent even numbered Fourier grids $\{k_i\}$ from the class $k_i\Delta \in [-\pi, \pi)$ or $k_i\Delta \in (-\pi, \pi]$, where Δ is the grid spacing. Even sized grids sample the frequency exactly at the Brillouin zone boundary (see Eq. (5.A) for the choice $k_i\Delta \in [-\pi, \pi)$). As a result the two possible grids differ only in one single grid point, the Nyquist frequency.

We first analyze the situation for the Fourier-derivative. The Nyquist frequency is given by $k_{\text{Ny}} = \pm\pi/\Delta$, where an even-sized grid contains either the positive or negative Nyquist frequency. Typically this small difference does not matter for the periodic function $\hat{f}(k)$ because the Fourier coefficients are also equivalent in the single point $\pm k_{\text{Ny}}$ where the two

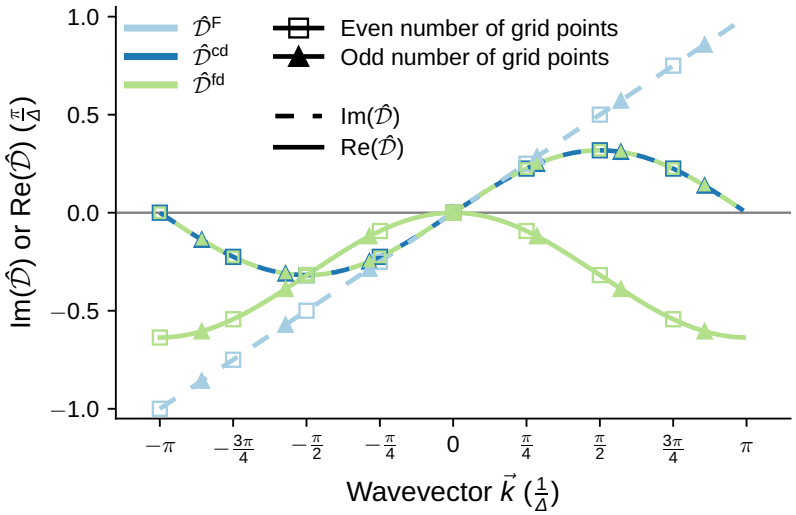


Figure 5.2.4: Real (continuous line) and imaginary (dashed line) parts of the three different derivative operators, $\hat{\mathcal{D}}^{\mathrm{F}}$ (Fourier derivative, light blue), $\hat{\mathcal{D}}^{\mathrm{cd}}$ (central-differences, dark blue) and $\hat{\mathcal{D}}^{\mathrm{fd}}$ (forward-differences, green). The real part of the central-differences and Fourier-type projection operator are zero everywhere and therefore not shown. Data points are shown for an even grid (open squares) with eight points and for an odd grid (triangles) with seven points, see Eq. (5.A). Only the even grid has a data point at the Nyquist frequency, here chosen to be at $-\pi/\Delta$.

grids differ, $\hat{f}(-k_{\mathrm{Ny}})$ from the one grid containing $-k_{\mathrm{Ny}}$ has the same value as $\hat{f}(+k_{\mathrm{Ny}})$ from the other grid containing $+k_{\mathrm{Ny}}$. Thus, the Fourier series does not depend whether the positive or negative Nyquist frequency is included in the grid and is therefore unambiguous for any (sampled) function $f(x)$. However, when analyzing the Fourier series of $\mathcal{D}^{\mathrm{F}} f(x)$, where the upper index "F" indicates the explicit use of the Fourier derivative $\hat{\mathcal{D}}^{\mathrm{F}}(k) = \mathrm{i}k$, evaluated on grid points x,

$$\mathcal{D}^{\mathrm{F}} f(x) = \frac{1}{N} \sum_k \mathrm{i}k \hat{f}(k) \exp(\mathrm{i}kx) = \frac{1}{N} \sum_k S(k, x), \tag{5.0}$$

we find for the summand $S(k_{\mathrm{Ny}}, x)$ at the Nyquist frequency

$$S(k_{\mathrm{Ny}}, x) = \mathrm{i}k_{\mathrm{Ny}} \hat{f}(k_{\mathrm{Ny}}) \exp\left(\mathrm{i}k_{\mathrm{Ny}} x\right) = \mathrm{i}k_{\mathrm{Ny}} \hat{f}(k_{\mathrm{Ny}}) \cos(k_{\mathrm{Ny}} x) \tag{5.0}$$

where we used $k_{\mathrm{Ny}} x \propto \pi n, n \in \mathbb{Z}$. Since $\hat{\mathcal{D}}^{\mathrm{F}}(k_{\mathrm{Ny}}) = \mathrm{i}k_{\mathrm{Ny}} \neq -\mathrm{i}k_{\mathrm{Ny}} = \hat{\mathcal{D}}^{\mathrm{F}}(-k_{\mathrm{Ny}})$ the summand $S(k_{\mathrm{Ny}}, x) \neq S(-k_{\mathrm{Ny}}, x)$ which gives an ambiguity in the Fourier series of $\mathcal{D}^{\mathrm{F}} f(x)$ for even sized grids at the Nyquist frequency whenever $\hat{f}(k_{\mathrm{Ny}}) \neq 0$. The outcome of the Fourier series in Eq. (5.2.7) depends on which one of the two possible even sized grids is taken. This ambiguity vanishes for odd-sized grids, since the function is never evaluated at the Nyquist frequency and there is only a single possible choice for the Fourier grid. Figure 5.2.4 plots $\hat{\mathcal{D}}^{\mathrm{F}}(k)$ at the discrete evaluation points for even and odd-sized grids as an illustration of this problem.

We now analyze the finite-differences (FD) and finite-element (FE) derivative operators. All derivative operators of this type resolve the ambiguity at the Nyquist frequency. However, we have to take care of the spectrum of these operators. The convolution implicit to the derivative can be represented by the real circular (system-)matrix $\mathbf{B}^{IJ}$,

$$\nabla \vec{\chi}^I \approx \sum_J \mathbf{B}^{IJ} \cdot \vec{\chi}^J. \tag{5.0}$$

The relation

$$\nabla\vec{\chi} \approx \mathcal{F}^{-1}\left\{\hat{\boldsymbol{\mathcal{D}}}(k)\cdot\hat{\vec{\chi}}(k)\right\} \tag{5.0}$$

connects the matrix $\mathbf{B}^{IJ}$ to the generalized derivative operator $\hat{\boldsymbol{\mathcal{D}}}(k)$ and by Eq. (5.2.3) also to the stencil coefficients s^{ij} from Sec. 5.2.3. Since the Fourier transform is only used to compute the convolution of Eq. (5.2.7), there is no ambiguity at the Nyquist frequency. However, $\hat{\boldsymbol{\mathcal{D}}}(k)$ must be invertible for the computation of Eq. (5.2.2). This is equivalent to requiring that the rank of the matrix $\hat{\boldsymbol{\mathcal{D}}}^T\hat{\boldsymbol{\mathcal{D}}}$ or equivalent $\mathbf{B}^T\mathbf{B}$ must be $N-1$. While we need to check this explicitly for the finite-differences stencils, it is well known that this is fulfilled for fully-integrated finite-element formulations. This means that there is no ambiguity between even and odd grid points for the finite-element projection presented here.

5.3 Examples and validation

In the following, we describe four two-dimensional examples to demonstrate the methods developed above with a focus on ringing phenomena. In a last example we analyze the convergence behavior of the described methods. First, we investigate a single soft voxel in a uniform hard matrix under biaxial strain. This minimal example already shows strong ringing artifacts in the xy-component of the stress tensor. Second, we analyze a cell with two pillars separated by one layer of voxels with the Young modulus set to zero. This example shows the ability to handle infinite material contrast with vacuum [128, 85] and simulate a free surface despite the intrinsic periodic boundary conditions. Additionally, one of the pillars contains an inhomogeneity which gives rise to ringing artifacts in the original spectral formulation. Third, we test the numeric correctness of the results by investigating an Eshelby inhomogeneity. We compare the results obtained by the FFT-based method against the analytical Eshelby solution, corrected for periodic boundary conditions. Finally, we demonstrate the feasibility of the method for complex constitutive laws on a damage mechanics problem. Additionally we discuss the computational properties of the different methods using the example of a random two-phase material. For all examples, we solve Eq. (5.2.1) with a coupled Newton-Raphson conjugate-gradient solver as also used by other groups [129, 75] and outlined in 5.C.

5.3.1 Single voxel inhomogeneity

A classical continuum mechanics problem is the inhomogeneity, an inclusion of a material in a matrix with different material properties. At the boundary of the inhomogeneity, there is a discrete change in the material properties which usually leads to Gibbs ringing artifacts in spectral methods. As minimal example of such an inhomogeneity, we present a single voxel inhomogeneity (in red) placed in the center of a 17×17 voxel matrix (in green) as shown in Fig. 5.3.1a. The matrix with Young modulus E_{hard} is ten times harder than the inhomogeneity $E_{\text{soft}} = E_{\text{hard}}/10$, and both have the same Poisson ratio of $\nu = 0.33$. The material response is described by an isotropic finite strain linear elastic law as described in Refs. [8, 25].

We apply a biaxial tensile strain of 10% ($F_{xx} = F_{yy} = 0.1$ and $F_{xy} = F_{yx} = 0$) and investigate the shear component P_{xy} of the first Piola-Kirchhoff stress for different implementations of the projection operator. The shear component of the stress shows the strongest ringing artifacts. The normal stress components P_{xx} and P_{yy} behave similar to the shear component, i.e. all components show reduced or eliminated ringing for the discrete projection operators discussed here. Figure 5.3.1 gives an overview of the results where the first row, panels (b.1) to (g.1),

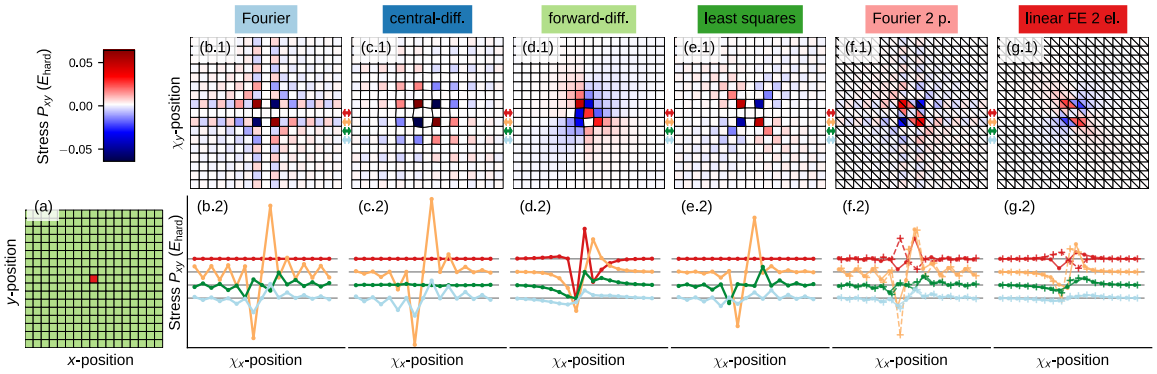


Figure 5.3.1: Soft single voxel inhomogeneity in hard matrix ($E_{\text{hard}} = 10\,E_{\text{soft}}$) at 10% biaxial strain computed on a 17×17 grid. **(a)** shows the phase setup of the central inhomogeneity (red) embedded in the matrix (green). The first row **(x.1)** shows the color coded shear component of the first Piola-Kirchhoff stress P_{xy} on the deformed grid. The second row gives the shear stress along rows of the grid as indicated by the colored double arrows in between the subfigures of the first row. In red the row through the inhomogeneity voxel and then going down row by row in orange, green and blue. The point markers indicate the voxel data and the lines are only added to guide the eye. In the last two columns the continuous line with point markers represents the shear stress of the lower triangle and the dashed line with plus markers represents the upper triangle (see Fig. 5.2.2). The data points are presented at the geometric center of the triangles and voxels. The gray line is the level of zero stress for each row and the y-scaling is the same for all columns **(x.2)** to make a direct comparison possible. The columns represent the data generated with different projection operators as indicated by the column titles: **(b.i)** Fourier, **(c.i)** central-differences, **(d.i)** forward-differences, **(e.i)** least squares, **(f.i)** Fourier type on two evaluation points per voxel and **(g.i)** the linear finite element type projection on two elements per voxel.

show the color coded stress and the second row, panels (b.2) to (g.2), give a more detailed look on the stress along selected rows of the matrix as indicated by the colored arrows in between the subfigures in the first row of Fig. 5.3.1. The lines in the panels (b.2) to (g.2) are ordered from top to bottom starting from the center row, i.e. the row with the inhomogeneity in red. Each column represents the result found with a different projection operator: Column (b) for the Fourier-type projection operator as given by Eq. (5.2.1), (c) for central-differences given by Eq. (5.2.4), (d) for forward-differences given by Eq. (5.2.4), (e) for the least square scheme described in Sec. 5.2.5, (f) for the Fourier-type projection operator on two evaluation points per voxel (see 5.D) and (g) for linear finite elements on two elements per voxel as derived in Sec. 5.2.6.

Panel (b) show the stress field for the original method (e.g. Ref. [25]), with a projection operator based on the Fourier derivative. As expected we observe strong ringing artifacts leading to a checkerboard pattern of the stress field. The stress field and its oscillations are strongest at the inhomogeneity and decay with increasing distance to the discontinuity. However, the ringing does not disappear even at the edges of the cell which was also tested for finer grids, different material contrasts and a slightly inhomogeneous matrix (results not shown). The symmetry of the setup leads to a line of zero stress in the row and column that contains the inhomogeneity (see Fig. 5.3.1b.1 and red line in Fig. 5.3.1b.2).

Results obtained with the central-differences projection operator are shown in Fig. 5.3.1c.1 and c.2. The Gibbs ringing artifacts should be strongly suppressed for this method, however we observe a checkerboard pattern of different style compared to (b). This checkerboard pattern originates in the well-known [37, 121] (odd-even) decoupling into two subgrids over

short distances of the central-differences stencil shown in Fig.5.2.1a. The decoupling of the two grids is not complete because of an odd-sized grid (17×17) and the oscillations decay with increasing distance from the inhomogeneity.

The forward-differences stencil, results shown in panels (d), leads to an oscillation-free but slightly asymmetric stress field which originates from the asymmetry of the forward-differences scheme (see Fig.5.2.1b). This asymmetry is corrected by the least square stencil shown in panels (e). However, the stress has also a checkerboard pattern with checkerboard characteristics of (b.1) and (c.1). We note that the reason for this ringing artifact is neither the Gibbs phenomenon nor the decoupling of two subgrids but the fact that the least-squares derivative cannot represent arbitrary deformations of the voxels as discussed in Sec. 5.2.5. This discussion, and the outcomes from (b) to (e), indicate that a symmetric and ringing free stress field cannot be obtained by a method with a single deformation gradient per voxel.

Therefore, we also investigated projection operators evaluated on two evaluation points per voxel. For the Fourier type projection operator on two evaluation points per voxel, as described in 5.D, we still observe ringing, see panel (f). However, ringing is reduced with respect to the Fourier derivative on a single evaluation point. In panel (f.2) the continuous line represents the stress values in the lower triangle ($P_{xy}^{(1)}$) and the dashed line represents the stress in the upper triangle ($P_{xy}^{(2)}$) (cf. Fig. 5.2.2c). In difference to the stress fields computed with a single evaluation point per voxel the two evaluation points per voxel slightly break the symmetry of the problem. This can be seen in the non-zero stress along the row of the inhomogeneity shown by the red line in (f.2).

Finally, we find a ringing free stress field for the discrete projection operator obtained from linear finite elements on two elements per voxel, panel (g). The asymmetry between the two elements of a voxel discussed in the previous paragraph persists for this formulation. The stress field result presented in panel (g) seems to be the most appropriate solution to the problem due to its smooth, ringing free field. This conclusion will be supported by the following three examples. We would like to again emphasize that ringing in this example does not only result from the Gibbs phenomenon, which does not exist for a discrete projection operator (as is evident for the forward-differences in panel (d)). A description of local deformation with too few degrees of freedom as shown (e.g., the least square type projection in (e)) also gives rise to ringing.

5.3.2 Two pillars and vacuum

We use a setup consisting of two pillars, as shown in Fig. 5.3.2.a, to qualitatively investigate an infinite material contrast and the ability to represent a free surface. This would allow breaking the periodic boundary conditions which are intrinsic to FFT-based methods. We choose a simulation domain of 17×17 voxels and subdivide it into two pillars (in green) with Youngs modulus E_{hard} and Poisson number $\nu = 0.33$. The two pillars are separated by a layer consisting of single voxel of a material with zero stiffness (in light blue), i.e. $E_{\text{vac}} = 0$. One can think of this material as “air” or “vacuum”. At the surface of the pillars we thus have an infinite material contrast. The left pillar, of width 7 voxels, has additionally an inhomogeneity (in red) of three voxels in its center. The soft inhomogeneity has a Youngs modulus of $E_{\text{soft}} = E_{\text{hard}}/10$ and the same Poisson number of $\nu = 0.33$ as the pillar. The inhomogeneity was introduced to generate a non-homogeneous strain field with the ringing artifacts documented in the previous section. The setup is strained by 10% in the y-direction and held at constant size in x-direction. Simulations use the same finite strain model as those of Sec. 5.3.1.

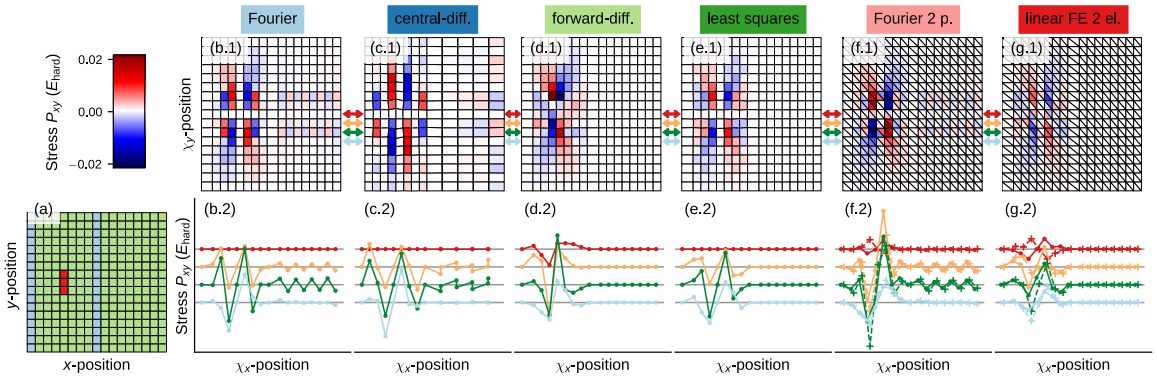


Figure 5.3.2: The first Piola-Kirchhoff shear stress P_{xy} of two pillars at 10% strain in y-direction. **(a)** displays the phase setup. The left pillar (green) has an inhomogeneity (red) of three voxels in its center which are ten times softer than the rest of the pillar. The right pillar (green) is separated by two layers of zero stiffness material (blue) from the left pillar. The first row **(x.1)** presents the shear stress field for the full grid. The second row **(x.2)** presents the shear stress as function of x for selected rows of the simulation grid indicated by colored arrows in the first row **(x.1)**. The rest of the figure is illustrated as described for Fig. 5.3.1.

Figure 5.3.2 is organized in the same manner as Fig. 5.3.1. In the first row, panels (b.1) to (g.1), we show the color coded stress field P_{xy}. In the second row, panels (b.2) to (g.2), we present the stress field as function of χ_x at fixed positions χ_y, starting from the center of the inhomogeneity (red line) and going row by row down in orange, green and light blue. The gray vertical lines represent zero stress for each curve. The continuous and dashed lines in panels (f.2) and (g.2) represent the stress of the lower and upper triangular element of a voxel, respectively. The columns are ordered as in Fig. 5.3.1: (b) Fourier type projection, (c) central-differences, (d) forward-differences, (e) least squares projection, (f) Fourier-type projection on two evaluation points and (g) linear finite-element projection on two elements.

For the Fourier-type projection operator in panel (b), we observe ringing artifacts in the left pillar originating from the inhomogeneity. These artifacts are transmitted through the vacuum region to the right pillar. The shear component of the stress should be zero in the right pillar but shows clear ringing artifacts. The "vacuum" region of zero stiffness is therefore not able to decouple the two pillars. At the symmetry axis in x- and y-direction of the inhomogeneity we can again observe a region of zero stress of single voxel thickness.

Central-differences, panel (c), lead to a strong decoupling of the grid into two sub grids. The vacuum cannot decouple the strain field in the pillars because the stencil has a range of three voxels. This leads to strong oscillations in x-direction in the two subgrids. In the right pillar, that has a width of an even number of grid points in the x-direction, the decoupling results in almost zero width of the voxels of one sub grid. (Only four voxels are clearly visible in panel (c.1).) Remarkably the left pillar has no ringing in y-direction in the columns of non zero stress. In this example, it becomes very clear that the central-differences are sensitive to the setup and number of grid points. For slightly different widths of the pillars or a different mesh grid one can observe these artifacts also in the other pillar (not shown).

Panels (d) and (e) (forward-differences and least-squares), show the same behavior as discussed for a single voxel inhomogeneity in the previous Sec. 5.3.1 with an asymmetric, but non-ringing response for the forward-differences and a symmetric and ringing response for the least squares scheme. However, the right pillar shows zero shear stress, indicating that for these projection operators the vacuum layer is able to decouple the two pillars. The vacuum

layer grows in x-direction to absorb the shrinkage of the two pillars (since $\nu = 0.33$) while the average strain in x-direction is zero. At the surfaces of the left pillar the shear stress field goes to zero as one would expect it for a surface. These discrete derivative schemes decouple the two pillars because their stencil extend only between neighboring nodes.

The Fourier type projection on two evaluation points in panel (f) leads to similar results in the left pillar as for the single voxel inhomogeneity. A ringing artifact from the left pillar is observed in the right pillar which indicates a coupling between the two pillars through the vacuum region. As in the first example, Fig. 5.3.1f, the two evaluation points lead to a slight symmetry breaking resulting in non-zero stress at the symmetry axis through the center of the inhomogeneity in the left pillar, i.e. the red line in panel (f.2).

For the linear finite-element projection shown in panel (g), we again observe artifact-free results. The left pillar shows a smooth, ringing free and symmetric (besides the previous discussed slight asymmetry between upper and lower triangle) stress field originating from the inhomogeneity. The vacuum regions grow in x-direction to absorb the shrinking of the pillars in that direction. In the right pillar we find no artifacts from the stress field of the left pillar; thus the pillars are fully decoupled.

This example shows that it is possible to simulate infinite material contrast with vacuum as the soft phase. A single layer consisting of material of zero stiffness can decouple different regions in the RVE and thus break the intrinsic periodic boundary conditions of the FFT-based method in one direction. As expected from the theoretical considerations in Sec. 5.2 the linear finite-elements projection operator has the best performance and results in a stress field that is artifact free and qualitatively correct. To further investigate the methods developed here, we continue with a quantitative analysis of an Eshelby inhomogeneity.

5.3.3 Eshelby inhomogeneity

The Eshelby inhomogeneity is similar to the first example of a minimal inhomogeneity consisting of a single voxel. The Eshelby inhomogeneity is an ellipsoidal body inside an infinite elastic medium where the elastic medium differs in its material properties from the ellipsoidal body. The analytical solution to the Eshelby problem is well known [33, 34, 105, 92]. We choose the specific (cylindrical) geometry shown in Fig. 5.3.3a with a hard matrix (light green) of Youngs modulus E_{hard} and a soft inhomogeneity (red) of Young's modulus $E_{\text{soft}} = E_{\text{hard}}/10$ and zero eigenstrain. The numerical calculations employ a fine mesh of 151×151 voxels to properly resolve the cylindrical inhomogeneity with half axes of 10% of the domain edge lengths. The inhomogeneity is placed in the center of the domain and centered on a voxel to retain a symmetric discretized area. For the numerical calculations we use the small-strain formulation (see 5.B) since the analytical Eshelby expressions are also obtained in this limit.

The results of these calculations are summarized in Figure 5.3.3. Rows (c.1) to (i.1) show the full solution of the shear strain ε_{xy} on the 151×151 grid. The next row, panels (c.2) to (i.2), show a zoom of the region containing just the inhomogeneity. The three colored lines at the center (dark green), upper half (orange) and lower half (purple) of the inhomogeneity in panel (a) indicate the location of the strain components that are shown in the third row, panels (c.3) to (i.3), for the normal strain ε_{xx} in x-direction at the center and in the fourth row, panels (c.4) to (i.4), for the shear strain ε_{xy} for the upper and lower half of the inhomogeneity. Note that this data is shown only over the zoomed region, not the full calculation. The columns (c) to (h) again represent results obtained for different projection operators as indicated in each column: (c) is the Fourier-type projection, (d) central-differences, (e) forward-differences, (f) the least square type projection, (g) the Fourier type projection on two evaluation points per

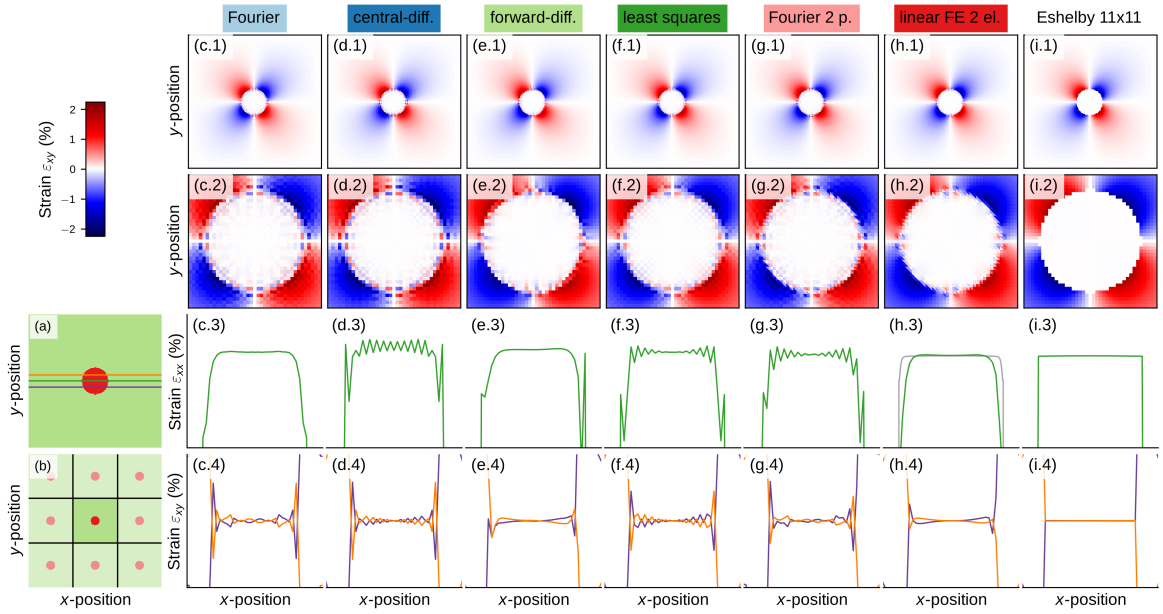


Figure 5.3.3: Shear strain of an Eshelby inhomogeneity at biaxial applied strain $\varepsilon_{xx} = \varepsilon_{yy} = 1\%$ at the boundaries on a 151×151 grid. **(a)** shows the phase setup of a soft inhomogeneity (red) in a hard, $E_{\text{hard}} = 10E_{\text{soft}}$, matrix (green). The colored lines indicate the rows for which the strain is shown in the third and second row. **(b)** illustrates the correction of the analytical Eshelby inhomogeneity for periodic boundary conditions. The strain field of the central inhomogeneity is corrected by adding up the strain field from surrounding inhomogeneities as illustrated here in a 3×3 matrix. The analytical solution presented in column **(i)** was done for a 11×11 matrix, i.e., including five periodic images in both the positive and negative horizontal and vertical direction. In the first row **(x.1)** we present the color coded shear strain ε_{xy} on the whole grid. The second row **(x.2)** is a zoom on the shear strain field at the inhomogeneity with the same color coding. The third and fourth row show the strain as function of x along selected rows of the zoomed region in row two. The third row **(x.3)** shows the normal strain in x-direction along the green row indicated in **(a)**. Row four **(x.4)** presents the shear strain along the middle row of the upper half cylinder (orange) and the middle row through the lower half cylinder (purple) as indicated in **(a)**. The columns present the results gained with the indicated projection operators: **(c.i)** Fourier, **(d.i)** central-differences, **(e.i)** forward-differences, **(f.i)** least squares, **(g.i)** Fourier on two evaluation points per voxel, **(h.i)** linear finite elements on two elements per voxel and the column **(i)** presents the analytical Eshelby solution. In the columns **(g)** and **(h)** the third and fourth row show the strain values in the lower triangle, triangle one in Fig. 5.2.2c. In panel **(h.3)** the additional gray curve shows the convergence of the numerical simulation towards the analytical result for an eleven times finer grid with 1661×1661 grid points.

voxel and (h) the linear finite-elements projection from sec. 5.2.6.

The additional column (i) represent the analytical results of the Eshelby inhomogeneity. The analytical result is obtained for an inhomogeneity in an infinite media at 1% biaxial strain ($\varepsilon_{xx} = \varepsilon_{yy} = 0.01,\ \varepsilon_{xy} = 0$) and is corrected for periodic boundary conditions by summing up the influence of 11×11 non-interacting periodic Eshelby inhomogeneities (see panel (b) of Fig. 5.3.3). This correction scheme for periodic boundary conditions converges quickly with the number of images. The average strain on the central inhomogeneity is used as the boundary conditions for the periodic numerical computations.

The Fourier-type projection operator gives qualitatively correct results. Panels (c.3) and (c.4) show a quantitative view of two components of the strain tensor that both clearly show oscillations within the inhomogeneity. As expected, we observe strong ringing artifacts which

lead to a deviation from the analytical result especially within the inhomogeneity. Note that the normal strain along the center line vanishes due to symmetry reasons and hence does not show ringing.

For the central-differences scheme in panels (d.i), we observe a symmetric checkerboard pattern of decreasing amplitude when approaching the center of the inhomogeneity. At the boundary of the inhomogeneity, there is a double ring-like pattern in the strain field originating from the local decoupling into two subgrids by this scheme.

The forward-differences scheme in column (e) produces ringing-free fields but with the drawback of the already discussed asymmetry, best shown in panel (e.3). However the asymmetry of the field is partly suppressed by working in the small strain limit where $\varepsilon_{xy} = \varepsilon_{yx}$. The asymmetry of the strain field is also noticeable in panel (e.4).

Column (f) shows the results produced by the least squares projection operator. As for the two previous examples we observe a checkerboard like pattern of decreasing intensity when approaching the center of the inhomogeneity (see panel (f.2)).

The Fourier-type projection on two evaluation points per voxel (panel (g)) produces a strain field similar to the standard case with a single point per voxel (panel (c)), however the ringing artifacts appear distributed more homogeneously over the entire inhomogeneity. Panels (g.3) and (g.4) show for clarity only the strain field for the lower triangular element (element one in Figure 5.2.2c). The ringing artifacts in (g.3) and (g.4) are less symmetric than the one of panels (c.3) and (c.4), originating from the symmetry breaking by the triangular mesh.

For the linear finite-element projection on two elements shown in column (h), we find ringing-free fields and the sharpest drop of the strain at the boundary of the inhomogeneity. Panel (h.3) shows the smoothest curves that are close to the analytical solution presented in (i.3). The normal strain shows a small variation across the inhomogeneity while the analytical solution (panel (i.3)) is almost constant. We find similar variation in the shear strain shown in panel (h.4). The gray line in panel (h.3) demonstrates exemplary the convergence of the numeric simulation towards the analytical result for a eleven times finer grid with 1661×1661 grid points. The curves shown in panels (h.3) and (h.4) are closest to the analytical result. We additionally note that, unlike mitigation of oscillations with higher-order finite-differences schemes, it does not appear that the finite-element projection scheme leads to a diffusive solution.

In summary, we find reasonable agreement with the analytical solution for all projection operators. Linear finite elements gives the smoothest curves and results closest to the analytical findings. As in the previous cases, only the forward-differences projection and the finite-elements projection eliminate the ringing artifact. For a finer grid we observe a convergence towards the analytical result. It is worth noting that the similar behaviour of the finite-elements and forward-differences projections is easily explained by the fact that the latter corresponds to the finite-element projection where only the lower left triangles are considered.

5.3.4 Damage problem

As a drawback of FFT-based solution methods, ringing artifacts can have a drastic effect on the solution of a homogenization problem. Damage mechanics problems are especially vulnerable to fluctuations in the stress field caused by ringing artifacts, since localization is one of the most important characteristics of such problems. A reliable, fast, and ringing-free homogenization method is therefore essential to address damage mechanics problems.

In order to illustrate the error introduced by the ringing artifact into a damage problem, we solve a two-dimensional problem representing a concrete microstructure using an Alkali-Silica

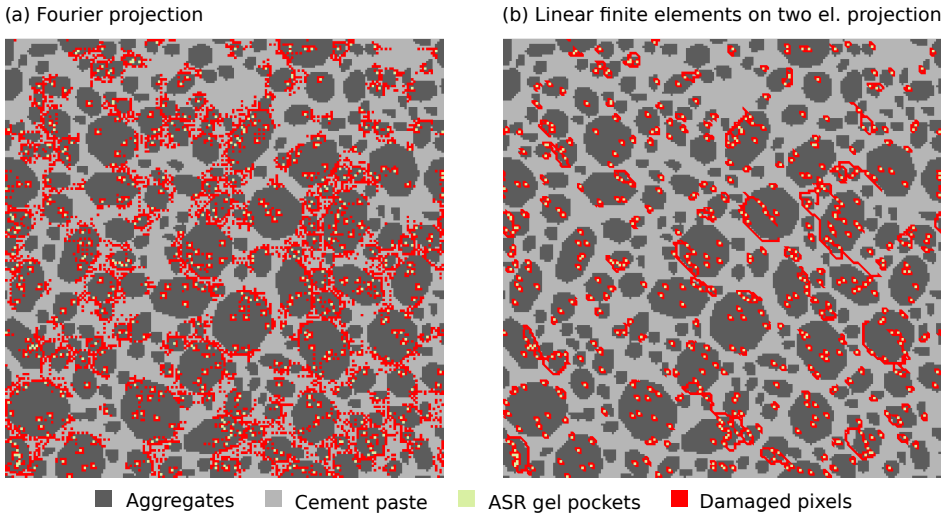


Figure 5.3.4: Damage fields in concrete micro-structure RVEs using (a) Fourier projection solver (b) FEM projection solver.

reaction (ASR) damage model. In this model, the expansion of gel pockets inside aggregates damages the microstructure.

In the modeled concrete microstructure, three phases are considered — a soft matrix with damage (Cement paste), hard inclusions with damage (Aggregates), and gels whose expansion is modeled by a growing spherical eigenstrain. Polygon-shaped aggregates are placed in the RVE using the level set approach (LSA) algorithm [137, 134], with an aggregate size distribution chosen to match sieve sizes from real concrete structures. The aggregate size distribution is truncated on the lower end in order to keep the shape of the aggregates physically sound considering the discretization grid. ASR gel pockets are placed randomly within the aggregate to fill 2% of the cell surface. Concrete paste and aggregate are represented by a linear damage model as their constitutive law [120, 30, 108]. Due to the brittleness of concrete, the damage part of the bi-linear damage laws is taken steeper than its elastic part.

In the damage phase, the damage surface threshold is defined by the magnitude of strain measured by the L_2-norm. As long as the damage material's strain is below a determined ϵ_u, it behaves as a linear elastic isotropic material with Young modulus of E_0 , and afterwards its stress decreases with equivalent stiffness of $-\alpha E_0$ until the stress becomes zero. From that point on, the material does not carry any stress (complete failure). An eigenstrain with the final amplitude of $20\epsilon_u$ was applied on gel voxels, placed as explained beforehand, in 1000 consecutive steps in the carried out simulation. The damage field caused by this loading scenario is depicted in Fig. 5.3.4 employing Fourier and FEM projection solvers. As observed in Fig. 5.3.4 the damage pattern evolved in the Fourier projection solver solution is checker-boarded and therefore non-physical. While as demonstrated in Fig. 5.3.4b, after damage initiation around ASR gel pockets, micro-cracks formed during the damage process coalesce to form cracks with lengths in the range of 0.2 times the RVE length. Comparison of Fig. 5.3.4a, b suggests that, in contrast with Fourier projections solvers, ringing-free spectral FEM projection solvers are capable to simulate mechanics damage RVE problems.

5.3.5 Convergence properties

We analyze the computational properties of the presented methods for a random two-phase system with a simple finite-strain hyper-elastic material as described in Ref. [25, 61]. The

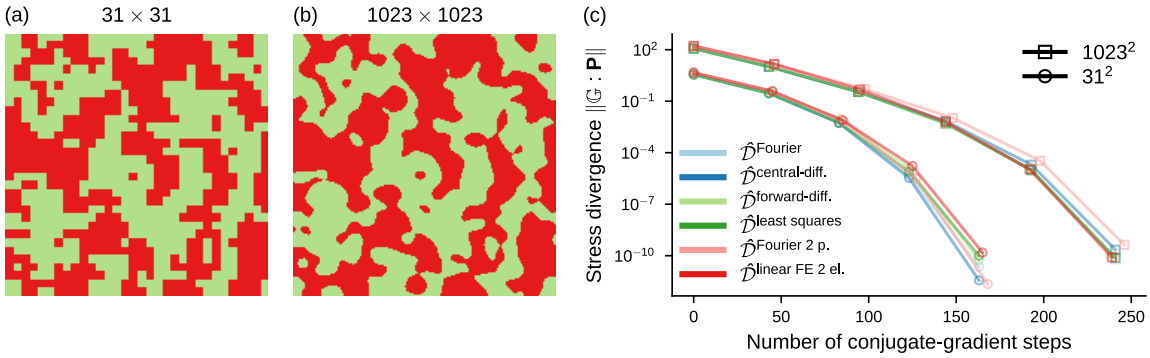


Figure 5.3.5: Convergence properties of a random two-phase system. **(a)** and **(b)** show the random two-phase system for a coarse 31×31 and a fine 1023×1023 grid respectively. The structures consists of a soft phase in red and a hard phase in green with Youngs moduli $E_{\text{hard}} = 10E_{\text{soft}}$ and a Poisson ratio of $\nu = 0.33$. **(c)** shows the convergence of the stress divergence norm $\|\mathbb{G} : \mathbf{P}\|$ with respect to the number of conjugate-gradient steps for the two investigated grid resolutions and all investigated derivative operators ($\hat{\mathcal{D}}^{\text{Fourier}}$ – Fourier derivative, $\hat{\mathcal{D}}^{\text{central-diff.}}$ – central-differences derivative, $\hat{\mathcal{D}}^{\text{forward-diff.}}$ – forward-differences derivative, $\hat{\mathcal{D}}^{\text{least squares}}$ – least squares derivative, $\hat{\mathcal{D}}^{\text{Fourier 2 p.}}$ – Fourier derivative on two evaluation points and $\hat{\mathcal{D}}^{\text{linear FE 2 el.}}$ – linear finite-elements derivative on two elements). The data points represent the value at each Newton-Raphson step and lines are only added to guide the eyes.

system is constructed from slit islands [88] of a two-dimensional random self-affine Gaussian field with Hurst exponent H=0.2, generated as described in Refs. [119, 58]. Self-affine scaling is cutoff at $0.3L$ at large distances and $0.07L$ at small distances, where L is the lateral (periodic) dimension of the cell. This slit island analysis yields a two-phase system with equal concentration of both phases. The two-phase system is generated on a 1023×1023 grid which we then downsample to 31×31 to vary resolution. This leads to the phase setups shown in Fig. 5.3.5 (a) and (b). Panel (c) of Fig. 5.3.5 displays the convergence of $\|\mathbb{G} : \mathbf{P}\|$ (the norm of the divergence of the stress) summed over all pixels. All projection operators show similar convergence properties for the two investigated grids. For the smaller grid with 31×31 voxels, the simulation starts with $\|\mathbb{G} : \mathbf{P}\| \approx 3.5$ for the projection operators forming a rectangular mesh (Fourier, central-differences, forward-differences and least square) and with $\|\mathbb{G} : \mathbf{P}\| \approx 4.7$ for a triangular mesh (Fourier with two evaluation points, linear finite elements) with twice the number of evaluation points. All projection methods need four Newton steps of about 41 conjugate-gradient (CG) steps per Newton step to converge. For the larger grid with 1023×1023 voxels the initial norm of the stress divergence is higher by about a factor 35 compared to the smaller grid. Because of the higher initial norm of the stress divergence, we need five Newton steps to converge to the required tolerance and about 48 CG steps per Newton iteration. However, this is a small increase of about 50% in total amount of CG steps compared to an increase in the number of voxels by a factor 10^3. In summary, we observe a similar behavior for all methods presented here.

5.4 Summary & Conclusion

We have extended the compatibility projection of Lahellec, Vondřejc, Zeman, de Geus and coworkers [75, 150, 25, 25] to standard finite-differences and finite-element basis sets. We show that using linear finite elements as a basis set eliminates ringing artifacts of the Fourier

basis but retains the advantages of the compatibility projection, such as the rapid convergence rate. In particular, this formulation allows perfect decoupling of periodic images with zero-stiffness regions of single-voxel width. This opens the method to the study of free surfaces or metamaterials (see also Refs. [128, 85]). We note that the ideas behind compatibility projection can be exploited to construct preconditioners for standard finite-element formulations beyond the linear elements presented here [100].

Among the projection operators examined in this paper, all but the least squares operator guarantee a compatible strain field. It is useful to think of those compatible projection operators in two categories; discretizations with local support and discretizations with global support, where local support refers to non-overlapping projection stencils. We observe that only two local support projections, the finite-element projection and the forward-differences projection eliminate ringing, unlike the Fourier and central-differences operators (global support and non-local support including neighbouring voxels). This observation allows a well known property of finite-element calculations, where discretization errors are to be expected to be significant at the length scale of the element size (= support size) by St. Venant's principle. This observation allows to interpret the ringing artifact as a discretization error to be expected at the length scale of the support.

5.A Discrete Fourier transformation

The 2D discrete Fourier transformation is defined as follows through out the paper. The generalization to 1D or 3D is straight forward. We divide the simulation domain of edge lengths L_x, L_y into N_x, N_y voxels of equal edge lengths $\Delta_i = L_i/N_i$ in each spatial direction $i \in \{x, y\}$. The lower left corner of voxel $n_x = I, n_y = J$ is then given by

$$r_i^{IJ} = \frac{L_i}{N_i} n_i, \quad n_i = 0, 1, \ldots, N_i - 1. \tag{5.0}$$

The corresponding wave vectors $\{\vec{k}^{IJ}\}$ with $m_x = I, m_y = J$ are

$$\begin{aligned} k_i^{IJ} =& \frac{2\pi}{L_i} m_i, \\ m_i =& \begin{cases} -\frac{N_i}{2}, \ldots, 0, 1, \ldots, \frac{N_i}{2} - 1\ , & N_i \text{ even}, \\ -\frac{N_i-1}{2}, \ldots, 0, 1, \ldots, \frac{N_i-1}{2}\ , & N_i \text{ odd}. \end{cases} \end{aligned}$$

For the discrete Fourier transform of the function $\vec{f}(\vec{r})$ we use

$$\mathcal{F}\left(\vec{f}(\vec{r})\right)(\vec{k}) = \hat{\vec{f}}(\vec{k}) = \sum_{\vec{r}} \vec{f}(\vec{r}) \exp\left(-\mathrm{i}\vec{k}\cdot\vec{r}\right) \tag{5.-1}$$

with the corresponding inverse transformation

$$\mathcal{F}^{-1}\left(\hat{\vec{f}}(\vec{k})\right)(\vec{r}) = \vec{f}(\vec{r}) = \frac{1}{N} \sum_{\vec{k}} \hat{\vec{f}}(\vec{k}) \exp\left(\mathrm{i}\vec{k}\cdot\vec{r}\right). \tag{5.-1}$$

5.B Small-strain projection

It is often useful to carry out calculations in the small-strain limit. Small strains are special because the strain tensors (that replaces the deformation gradient) has to remain symmetric.

For a formulation that involves multiple elements per voxel, we also require multiple symmetric strain tensors per voxel. While in the finite strain formulation, these were absorbed in our derivative indices α, β etc., we need to introduce a specific element index for the small-strain case and can no longer distinguish between derivatives and Cartesian coordinates because of the symmetry in between derivatives and coordinates. Additionally to the previous introduced indices we will therefore use capital Greek letters (Θ, Λ, Ξ, ...) to denote elements. Components of the strain tensor will be denoted by small Latin indices.

We now introduce the strain tensor $\mathbf{e}$ in lieu of the deformation gradient, Eq. (5.2.1). The strain tensor for element Θ is

$$\mathbf{e}_\Theta = \frac{1}{2}\left[\nabla_\Theta \otimes \vec{u} + (\nabla_\Theta \otimes \vec{u})^T\right], \tag{5.-1}$$

where ∇_Θ is the (potentially discrete) derivative operator for element Θ and $\vec{u}(\vec{r}) = \vec{\chi}(\vec{r}) - \vec{r}$ are the displacements from the undeformed positions $\vec{r}$. For a given $\mathbf{e}$, we minimize the residual $\mathcal{R} = \sum_\Theta \sum_{\vec{k}} \mathbf{R}^*_\Theta(\vec{k}) : \mathbf{R}_\Theta(\vec{k})$ with

$$\mathbf{R}_\Theta(\vec{k}) = \frac{1}{2}\left(\hat{\vec{\mathcal{D}}}_\Theta(\vec{k}) \otimes \hat{\vec{u}}(\vec{k}) + \hat{\vec{u}}(\vec{k}) \otimes \hat{\vec{\mathcal{D}}}_\Theta(\vec{k})\right) - \hat{\mathbf{e}}_\Theta(\vec{k}) \tag{5.-1}$$

with respect to $\vec{u}^*$, where we have transformed into the Fourier-space and introduced the gradient operator $\hat{\vec{\mathcal{D}}}_\Theta(\vec{k})$. The full residual is given by

$$\begin{aligned} 2\mathcal{R} = \sum_\Theta \Big(& \hat{\vec{\mathcal{D}}}^*_\Theta \cdot \hat{\vec{\mathcal{D}}}_\Theta \, \hat{\vec{u}}^* \cdot \hat{\vec{u}} + \hat{\vec{\mathcal{D}}}^*_\Theta \cdot \hat{\vec{u}} \, \hat{\vec{u}}^* \cdot \hat{\vec{\mathcal{D}}}_\Theta \\ & - \hat{\vec{\mathcal{D}}}^*_\Theta \cdot \mathbf{e}_\Theta \cdot \hat{\vec{u}}^* - \hat{\vec{u}}^* \cdot \mathbf{e}_\Theta \cdot \hat{\vec{\mathcal{D}}}^*_\Theta \\ & - \hat{\vec{\mathcal{D}}}_\Theta \cdot \mathbf{e}^*_\Theta \cdot \hat{\vec{u}} - \hat{\vec{u}} \cdot \mathbf{e}^*_\Theta \cdot \hat{\vec{\mathcal{D}}}_\Theta + 2\mathbf{e}^*_\Theta : \mathbf{e}_\Theta \Big). \end{aligned}$$

Minimization yields

$$\left(\mathbf{1} + \sum_\Theta \hat{\mathbf{g}}_{\Theta\Theta}\right) \cdot \hat{\vec{u}} = \sum_\Theta \left(\hat{\vec{\mathcal{D}}}^{-1}_\Theta \cdot \hat{\mathbf{e}}_\Theta + \hat{\mathbf{e}}_\Theta \cdot \hat{\vec{\mathcal{D}}}^{-1}_\Theta\right) \tag{5.-1}$$

with

$$\hat{\vec{\mathcal{D}}}^{-1}_\Theta = \frac{\hat{\vec{\mathcal{D}}}^*_\Theta}{\sum_\Lambda \hat{\vec{\mathcal{D}}}_\Lambda \cdot \hat{\vec{\mathcal{D}}}^*_\Lambda} \quad \text{and} \quad \hat{\mathbf{g}}_{\Theta\Lambda} = \hat{\vec{\mathcal{D}}}_\Theta \otimes \hat{\vec{\mathcal{D}}}^{-1}_\Lambda. \tag{5.-1}$$

Equation (5.B) can be formally solved to give

$$\hat{\vec{u}} = \hat{\mathbf{h}} \cdot \sum_\Theta \left(\hat{\vec{\mathcal{D}}}^{-1}_\Theta \cdot \hat{\mathbf{e}}_\Theta + \hat{\mathbf{e}}_\Theta \cdot \hat{\vec{\mathcal{D}}}^{-1}_\Theta\right). \tag{5.-1}$$

or

$$\hat{u}_i = \sum_\Theta \left(\hat{D}^{-1}_{\Theta,l}\hat{h}_{im} + \hat{h}_{il}\hat{D}^{-1}_{\Theta,m}\right)\hat{e}_{\Theta,lm}. \tag{5.-1}$$

with

$$\hat{\mathbf{h}} = \left(\mathbf{1} + \sum_\Theta \hat{\mathbf{g}}_{\Theta\Theta}\right)^{-1}. \tag{5.-1}$$

To arrive at the projection operator, we now need to insert Eq. (5.B) into Eq. (5.B). This yields the small-strain projection operator

$$\hat{G}_{\Theta\Lambda,ijlm} = \frac{1}{2}\left(\hat{g}_{\Theta\Lambda,il}\hat{h}_{jm} + \hat{g}_{\Theta\Lambda,im}\hat{h}_{jl} + \hat{g}_{\Theta\Lambda,jl}\hat{h}_{im} + \hat{g}_{\Theta\Lambda,jm}\hat{h}_{il}\right), \tag{5.-1}$$

where the projected strains are given by

$$\hat{e}_{\Theta,ij} = \sum_{\Lambda} \hat{G}_{\Theta\Lambda,ijlm}\hat{e}_{\Lambda,ml}. \tag{5.-1}$$

By combining the pairs of indices $\alpha = \Theta, j$ and $\beta = \Lambda, l$, we can write this in the same form as the large strain projection, $\hat{e}_{i\alpha} = \hat{G}_{i\alpha\beta j}\hat{e}_{j\beta}$. Note that for a single element $\Theta = 1$, we can write down the expression for $\hat{\mathbf{h}}$ analytically,

$$\hat{\mathbf{h}} = \mathbf{1} - \frac{1}{2}\hat{\mathbf{g}}_{11}. \tag{5.-1}$$

Using this expression and the Fourier derivative for $\hat{\vec{\mathcal{D}}}(\vec{k})$ gives the small-strain projection operator of Ref. [97, Section 6].

5.C Algorithm & implementation

All described methods are implemented in the open source software μSpectre [60]. We follow recent works [25] in solving Eq. (5.2.1) by a Newton-Raphson scheme coupled with a conjugate-gradient solver. The algorithm is described in detail in the panel Algorithm 5.1.

Algorithm 5.1 Solve Eq. (5.2.1), $\mathbb{G} : \mathbf{P}(\mathbf{F}) = 0$ for $\mathrm{N_{li}}$ load increments $\Delta\mathbf{F}_j$

1: **Initialize:**
2: $\eta_{\mathrm{eq.}}$, η_{NR}, η_{CG} ▷ equilibrium-, Newton-Raphson- and CG-tol.
3: $i_{\mathrm{NR,max}}$, $i_{\mathrm{CG,max}}$ ▷ max. iterations Newton-Raphson and CG
4: $\Delta\mathbf{F}_0, \ldots, \Delta\mathbf{F}_{\mathrm{N_{li}}}$ ▷ load increments
5: $\mathbf{F}_0 = \mathbf{1}$ or $\mathbf{0}$ ▷ finite-strain/small-strain initial state
6:
7: **for** $j = 0, 1, 2, \ldots, \mathrm{N_{li}}$ **do** ▷ load incremental loop
8: $\quad\mathbf{F}_0 = \mathbf{F}_0 + \Delta\mathbf{F}_j$ ▷ update initial state by load increment
9: $\quad b_0 = -\mathbb{G} : \mathbf{P}(\mathbf{F}_0)$
10: $\quad$**if** $\|b_0\| \leq \eta_{\mathrm{eq.}}$ **then** ▷ the problem was homogeneous
11: $\quad\quad$Proceed from line 7 ▷ next load increment
12: $\quad$**end if**
13: $\quad$**for** $i = 0, 1, 2, \ldots, i_{\mathrm{NR,max}}$ **do** ▷ Newton-Raphson iteration
14: $\quad\quad$**Solve for $\delta\mathbf{F}$ with conjugate-gradient:**
15: $\quad\quad\quad\mathbb{G} : \mathbb{K}(\mathbf{F}_i) : \delta\mathbf{F} = b_i$ in $i_{\mathrm{CG,max}}$ steps to accuracy η_{CG}
16: $\quad\quad\mathbf{F}_{i+1} = \mathbf{F}_i + \delta\mathbf{F}$
17: $\quad\quad b_{i+1} = -\mathbb{G} : \mathbf{P}(\mathbf{F}_{i+1})$
18: $\quad\quad$**if** $\|b_{i+1}\| \leq \eta_{\mathrm{eq.}}$ **or** $\|\delta\mathbf{F}\| / \|\mathbf{F}_{i+1}\| \leq \eta_{\mathrm{NR}}$ **then**
19: $\quad\quad\quad$Proceed to line 22 ▷ Newton-Raphson converged
20: $\quad\quad$**end if**
21: $\quad$**end for**
22: $\quad\mathbf{F}_0 = \mathbf{F}_{i+1}$ ▷ new initial state
23: **end for**

Note that $\mathbb{K}$ in the conjugate-gradient solver represents the local tangent stiffness. For the Newton-Raphson solver we use two stopping criteria in line 18 of Algorithm 5.1, of which only one has to be fulfilled for convergence. The first criteria is measuring the convergence of the stress divergence and the second one is evaluating the convergence of the Newton-Raphson steps. In line 10 we detect a trivial homogeneous material behaviour by measuring the stress divergence which makes the Newton-Raphson solver redundant for the specific load increment. The computational complexity for all projection operators is dominated by the FFT evaluation in Eq. (5.2.1) and thus given by $\mathcal{O}\left(N \log(N)\right)$ where N is the number of voxels. A discussion of performance and details of the algorithm will be presented in an upcoming paper [100].

5.D Fourier-type projection operator on two evaluation points per voxel

The Fourier-type derivative can be extended to several gradient evaluation points per voxel. This is useful to investigate the influence of Gibbs ringing separately from the effect of missing degrees of freedom. The standard Fourier-type derivative is evaluated at the grid points of the Fourier grid, which are the centers of the voxels. Hence, for a triangular mesh we additionally evaluate the Fourier derivative at the geometrical center of each triangle. For a two dimensional rectangular cell of edge lengths Δ_i, as shown in Fig. 5.2.2c, that means applying a shift of $\pm(\Delta_1, \Delta_2)/6$ from the center. The Fourier derivative operator $\hat{\mathcal{D}}_\alpha(\vec{k}) = \mathrm{i}k_\alpha$ acquires a phase to yield

$$\hat{\mathcal{D}}_{1,i}(\vec{k}) = \mathrm{i}k_i \exp\left(\frac{-\mathrm{i}}{6}\sum_i k_i\Delta_i\right),$$
$$\hat{\mathcal{D}}_{2,i}(\vec{k}) = \mathrm{i}k_i \exp\left(\frac{+\mathrm{i}}{6}\sum_i k_i\Delta_i\right),$$

where we used explicit the indices 1 and 2 to denote the derivative operator in the center of the lower and upper triangle as shown in Fig. 5.2.2.

Chapter 6

FFT-based homogenisation accelerated by low-rank tensor approximations

Abstract: *Fast Fourier transform (FFT) based methods have turned out to be an effective computational approach for numerical homogenisation. In particular, Fourier-Galerkin methods are computational methods for partial differential equations that are discretised with trigonometric polynomials. Their computational effectiveness benefits from efficient FFT based algorithms as well as a favourable condition number. Here these kind of methods are accelerated by low-rank tensor approximation techniques for a solution field using canonical polyadic, Tucker, and tensor train formats. This reduced order model also allows to efficiently compute suboptimal global basis functions without solving the full problem. It significantly reduces computational and memory requirements for problems with a material coefficient field that admits a moderate rank approximation. The advantages of this approach against those using full material tensors are demonstrated using numerical examples for the model homogenisation problem that consists of a scalar linear elliptic variational problem defined in two and three dimensional settings with continuous and discontinuous heterogeneous material coefficients. This approach opens up the potential of an efficient reduced order modelling of large scale engineering problems with heterogeneous material.*

Reproduced from:

[152] J. Vondřejc, D. Liu, **M. Ladecký**, and H. G. Matthies. FFT-based homogenisation accelerated by low-rank tensor approximations. *Computer Methods in Applied Mechanics and Engineering*, 364:112890, 2020. DOI: 10.1016/j.cma.2020.112890

My contribution:

I was involved in software implementation into a Python open-source library FFTHomPy, investigation of numerical behavior of the method, creation of results used in the publication, writing the first draft of the Section 6.4 on numerical examples, and review and editing of the whole manuscript.

CRediT: Methodology, Software, Investigation, Visualization, Writing - Review & Editing

6.1 Introduction

FFT-based methods. A fast Fourier transform (FFT) based method has been introduced as an efficient algorithm for numerical homogenisation in 1994 by Moulinec and Suquet [102]. The method, that has application in multiscale problems, represents an alternative discretisation approach to the finite element method. The effectiveness of FFT-based homogenisation relies on the facts that the system matrix is never assembled, the matrix-vector product in linear iterative solvers is provided very efficiently by FFT, and the condition number is independent of discretisation parameters.

Since the seminal paper in 1994 the methodology has been significantly developed. Originally the approach has been based on Lippmann-Schwinger equation, which is a formulation incorporating Green's function for an auxiliary homogeneous problem. Its connection to a standard variational formulation has been discovered in [150] by using the fact that Green's function is a projection on compatible fields (i.e. gradient fields in scalar elliptic problems), see [96]. It has allowed to fully remove the reference conductivity tensor from the formulation, and interpreted the method from the perspective of finite elements also in nonlinear problems [159, 25]. Moreover, the standard primal-dual variational formulations allow to compute guaranteed bounds on effective material properties [151], which provides tighter bounds than the Hashin-Shtrikman functional.

Significant attention has been focused on developing discretisation approaches that justify the original FFT-based homogenisation algorithm. Many efforts have been made on discretisation with trigonometric polynomials, starting with [158] and followed by [151, 147, 159, 126]. Other discretisation approaches are based on pixel-wise constant basis functions [19, 18], linear hexahedral elements [131], or finite differences [154, 155]. The variational formulations also allowed to derive convergence of approximate solutions to the continuous one [150, 126, 19].

The various discretisation approaches have been studied along with linear and non-linear solvers [36, 104, 158, 98, 18, 61, 159, 25, 127]. Other research directions focus, for example, on multiscale methods [70, 49, 28], highly non-linear problems in solid mechanics [10, 24, 16, 132], and parameter estimation features FFT and model reduction [42].

Low-rank approximations. The general idea of low-rank approximations is to express or compress tensors with fewer parameters, which can lead to a huge reduction in requirements for computer memory and possible significant computational speed-up. For matrices as second order tensors, the optimal low-rank approximation in mean square sense is based on the truncated singular value decomposition (SVD). A computationally cheaper choice is Cross Approximation [51, 7] which has only linear complexity in matrix size N. Low-rank formats or tensors of order larger than two include the canonical polyadic (CP), Tucker, and hierarchical schemes such as the tensor train and the quantic tensor-train form of [54, 72]. Low-rank formats are not only needed to compress the data tensor as the final delivered result of high-dimensional numerical modellings, but are also preferred to approximate tensors in the numerical solution process. In [48] the proper generalised decomposition is adopted for the construction of low-rank tensors in CP and Tucker formats in a numerical homogenisation from high-resolution images. It is also possible to compute the tensors directly in low-rank formats, which can be provided by a suitable solver [73, 142, 29, 6, 91]. The rank one tensors in low-rank approximations can be seen as suboptimal global basis functions.

However, the need to compute with tensors in low-rank formats requires one to deal with operations such as addition, element-wise multiplication, or Fourier transformation. Since the low-rank tensors are described with fewer parameters, the computational complexities are typically reduced, which may lead to significant speed-up of computations. However,

performing such operations with tensors in low-rank format, it typically happens that the representation rank of the tensors grows, which calls for their truncation, i.e. their approximation or reparametrisation with fewer parameters while keeping a reasonable accuracy [114, 115, 13]. This truncation of tensors may be viewed as a generalisation of the rounding of numbers, which occurs when working with floating point formats. In general, the applications of low-rank approximations are very broad, e.g. for stochastic problems with high number of random parameters [35, 91, 111, 68], acceleration of solutions to PDEs [67, 66], or model order reduction [112], but its application to FFT-based homogenisation is new. However, an alternative low-rank representation has been studied recently in [71].

Structure of the paper. In section 6.2, two state-of-the-art Fourier-Galerkin methods are described for a model homogenisation problem of a scalar elliptic equation. In particular, the two discretisation methods based on numerical and exact integration are described along with their corresponding linear systems. Then in section 6.3 the low-rank approximation techniques are summarised and their application within a Fourier-Galerkin method is discussed. In section 6.4, the effectiveness of low-rank approximations is demonstrated on several numerical examples.

Notation. We will denote vectors and matrices by boldface letters: $\boldsymbol{a} = (a_i)_{i=1,2,\dots,d} \in \mathbb{R}^d$ or $\boldsymbol{A} = (A_{ij})_{i,j=1}^d \in \mathbb{R}^{d\times d}$. Matrix-matrix and matrix-vector multiplications are denoted as $\boldsymbol{C} = \boldsymbol{AB}$ and $\boldsymbol{c} = \boldsymbol{Ab}$, which in Einstein summation notation reads $C_{ik} = A_{ij}B_{jk}$ and $b_i = A_{ij}b_j$ respectively. The Euclidean inner product will be referred to as $\boldsymbol{a}\cdot\bar{\boldsymbol{b}} = \sum_i a_i\bar{b}_i$, and the induced norm as $\|\boldsymbol{a}\| = \sqrt{\boldsymbol{a}\cdot\bar{\boldsymbol{a}}}$. Vectors, matrices, and tensors such as $\boldsymbol{x}$, $\boldsymbol{b}$, and $\boldsymbol{A}$ arising from discretisation will be denoted by the bold sans-serif font in order to highlight their special structure. For $\boldsymbol{N} = (N_1,\dots,N_d) \in \mathbb{N}^d$, the components of a tensor $\boldsymbol{A} \in \mathbb{R}^{\boldsymbol{N}} = \bigotimes_{\alpha=1}^d \mathbb{R}^{N_\alpha}$ of order d will be denoted as $\mathsf{A}[k_1,\dots,k_d]$. The multiindex notation will be also incorporated to simplify the components of the tensors, e.g. $\mathsf{A}[k_1,\dots,k_d] = \mathsf{A}[\boldsymbol{k}]$ for a multi-index $\boldsymbol{k} = [k_1,\dots,k_d]$. The space $\mathbb{R}^{\boldsymbol{N}}$, composed of tensors of order d, can be considered as a vector space, which allows to talk about its dimension as the number of basis vectors, i.e. $\dim \mathbb{R}^{\boldsymbol{N}} = \prod_{\alpha=1}^d N_\alpha$.

The space of square integrable $\mathcal{Y}$-periodic functions defined on a periodic cell $\mathcal{Y} = (-\frac{1}{2},\frac{1}{2})^d$ is denoted as $L^2(\mathcal{Y})$. The analogous space $L^2(\mathcal{Y};\mathbb{R}^d)$ collects $\mathbb{R}^d$-valued functions $\boldsymbol{v}: \mathcal{Y} \to \mathbb{R}^d$ with components v_i from $L^2(\mathcal{Y})$. Finally, $H^1_0(\mathcal{Y}) = \{v \in L^2(\mathcal{Y}) \,|\, \nabla v \in L^2(\mathcal{Y};\mathbb{R}^d), \int_{\mathcal{Y}} v(\boldsymbol{x})\curvearrowleft = 0\}$ denotes the Sobolev space of periodic functions with zero mean.

6.2 Homogenisation by Fourier-Galerkin methods

6.2.1 Model problem

A model problem in homogenisation [9] consists of a scalar linear elliptic variational problem defined on a unit domain $\mathcal{Y} = (-\frac{1}{2},\frac{1}{2})^d$ in a spatial dimension d (we consider both $d = 2$ and $d = 3$) with material coefficients $\boldsymbol{A}: \mathcal{Y} \to \mathbb{R}^{d\times d}$, which are required to be essentially bounded, symmetric, and uniformly elliptic. This means that for almost all $\boldsymbol{x} \in \mathcal{Y}$, there are constants $0 < c_A \le C_A < +\infty$ such that

$$\boldsymbol{A}(\boldsymbol{x}) = \boldsymbol{A}^T(\boldsymbol{x}), \qquad c_A\|\boldsymbol{v}\|^2 \le \boldsymbol{A}(\boldsymbol{x})\boldsymbol{v}\cdot\boldsymbol{v} \le C_A\|\boldsymbol{v}\|^2 \quad \text{for all } \boldsymbol{v} \in \mathbb{R}^d.$$

The homogenisation problem is focused on the computation of effective material properties $\boldsymbol{A}_{\mathrm{H}} \in \mathbb{R}^{d\times d}$. Its variational formulation is based on the minimisation of a microscopic energetic functional for constant vectors $\boldsymbol{E} \in \mathbb{R}^d$, which represents an *average* of the macroscopic

gradient, as

$$\boldsymbol{A}_{\mathrm{H}}\boldsymbol{E}\cdot\boldsymbol{E} = \min_{v\in H^1_0(\mathcal{Y})} a\Big(\boldsymbol{E}+\nabla v, \boldsymbol{E}+\nabla v\Big), \tag{6.0}$$

where the bilinear form $a : L^2(\mathcal{Y};\mathbb{R}^d)\times L^2(\mathcal{Y};\mathbb{R}^d)\to\mathbb{R}$ is defined as

$$a\Big(\boldsymbol{e},\boldsymbol{w}\Big) := \int_{\mathcal{Y}} \boldsymbol{A}(\boldsymbol{x})\boldsymbol{e}(\boldsymbol{x})\cdot\boldsymbol{w}(\boldsymbol{x})\curvearrowleft.$$

The minimisation Sobolev space $H^1_0(\mathcal{Y})$ consists of zero-mean $\mathcal{Y}$-periodic microscopic fields $v : \mathbb{R}^d\to\mathbb{R}$, which have locally square integrable weak gradient and finite L^2-norm on $\mathcal{Y}$; together with (6.2.1) it satisfies the existence of a unique minimiser. Note that the minimisation problem (6.2.1) corresponds to the scalar elliptic partial differential equation $-\nabla\cdot[\boldsymbol{A}(\boldsymbol{x})\nabla u(\boldsymbol{x})] = f(\boldsymbol{x})$ with a special right-hand side $f(\boldsymbol{x}) = -\nabla\cdot\boldsymbol{A}(\boldsymbol{x})\boldsymbol{E}$ and periodic boundary conditions.

6.2.2 Fourier-Galerkin methods

Alternatively, the minimisers in (6.2.1) are described by a weak formulation: find $u\in H^1_0(\mathcal{Y})$ such that

$$a\Big(\nabla u,\nabla v\Big) = -a\Big(\boldsymbol{E},\nabla v\Big)\quad \forall\, v\in H^1_0(\mathcal{Y}).$$

This formulation is the starting point for a discretisation using Galerkin approximations, when the trial and test spaces are substituted with finite dimensional ones. We choose to discretise the function space using trigonometric polynomials, which leads to a Fourier-Galerkin method.

In order to compute the effective matrix $\boldsymbol{A}_{\mathrm{H}}$ one has to solve d minimisation problems or weak formulation for different $\boldsymbol{E}$, which are usually taken as the canonical basis of $\mathbb{R}^d$. Here we consider exclusively $\boldsymbol{E} = (\delta_{1,i})_{i=1}^d\in\mathbb{R}^d$ (i.e. in 3D $\boldsymbol{E} = [1,0,0]$); therefore, the $(1,1)$-component of the homogenised properties will be of particular interest, i.e. $\boldsymbol{A}_{\mathrm{H}}\boldsymbol{E}\cdot\boldsymbol{E} = \boldsymbol{A}_{\mathrm{H},11} =: A_{\mathrm{H}}$.

6.2.2.1 Trigonometric polynomials

The Fourier-Galerkin method, [125, 150, 147] is built on discretisations using the space of *trigonometric polynomials*

$$\mathscr{T}_{\boldsymbol{N}} = \Big\{ \sum_{\boldsymbol{k}\in\mathbb{Z}_{\boldsymbol{N}}} \widehat{v}[\boldsymbol{k}]\varphi^{\boldsymbol{k}} \;\Big|\; \widehat{v}[\boldsymbol{k}]\in\mathbb{C},\ \text{and } \widehat{v}[\boldsymbol{k}] = \overline{\widehat{v}[-\boldsymbol{k}]}\Big\},$$

where $\varphi^{\boldsymbol{k}}(\boldsymbol{x}) = \exp(2\pi\mathrm{i}\,\boldsymbol{k}\cdot\boldsymbol{x})$ are the well-known Fourier basis functions. The number of discretisation points $\boldsymbol{N} = [N,\dots,N]\in\mathbb{R}^d$ in this work take only odd values because an even N introduces Nyquist frequencies that have to be omitted to obtain a conforming approximation, see [151] for details.

There are also other natural basis vectors $\varphi^{\boldsymbol{k}}_{\boldsymbol{N}} : \mathcal{Y}\to\mathbb{R}$, the so-called fundamental trigonometric polynomials. They are expressed as a linear combination

$$\varphi^{\boldsymbol{k}}_{\boldsymbol{N}}(\boldsymbol{x}) = \frac{1}{|\boldsymbol{N}|_{\Pi}}\sum_{\boldsymbol{m}\in\mathbb{Z}_{\boldsymbol{N}}} \omega_{\boldsymbol{N}}^{-\boldsymbol{k}\boldsymbol{m}}\varphi^{\boldsymbol{m}}(\boldsymbol{x})\ \text{for}\ \boldsymbol{x}\in\mathcal{Y},$$

of Fourier basis function $\varphi^{\boldsymbol{m}}$ with complex-valued weights $\omega_{\boldsymbol{N}}^{\boldsymbol{mk}} = \exp\Big(2\pi\mathrm{i}\sum_{\alpha=1}^{d}\frac{m_\alpha k_\alpha}{N_\alpha}\Big)$ for $\boldsymbol{m},\boldsymbol{k}\in\mathbb{Z}_{\boldsymbol{N}}$. The weights are from the discrete Fourier transform (DFT) matrices in $\mathbb{C}^{\boldsymbol{N}\times\boldsymbol{N}}$ with components

$$\mathcal{F}_{\boldsymbol{N}}[\boldsymbol{m},\boldsymbol{k}] = \frac{1}{|\boldsymbol{N}|_{\Pi}}\omega_{\boldsymbol{N}}^{-\boldsymbol{mk}}, \qquad \mathcal{F}_{\boldsymbol{N}}^{-1}[\boldsymbol{m},\boldsymbol{k}] = \omega_{\boldsymbol{N}}^{\boldsymbol{mk}} \quad \text{for } \boldsymbol{m},\boldsymbol{k}\in\mathbb{Z}_{\boldsymbol{N}}.$$

The coefficients of trigonometric polynomials in the two different base are connected by the discrete Fourier transform (DFT), particularly expressed as

$$v(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}_{\boldsymbol{N}}}\hat{v}[\boldsymbol{k}]\varphi^{\boldsymbol{k}}(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}_{\boldsymbol{N}}}\mathsf{v}[\boldsymbol{k}]\varphi_{\boldsymbol{N}}^{\boldsymbol{k}}(\boldsymbol{x}) \quad \text{and} \quad \hat{\mathsf{v}} = \mathcal{F}_{\boldsymbol{N}}\mathsf{v}.$$

Due to the Dirac-delta property $\varphi_{\boldsymbol{N}}^{\boldsymbol{l}}(\boldsymbol{x}_{\boldsymbol{N}}^{\boldsymbol{k}}) = \delta_{\boldsymbol{kl}}$ of the fundamental trigonometric polynomials on a regular grid of points $\boldsymbol{x}_{\boldsymbol{N}}^{\boldsymbol{k}} = \frac{k_\alpha}{N_\alpha}$ for $\boldsymbol{k},\boldsymbol{l}\in\mathbb{Z}_{\boldsymbol{N}}$, the coefficients of the trigonometric polynomials are equal to the function values at the grid points, i.e. $\mathsf{v}[\boldsymbol{k}] = v(\boldsymbol{x}_{\boldsymbol{N}}^{\boldsymbol{k}})$.

Differential operators are naturally applied on trigonometric polynomials. In particular the gradient

$$\nabla v(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}_{\boldsymbol{N}}}\hat{v}[\boldsymbol{k}]\nabla\varphi^{\boldsymbol{k}}(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}_{\boldsymbol{N}}}2\pi\mathrm{i}\boldsymbol{k}\hat{v}[\boldsymbol{k}]\varphi^{\boldsymbol{k}}(\boldsymbol{x}),$$

corresponds to the application of the operator $\widehat{\nabla}_{\boldsymbol{N}}:\mathbb{C}^{\boldsymbol{N}}\to\mathbb{C}^{d\times\boldsymbol{N}}$ on Fourier coefficients as $(\widehat{\nabla}_{\boldsymbol{N}}\hat{\mathsf{v}})[\alpha,\boldsymbol{k}] = 2\pi\mathrm{i}k_\alpha\hat{\mathsf{v}}[\boldsymbol{k}]$. The adjoint operator $\widehat{\nabla}_{\boldsymbol{N}}^{*}:\mathbb{C}^{d\times\boldsymbol{N}}\to\mathbb{C}^{\boldsymbol{N}}$ corresponding to the divergence is then expressed as

$$(\widehat{\nabla}_{\boldsymbol{N}}^{*}\hat{\boldsymbol{w}})[\boldsymbol{k}] = \sum_{\alpha=1}^{d}-2\pi\mathrm{i}k_\alpha\hat{\boldsymbol{w}}[\alpha,\boldsymbol{k}].$$

Then the gradient operator can be expressed with respect to the basis with fundamental trigonometric polynomials as

$$\nabla v(\boldsymbol{x}) = \sum_{\boldsymbol{k}}\Big(\boldsymbol{\mathcal{F}}_{\boldsymbol{N}}^{-1}\widehat{\nabla}_{\boldsymbol{N}}\mathcal{F}_{\boldsymbol{N}}\mathsf{v}\Big)[\boldsymbol{k}]\varphi_{\boldsymbol{N}}^{\boldsymbol{k}}(\boldsymbol{x})$$

where the d-fold discrete Fourier transform (emphasises with bold) $\boldsymbol{\mathcal{F}}_{\boldsymbol{N}} = \mathbb{C}^{d\times\boldsymbol{N}}\to\mathbb{C}^{d\times\boldsymbol{N}}$ acts individually on each component of the vector field $(\boldsymbol{\mathcal{F}}_{\boldsymbol{N}}\boldsymbol{w})[\alpha] = \mathcal{F}_{\boldsymbol{N}}\boldsymbol{w}[\alpha]$ for $\alpha = 1,\cdots,d$.

The numerical treatment of the weak formulation (6.2.2) or a corresponding Galerkin approximation requires the use of numerical integration. In this manuscript we incorporate two versions: an exact integration [147] as described in sub-section 6.2.2.3, and a numerical integration as described in sub-section 6.2.2.2.

6.2.2.2 The Fourier-Galerkin method with numerical integration (GaNi)

This numerical integration based on the rectangle (or the mid-point) rule corresponds to the original Moulinec-Suquet algorithm [102, 103], as the resulting discrete solution vectors fully coincide. This approach, applied to the bilinear form (6.2.1) on regular grids, reads

$$a\Big(\boldsymbol{e},\boldsymbol{w}\Big) \approx a_{\boldsymbol{N}}\Big(\boldsymbol{e}_{\boldsymbol{N}},\boldsymbol{w}_{\boldsymbol{N}}\Big) = \sum_{\boldsymbol{k}\in\mathbb{Z}_{\boldsymbol{N}}}\boldsymbol{A}(\boldsymbol{x}_{\boldsymbol{N}}^{\boldsymbol{k}})\boldsymbol{e}_{\boldsymbol{N}}(\boldsymbol{x}_{\boldsymbol{N}}^{\boldsymbol{k}})\cdot\boldsymbol{w}_{\boldsymbol{N}}(\boldsymbol{x}_{\boldsymbol{N}}^{\boldsymbol{k}}) = \Big(\widetilde{\boldsymbol{A}}\boldsymbol{e},\boldsymbol{w}\Big)_{\mathbb{R}^{d\times\boldsymbol{N}}},$$

where $\boldsymbol{e}$ and $\boldsymbol{w}$ store the function values on the grid (e.g. $e[\alpha, \boldsymbol{k}] = e_{\boldsymbol{N},\alpha}(\boldsymbol{x}^{\boldsymbol{k}}_{\boldsymbol{N}})$), and $\widetilde{\boldsymbol{A}} \in \mathbb{R}^{d\times d\times \boldsymbol{N}\times \boldsymbol{N}}$ is a block diagonal tensor with components

$$\widetilde{\boldsymbol{A}}[\alpha, \beta, \boldsymbol{k}, \boldsymbol{l}] = \delta_{\boldsymbol{kl}} A_{\alpha\beta}(\boldsymbol{x}^{\boldsymbol{k}}_{\boldsymbol{N}});$$

but one only needs to store the diagonals, which can be done in a tensor of shape $d \times d \times \boldsymbol{N}$.

The numerical integration leads to an approximate formulation of the Galerkin approximation of (6.2.2):

$$\text{find } u \in \mathscr{T}_{\boldsymbol{N}}: \quad a_{\boldsymbol{N}}\Big(\nabla u_{\boldsymbol{N}}, \nabla v_{\boldsymbol{N}}\Big) = -a_{\boldsymbol{N}}\Big(\boldsymbol{E}, \nabla v_{\boldsymbol{N}}\Big), \quad \forall v_{\boldsymbol{N}} \in \mathscr{T}_{\boldsymbol{N}};$$

note that the approximation is exact for constant material coefficients $\boldsymbol{A}$. This formulation, that can be seen also as a collocation method [158], is equivalent to the original Moulinec and Suquet formulation [102] in the sense that the solution vectors coincide [150]. However, the formulation here builds on the variational formulation [150] solved for the potential field (instead of gradient one).

The combination of numerical integration and differentiation of trigonometric polynomials (6.2.2.1) allows to approximate the bilinear form in terms of the nodal values of potential fields

$$a_{\boldsymbol{N}}\Big(\nabla u_{\boldsymbol{N}}, \nabla v_{\boldsymbol{N}}\Big) = \Big(\widetilde{\boldsymbol{A}}\boldsymbol{\mathcal{F}}^{-1}_{\boldsymbol{N}}\widehat{\nabla}_{\boldsymbol{N}}\mathcal{F}_{\boldsymbol{N}}\boldsymbol{u}, \boldsymbol{\mathcal{F}}^{-1}_{\boldsymbol{N}}\widehat{\nabla}_{\boldsymbol{N}}\mathcal{F}_{\boldsymbol{N}}\boldsymbol{v}\Big)_{\mathbb{R}^{d\times \boldsymbol{N}}}.$$

In order to deduce the linear system, all operators acting on test vectors $\boldsymbol{v}_{\boldsymbol{N}}$ are moved to the trial vector $\boldsymbol{u}_{\boldsymbol{N}}$ as adjoint operators to reveal the linear system in the original space

$$\mathcal{F}^{-1}_{\boldsymbol{N}}\widehat{\nabla}^*_{\boldsymbol{N}}\boldsymbol{\mathcal{F}}_{\boldsymbol{N}}\widetilde{\boldsymbol{A}}\boldsymbol{\mathcal{F}}^{-1}_{\boldsymbol{N}}\widehat{\nabla}_{\boldsymbol{N}}\mathcal{F}_{\boldsymbol{N}}\boldsymbol{u} = -\mathcal{F}^{-1}_{\boldsymbol{N}}\widehat{\nabla}^*_{\boldsymbol{N}}\boldsymbol{\mathcal{F}}_{\boldsymbol{N}}\widetilde{\boldsymbol{A}}\boldsymbol{E},$$

where $\boldsymbol{E} \in \mathbb{R}^{d\times \boldsymbol{N}}$ is constant with components $\boldsymbol{E}[\alpha, \boldsymbol{k}] = E_\alpha$. One may notice that the system can be solved in Fourier space to save one computation of FFT and its inverse, which leads to the linear system in Fourier space

$$\widehat{\nabla}^*_{\boldsymbol{N}}\boldsymbol{\mathcal{F}}_{\boldsymbol{N}}\widetilde{\boldsymbol{A}}\boldsymbol{\mathcal{F}}^{-1}_{\boldsymbol{N}}\widehat{\nabla}_{\boldsymbol{N}}\widehat{\boldsymbol{u}} = -\widehat{\nabla}^*_{\boldsymbol{N}}\boldsymbol{\mathcal{F}}_{\boldsymbol{N}}\widetilde{\boldsymbol{A}}\boldsymbol{E}.$$

6.2.2.3 Fourier-Galerkin method with exact integration (GA)

For many types of material coefficients (6.2.1) and basis functions, there is a possibility to integrate the bilinear forms in the weak formulation exactly, which leads to a Galerkin approximation with exact integration

$$\text{find } u \in \mathscr{T}_{\boldsymbol{N}}: \quad a\Big(\nabla u_{\boldsymbol{N}}, \nabla v_{\boldsymbol{N}}\Big) = a\Big(\boldsymbol{E}, \nabla v_{\boldsymbol{N}}\Big) \quad \forall v_{\boldsymbol{N}} \in \mathscr{T}_{\boldsymbol{N}}.$$

However, the exact integration of the Fourier-Galerkin formulation, in contrast to FEM, leads to a full linear system, which can be overcome with a double-grid integration with projection (DoGIP) [151, 147]. The DoGIP is a general method applicable also within the finite element method [148]. The original evaluation of the material law on a grid of size $\boldsymbol{N}$ is reformulated as an evaluation on a double grid $2\boldsymbol{N} - \boldsymbol{1}$ with modified material coefficients; they can be expressed as a modification of the original material coefficients.

The main idea relies on expressing gradients of the trial and a test function together

$$\nabla u_{\boldsymbol{N}}(\boldsymbol{x}) \otimes \nabla v_{\boldsymbol{N}}(\boldsymbol{x}) = \boldsymbol{e}_{\boldsymbol{N}}(\boldsymbol{x}) \otimes \boldsymbol{w}_{\boldsymbol{N}}(\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\mathbb{Z}_{2\boldsymbol{N}-\boldsymbol{1}}} \boldsymbol{e}[:, \boldsymbol{k}] \otimes \boldsymbol{w}[:, \boldsymbol{k}]\varphi^{\boldsymbol{k}}_{2\boldsymbol{N}-\boldsymbol{1}}(\boldsymbol{x})$$

with respect to the basis of the double grid space consisting of trigonometric polynomials with doubled frequencies $\mathcal{T}_{2\boldsymbol{N}-\boldsymbol{1}}$; the arrays $\boldsymbol{e}$ and $\boldsymbol{w}$ store the values of the trigonometric polynomials on the double grid, e.g. $e[\alpha, \boldsymbol{k}] = \boldsymbol{e}_{\boldsymbol{N},\alpha}(\boldsymbol{x}^{\boldsymbol{k}}_{2\boldsymbol{N}-\boldsymbol{1}})$ for $\alpha \in \{1, \dots, d\}$ and $\boldsymbol{k} \in \mathbb{Z}_{2\boldsymbol{N}-\boldsymbol{1}}$. Then the bilinear form can be expressed on the double grid

$$a\Big(\boldsymbol{e}_{\boldsymbol{N}}, \boldsymbol{w}_{\boldsymbol{N}}\Big) = \sum_{\boldsymbol{k}\in\mathbb{Z}_{2\boldsymbol{N}-\boldsymbol{1}}} \int_{\mathcal{Y}} \boldsymbol{A}(\boldsymbol{x})\varphi^{\boldsymbol{k}}_{2\boldsymbol{N}-\boldsymbol{1}}(\boldsymbol{x})\curvearrowleft : \boldsymbol{e}_{2\boldsymbol{N}-\boldsymbol{1}}[:, \boldsymbol{k}] \otimes \boldsymbol{w}_{2\boldsymbol{N}-\boldsymbol{1}}[:, \boldsymbol{k}] = \Big(\boldsymbol{A}\boldsymbol{e}, \boldsymbol{w}\Big)_{\mathbb{R}^{d\times(2\boldsymbol{N}-\boldsymbol{1})}}$$

where : is a double contraction between two matrices of size $d \times d$ and the material coefficients are defined as

$$A[\alpha, \beta, \boldsymbol{k}, \boldsymbol{l}] = \delta_{\boldsymbol{k}\boldsymbol{l}} \int_{\mathcal{Y}} A_{\alpha\beta}(\boldsymbol{x})\varphi^{\boldsymbol{k}}_{2\boldsymbol{N}-\boldsymbol{1}}(\boldsymbol{x})\curvearrowleft \quad \text{for } \alpha, \beta \in \{1, \dots, d\} \text{ and } \boldsymbol{k}, \boldsymbol{l} \in \mathbb{Z}_{2\boldsymbol{N}-\boldsymbol{1}}.$$

This integration can be performed exactly for a large class of material coefficients. In particular in [147, 151], square or circular inclusions have been considered, as well as image-based composites, materials with coefficients constant or bilinear over pixels (voxels in 3D). Moreover, the evaluation of modified material coefficients can be performed effectively by FFT.

In order to derive the linear system, we have to still describe the interpolation from the original to the double grid space. As the spaces of trigonometric polynomials are nested $\mathcal{T}_{\boldsymbol{N}} \subset \mathcal{T}_{\boldsymbol{M}}$ for $\boldsymbol{N} < \boldsymbol{M}$ (element-wise), we can just inject the polynomial to the bigger space by adding trigonometric polynomials with zero Fourier coefficients. This can be represented by the zero-padding injection operator $\mathcal{I} : \mathbb{C}^{d\times\boldsymbol{N}} \to \mathbb{C}^{d\times(2\boldsymbol{N}-\boldsymbol{1})}$, defined as

$$(\mathcal{I}\widehat{\boldsymbol{w}})[:, \boldsymbol{k}] = \begin{cases} \widehat{\boldsymbol{w}}[:, \boldsymbol{k}], & \text{for } \boldsymbol{k} \in \mathbb{Z}_{\boldsymbol{N}} \\ \boldsymbol{0} & \text{for } \boldsymbol{k} \in \mathbb{Z}_{2\boldsymbol{N}-\boldsymbol{1}} \setminus \mathbb{Z}_{\boldsymbol{N}} \end{cases}.$$

Its adjoint operator $\mathcal{I}^* : \mathbb{C}^{d\times(2\boldsymbol{N}-\boldsymbol{1})} \to \mathbb{C}^{d\times\boldsymbol{N}}$ just removes the frequencies $\boldsymbol{k} \in \mathbb{Z}_{2\boldsymbol{N}-\boldsymbol{1}} \setminus \mathbb{Z}_{\boldsymbol{N}}$, i.e. projects on the $\boldsymbol{k} \in \mathbb{Z}_{\boldsymbol{N}}$.

This allows us to deduce the linear system with exact integration

$$\widehat{\nabla}^*_{\boldsymbol{N}}\mathcal{I}^*\boldsymbol{\mathcal{F}}_{2\boldsymbol{N}-\boldsymbol{1}}\boldsymbol{A}\boldsymbol{\mathcal{F}}^{-1}_{2\boldsymbol{N}-\boldsymbol{1}}\mathcal{I}\widehat{\nabla}_{\boldsymbol{N}}\widehat{\boldsymbol{u}} = -\widehat{\nabla}^*_{\boldsymbol{N}}\mathcal{I}^*\boldsymbol{\mathcal{F}}_{2\boldsymbol{N}-\boldsymbol{1}}\boldsymbol{A}\boldsymbol{E},$$

which has very similar structure compared to the scheme based on numerical integration (6.2.2.2).

6.2.3 Preconditioning

Following the recent paper [99], the preconditioning of both linear systems (6.2.2.2) and (6.2.2.3) is based on a Laplacian expressed in the Fourier domain as

$$\widehat{\boldsymbol{P}}[\boldsymbol{k}, \boldsymbol{l}] = \delta_{\boldsymbol{k}\boldsymbol{l}}\boldsymbol{k} \cdot \boldsymbol{l} \quad \text{for } \boldsymbol{k}, \boldsymbol{l} \in \mathbb{Z}_{\boldsymbol{N}},$$

which is a simple diagonal preconditioner. Its inverse is given by the Moore-Penrose pseudoinverse $\widehat{\boldsymbol{P}}^{-1}[\boldsymbol{k}, \boldsymbol{l}] = \delta_{\boldsymbol{k}\boldsymbol{l}}\frac{1}{\boldsymbol{k}\cdot\boldsymbol{k}}$ for $\boldsymbol{k} \in \mathbb{Z}_{\boldsymbol{N}} \setminus \{\boldsymbol{0}\}$ and $\widehat{\boldsymbol{P}}^{-1}[\boldsymbol{0}, \boldsymbol{0}] = 0$; the latter condition enforces the zero-mean property of the approximated vectors. The preconditioned systems are explicitly stated for both discretisation schemes

$$\widehat{\boldsymbol{P}}^{-1}\widehat{\nabla}^*_{\boldsymbol{N}}\boldsymbol{\mathcal{F}}_{\boldsymbol{N}}\widetilde{\boldsymbol{A}}\boldsymbol{\mathcal{F}}^{-1}_{\boldsymbol{N}}\widehat{\nabla}_{\boldsymbol{N}}\widehat{\boldsymbol{u}} = -\widehat{\boldsymbol{P}}^{-1}\widehat{\nabla}^*_{\boldsymbol{N}}\boldsymbol{\mathcal{F}}_{\boldsymbol{N}}\widetilde{\boldsymbol{A}}\boldsymbol{E},$$

for the preconditioning of (6.2.2.2), and

$$\widehat{\boldsymbol{P}}^{-1}\widehat{\nabla}^*_{\boldsymbol{N}}\mathcal{I}^*\boldsymbol{\mathcal{F}}_{2\boldsymbol{N}-\boldsymbol{1}}\boldsymbol{A}\boldsymbol{\mathcal{F}}^{-1}_{2\boldsymbol{N}-\boldsymbol{1}}\mathcal{I}\widehat{\nabla}_{\boldsymbol{N}}\widehat{\boldsymbol{u}} = -\widehat{\boldsymbol{P}}^{-1}\widehat{\nabla}^*_{\boldsymbol{N}}\mathcal{I}^*\boldsymbol{\mathcal{F}}_{2\boldsymbol{N}-\boldsymbol{1}}\boldsymbol{A}\boldsymbol{E}.$$

for the preconditioning of (6.2.2.3).

6.3 FFT-based methods with low-rank approximations

Applying low-rank approximation techniques is of particular interest for problems with a huge number of degrees of freedom. The low-rank approximations can not only furnish a posterior data compression of the solution array, but also reduce computational complexity by exploiting low-rank format representations in the solution process. For the latter one needs some operations such as additions, element-wise multiplication, and the fast Fourier transform (FFT) to be implemented on tensors in low-rank format. In this section we introduce an FFT-based solution process incorporating low-rank representations of tensors. In the following sub-section 6.3.1, the low-rank approximation formats are summarised along the corresponding operations; details can be found in textbooks or in appendix 6.A. Then the application of low-rank approximation for the Fourier-Galerkin method is described and discussed in sub-section 6.3.2, and the suitable linear solvers in sub-section 6.3.3.

6.3.1 Overview of low-rank formats

Here we give a brief introduction of three types of low-rank tensors that are applied in this work, they are of canonical polyadic (CP), Tucker, and tensor train format respectively. The CP format is only used for tensors of order two because of its intrinsic difficulty in finding optimal approximation for tensors with higher order. The necessity and impact of rank truncation is also emphasized. Interested readers are provided by more details about the operations on the low-rank tensors in the Appendix 6.A.

6.3.1.1 Canonical polyadic format

A CP r-term approximation of a tensor $\boldsymbol{v} \in \mathbb{K}^{N_1 \times \cdots \times N_d}$ (the field $\mathbb{K}$ is $\mathbb{R}$ or $\mathbb{C}$) is a sum of r rank-1 tensors. In this work the CP format is only used for tensors of order two ($d = 2$), i.e. matrices. In this case the representation has the form:

$$\boldsymbol{v} \approx \widetilde{\boldsymbol{v}} = \sum_{i=1}^{r} c[i] \boldsymbol{b}^{(1)}[i] \otimes \boldsymbol{b}^{(2)}[i],$$

where $\boldsymbol{c} \in \mathbb{R}^r$ stores the coefficients with respect to vectors $\boldsymbol{b}^{(j)} \in \mathbb{K}^{r \times N_j}$ in the directions of indices j. A low-rank representation for order-2 tensors (matrices) can be obtained by various matrix factorizing methods, among which the Singular Value Decomposition (SVD) is prominent as it provides a factorization that minimises the Frobenius-norm error of an r-term approximation. The level of compression (reduction of memory requirements) depends on the rank r. In order to find a solution in such a low-rank form, it requires to perform several operations occurring in the Fourier Galerkin method, particularly the FFT and element-wise multiplication.

The linearity and the tensor-product structure of the Fourier transform facilitates to express d-dimensional FFT of a tensor (of order d) as the sum of tensor products of 1-dimensional FFTs, i.e.,

$$\mathcal{F}_{\boldsymbol{N}}(\widetilde{\boldsymbol{v}}) = \sum_{i=1}^{r} c[i] \mathcal{F}_{N_1}(\boldsymbol{b}^{(1)}[i]) \otimes \mathcal{F}_{N_2}(\boldsymbol{b}^{(2)}[i]).$$

For the same number of tensor components in all directions j, i.e. $N_j = N$, this d-dimensional FFT algorithm has a complexity $O(drN \log N)$, which is much better than $O(dN^d \log N)$ for

the full tensor, when the rank r is kept low. Note that this operation does not change the rank of a transformed tensor.

Another operation that occurs in the Fourier-Galerkin method is the sum and the element-wise (Hadamard) product of two tensors in low-rank format. In the case of the CP format it is computed as:

$$\widetilde{\boldsymbol{v}} + \widetilde{\boldsymbol{w}} = \sum_{i=1}^{r} c_{\boldsymbol{v}}[i] \left(\boldsymbol{b}_{\boldsymbol{v}}^{(1)}[i] \otimes \boldsymbol{b}_{\boldsymbol{v}}^{(2)}[i] \right) + \sum_{k=1}^{s} c_{\boldsymbol{w}}[k] \left(\boldsymbol{b}_{\boldsymbol{w}}^{(1)}[k] \otimes \boldsymbol{b}_{\boldsymbol{w}}^{(2)}[k] \right),$$
$$\widetilde{\boldsymbol{v}} \odot \widetilde{\boldsymbol{w}} = \sum_{i=1}^{r} \sum_{k=1}^{s} c_{\boldsymbol{v}}[i] c_{\boldsymbol{w}}[k] \left(\boldsymbol{b}_{\boldsymbol{v}}^{(1)}[i] \odot \boldsymbol{b}_{\boldsymbol{w}}^{(1)}[k] \right) \otimes \left(\boldsymbol{b}_{\boldsymbol{v}}^{(2)}[i] \odot \boldsymbol{b}_{\boldsymbol{w}}^{(2)}[k] \right).$$

While addition of two tensor costs no floating point operations and only requires more memory, the element-wise multiplication has a complexity of $O(rsdN)$, which is significantly less than the N^d operations for full tensors, especially when the ranks r and s are much smaller than N.

6.3.1.2 Tucker format

The decomposition of higher order tensors has many variants. The Tucker format representation is linked to the definition of a tensor subspace $\mathcal{V} = \bigotimes_{j=1}^{d} \mathcal{V}^j$ where $\mathcal{V}^j$ is a subspace of $\mathbb{R}^{N_j}$ generated by the span of vectors $\{\boldsymbol{b}^{(j)}[i] \,|\, i = 1, \ldots, r_j\}$; these vectors, which may be a frame, are typically chosen as an orthogonal or orthonormal basis. The Tucker format is then a linear combination of tensor products of all possible combinations of basis vectors in different directions, i.e.

$$\boldsymbol{v} \approx \sum_{i_1=1}^{r_1} \cdots \sum_{i_d=1}^{r_d} c[i_1, \ldots, i_d] \bigotimes_{j=1}^{d} \boldsymbol{b}^{(j)}[i_j] \in \mathbb{R}^{\boldsymbol{N}},$$

where the core $\boldsymbol{c} \in \bigotimes_{\alpha=1}^{d} \mathbb{R}^{r_\alpha}$ is a tensor of order d. The CP format is then a special form of the Tucker format with a diagonal core. Note that naturally there can be different number of basis vectors in different directions.

6.3.1.3 The Tensor train (TT) format

The tensor train is another format which is suitable for the decomposition of higher order tensor. The idea is based on recursive decompositions done sequentially along the *tensor's* individual spatial dimensions. For tensors of order 3, the decomposition of the tensor of size $N \times N \times N$ is computed in two steps. Using the standard SVD algorithm, the decomposition is first computed on the reshaped matrix of size $N \times N^2$. It is followed by the decomposition of the reshaped right-singular vectors, i.e. of the matrix of size $N \times N$. The above recursive decomposition thus leads to

$$\boldsymbol{v} = \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \boldsymbol{b}^{(1)}[1, :, i_1] \otimes \boldsymbol{b}^{(2)}[i_1, :, i_2] \otimes \boldsymbol{b}^{(3)}[i_2, :, 1],$$

where the vectors $\boldsymbol{b}^{(j)}[i_{j-1}, :, i_j] \in \mathbb{R}^{N_j}$ are vectors in direction j. The tensor's components can be explicitly written for $\boldsymbol{k} = (k_1, k_2, k_3)$ as

$$\boldsymbol{v}[\boldsymbol{k}] = \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} b^{(1)}[1, k_1, i_1] b^{(2)}[i_1, k_2, i_2] b^{(3)}[i_2, k_3, 1].$$

For $d = 2$ it is identical to the CP format.

The tensor train format in (6.3.1.3) is again expressed as a linear combination of rank one tensors, on which a d-dimensional FFT can be applied through a series of one-dimensional FFTs on the train *carriages* along the second index, i.e. applied on the vectors $\boldsymbol{b}^{(j)}[i_{j-1}, :, i_j] \in \mathbb{K}^{N_j}$ for all i_j. The operations addition or element-wise multiplication are discussed in the Appendix 6.A.3.

6.3.1.4 Rank truncation

Rank truncation is the way to reduce computational complexity by a reasonable compromise in the precision of the low-rank approximations. It is particularly necessitated by the fact that operations on low-rank tensors like addition and element-wise multiplication usually inflate the representation rank $\boldsymbol{r}$, potentially at a very fast rate, which is detrimental to a fast computation. On the other hand, in the resulted representation, a large part of the $\boldsymbol{r}$ terms are not essential and can be given up without or with minor loss of accuracy, if done correctly.

Rank truncations of tensors in the three low-rank formats are all based on QR decomposition, SVD, or high order SVD (HOSVD) [53], which provide optimal or suboptimal truncations and error estimates.

Other truncations are also possible. Particularly, the element-wise multiplication of two tensors with rank r results in a tensor of rank $s = r^2$, which is truncated with computational complexity $O(Ns^2)$ for CP and Tucker and $O(Ns^3)$ for TT format. In case of higher rank r of the original tensors, the truncation become computational bottleneck. To speed up the basis orthogonalization procedure, the basis with relatively small norms can also be removed before the orthogonalization to trade accuracy for efficiency. This is usually beneficial in an iterative solver.

We supplement a more detailed introduction to the truncation procedure in each low-rank format in the Appendix 6.A.

6.3.2 Applications of low-rank approximations on the linear systems

Here, we discuss the application of low-rank formats on the linear systems (6.1), which are again stated here for the reader's convenience

$$\begin{aligned} \overbrace{\widehat{\boldsymbol{P}}^{-1}\widehat{\nabla}_{\boldsymbol{N}}^{*}\boldsymbol{\mathcal{F}}_{\boldsymbol{N}}\widetilde{\boldsymbol{A}}\boldsymbol{\mathcal{F}}_{\boldsymbol{N}}^{-1}\widehat{\nabla}_{\boldsymbol{N}}}^{\widetilde{\boldsymbol{C}}}\,\widehat{\boldsymbol{u}} &= \overbrace{-\widehat{\boldsymbol{P}}^{-1}\widehat{\nabla}_{\boldsymbol{N}}^{*}\boldsymbol{\mathcal{F}}_{\boldsymbol{N}}\widetilde{\boldsymbol{A}}\boldsymbol{E}}^{\widetilde{\boldsymbol{b}}}, \\ \underbrace{\widehat{\boldsymbol{P}}^{-1}\widehat{\nabla}_{\boldsymbol{N}}^{*}\mathcal{I}^{*}\boldsymbol{\mathcal{F}}_{2\boldsymbol{N}-\boldsymbol{1}}\boldsymbol{A}\boldsymbol{\mathcal{F}}_{2\boldsymbol{N}-\boldsymbol{1}}^{-1}\mathcal{I}\widehat{\nabla}_{\boldsymbol{N}}}_{\boldsymbol{C}}\,\widehat{\boldsymbol{u}} &= \underbrace{-\widehat{\boldsymbol{P}}^{-1}\widehat{\nabla}_{\boldsymbol{N}}^{*}\mathcal{I}^{*}\boldsymbol{\mathcal{F}}_{2\boldsymbol{N}-\boldsymbol{1}}\boldsymbol{A}\boldsymbol{E}}_{\boldsymbol{b}}. \end{aligned}$$

The solution vector $\boldsymbol{u}$ (or their Fourier coefficients $\widehat{\boldsymbol{u}} = \boldsymbol{F}\boldsymbol{u}$) stores the values of the trigonometric polynomial on the d-dimensional regular discretisation grid. Therefore the solution vector can be naturally represented as a tensor of order d, which allows a low-rank representation. In order to avoid the computation of the full tensor and its decomposition, the low-rank tensor $\widehat{\boldsymbol{u}}$ is computed by a suitable iterative solver introduced in 6.3.3. It requires to perform matrix vector multiplication for a low-rank tensor $\boldsymbol{v}$, which is approximated as

$$\begin{aligned} \widetilde{\boldsymbol{C}}\boldsymbol{v} &\approx \mathcal{T}\widehat{\boldsymbol{P}}^{-1}\mathcal{T}\widehat{\nabla}_{\boldsymbol{N}}^{*}\boldsymbol{\mathcal{F}}_{\boldsymbol{N}}\mathcal{T}\widetilde{\boldsymbol{A}}\boldsymbol{\mathcal{F}}_{\boldsymbol{N}}^{-1}\widehat{\nabla}_{\boldsymbol{N}}\boldsymbol{v}, \\ \boldsymbol{C}\boldsymbol{v} &\approx \mathcal{T}\widehat{\boldsymbol{P}}^{-1}\mathcal{T}\widehat{\nabla}_{\boldsymbol{N}}^{*}\mathcal{I}^{*}\boldsymbol{\mathcal{F}}_{2\boldsymbol{N}-\boldsymbol{1}}\mathcal{T}\boldsymbol{A}\boldsymbol{\mathcal{F}}_{2\boldsymbol{N}-\boldsymbol{1}}^{-1}\mathcal{I}\widehat{\nabla}_{\boldsymbol{N}}\boldsymbol{v}; \end{aligned}$$

similarly, the right-hand side of the linear systems is approximated by a low-rank tensors

$$\widetilde{\boldsymbol{b}} = -\mathcal{T}\widehat{\boldsymbol{P}}^{-1}\widehat{\nabla}_{\boldsymbol{N}}^{*}\boldsymbol{\mathcal{F}}_{\boldsymbol{N}}\widetilde{\boldsymbol{A}}\boldsymbol{E},$$
$$\boldsymbol{b} = -\mathcal{T}\widehat{\boldsymbol{P}}^{-1}\widehat{\nabla}_{\boldsymbol{N}}^{*}\mathcal{I}^{*}\boldsymbol{\mathcal{F}}_{2\boldsymbol{N}-\boldsymbol{1}}\boldsymbol{A}\boldsymbol{E}.$$

These approximations involve several operations in low-rank formats such as differentiation, divergence, Fourier transform, and the truncation operator $\mathcal{T}$, which keeps the rank $\boldsymbol{r}$ at an affordable level. The operations are tabulated in the Table 6.3.1 together with the corresponding implementations in low-rank format and their impact on the rank $\boldsymbol{r}$.

Operation	low-rank tensor implementation	Rank $\boldsymbol{r}$
Differentiation (gradient)	element-wise multiplication	remains unchanged
Divergence	element-wise multiplication and addition	is increased
Evaluation of material law	element-wise multiplication	is increased
d-dimensional FFT	series of 1D FFTs	remains unchanged
Preconditioning	element-wise multiplication	is increased

Table 6.3.1: Operations and their implementations in low-rank formats

Since the material coefficients $\widetilde{\boldsymbol{A}}, \boldsymbol{A}$ and the preconditioner $\boldsymbol{P}^{-1}$ are diagonal or block-diagonal for non-isotropic material coefficients, the related matrix-vector multiplications are implemented as element-wise multiplications, which inevitably inflates the representation rank of the tensors in low-rank format. We apply a rank truncation after each multiplication to keep the computational complexity at a relatively low level, while maintaining reasonable accuracy in the solution.

The application of the gradient and divergence in Fourier space is also implemented as element-wise multiplications. The differentiation operator for trigonometric polynomials is by nature a rank-1 tensor in the form

$$\widehat{\nabla}_{\boldsymbol{N}} = [2\pi\mathrm{i}\boldsymbol{K}_1 \otimes \boldsymbol{1} \otimes \boldsymbol{1}, \boldsymbol{1} \otimes 2\pi\mathrm{i}\boldsymbol{K}_2 \otimes \boldsymbol{1}, \boldsymbol{1} \otimes \boldsymbol{1} \otimes 2\pi\mathrm{i}\boldsymbol{K}_3]$$

in the 3D setting, where $\boldsymbol{K}_\alpha = (k \in \mathbb{Z}\,;\, |k| < N/2)$ is a vector of all discrete frequencies in direction α. So the corresponding element-wise multiplication keeps the rank of tensors unchanged. However, for the divergence the contraction along the first component of $\widehat{\nabla}_{\boldsymbol{N}}$ is provided by the operation addition of two low-rank formats, which increases the rank, and hence a truncation has to be performed.

The last operation that occurs in the system is the d-dimensional fast Fourier transform (FFT) which is efficiently evaluated using 1-dimensional FFTs. Moreover the rank of the tensor remains the same in this operation.

6.3.3 Linear solvers

For the full solver we have used preconditioned conjugate gradients, which is considered to be the best available solver for FFT-based homogenisation [98, 99]. However, the linear systems with low-rank approximations require solvers that are insensitive to small perturbations, as the matrix-vector product is computed only approximately, due to the truncation of tensors. Therefore conjugate gradient method that builds on the orthogonalisation of Krylov subspace vectors using a short-term recurrence relation is inappropriate.

The systems with low-rank approximations are solved here with minimal residual iteration [123] which is closely related to Richardson iteration. The latter is well established in the FFT-based community, as it corresponds to the original Moulinec-Suquet algorithm. Both methods solve the linear system $\boldsymbol{C}\boldsymbol{u} = \boldsymbol{d}$, see (6.2) and (6.3) for details, by the iteration

$$\boldsymbol{u}_{(i+1)} = \boldsymbol{u}_{(i)} + \omega \underbrace{(\boldsymbol{d} - \boldsymbol{C}\boldsymbol{u}_{(i)})}_{\boldsymbol{r}_{(i)}} = \Big(\boldsymbol{I} - \omega\boldsymbol{C}\Big)\boldsymbol{u}_{(i)} + \omega\boldsymbol{d}.$$

In the Richardson iteration, the parameter ω is chosen such that the iteration matrix $(\boldsymbol{I} - \omega\boldsymbol{C})$ has a norm smaller than one to guarantee convergence. A fixed value ω is set on the basis of a priori knowledge about the extreme eigenvalues of the system matrix $\boldsymbol{C}$, i.e.

$$\omega = \frac{2}{\lambda_{\min}(\boldsymbol{C}) + \lambda_{\max}(\boldsymbol{C})},$$

because it satisfies the minimal norm of the iterative matrix as proposed in [103] for FFT-based homogenisation. Here $\lambda_{\min}(\boldsymbol{C})$ denotes the smallest positive eigenvalue, as the system matrix is only positive semidefinite. In particular, the linear systems in (6.1) contains one zero eigenvalue corresponding to the constant fields, while the linear systems that are formulated in traditional FFT-based homogenisation for gradients fields contain many zero eigenvalues corresponding to the eigenspace composed of divergence-free fields. In both cases the solver produces the solution in the space of compatible fields.

In the minimal residual iteration, the parameter ω is chosen at each iteration as the minimizer of the next residual $\boldsymbol{r}_{(i+1)}$ over all increments of $\boldsymbol{u}$ in the direction of $\boldsymbol{r}_{(i)}$., i.e.

$$\boldsymbol{u}_{(i+1)} = \boldsymbol{u}_{(i)} + \omega_{(i)}\boldsymbol{r}_{(i)}, \quad \text{with } \omega_{(i)} = \frac{\Big(\boldsymbol{C}\,\boldsymbol{r}_{(i)}, \boldsymbol{r}_{(i)}\Big)}{\|\boldsymbol{C}\,\boldsymbol{r}_{(i)}\|^2}.$$

We adopt the latter method in this work, because of our observation that the minimal residual iteration is more robust than Richardson iteration, for which we have observed a divergence when a massive truncation has been used during the iterations. For a low-rank approximation of a solution vector, note that the solver has to deal with the matrix vector product $\boldsymbol{C}\boldsymbol{u}_{(i)}$, which is computed only approximately (6.3) to limit the growth of the solution rank. The rank also grows by the operation addition during the iteration. Therefore, a truncation is included at each step of the low-rank variant of the minimal residual iteration, i.e.

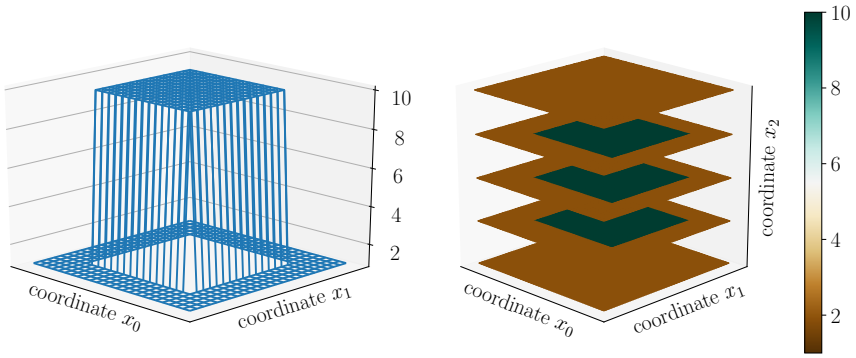
$$\boldsymbol{u}_{(i+1)} = \mathcal{T}[\boldsymbol{u}_{(i)} + \omega_{(i)}(\boldsymbol{d} - \boldsymbol{C}\boldsymbol{u}_{(i)})].$$

6.4 Numerical results

The methodology described in the previous sections is tested on several numerical examples with material parameters defined in section 6.4.1. We compare two numerical homogenisation schemes: the Fourier-Galerkin method with numerical integration (GaNi) and a version with exact integration (Ga), described in sections 6.2.2.3 and 6.2.2.2. The preconditioned linear systems stated in (6.1) are solved by conjugate gradient method. The same systems that are equipped with low-rank tensor approximations are solved by the minimal residual iteration, which is discussed in section 6.3.3.

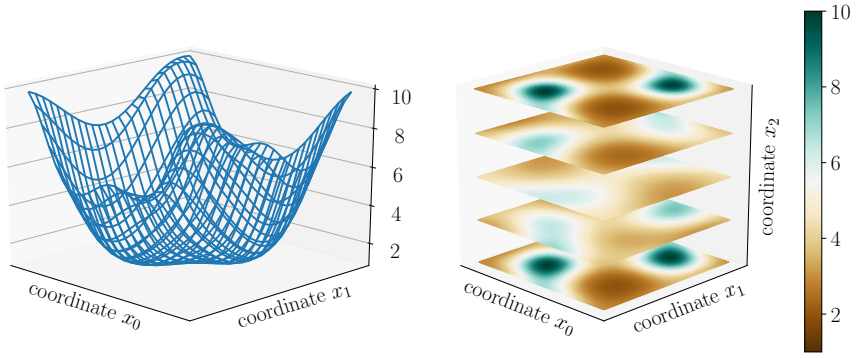
The numerical results were calculated using software FFTHomPy (FFT-based Homogenisation in Python), which is freely available at https://github.com/vondrejc/FFTHomPy; the software contains examples, which are described in the following sections.

6.4.1 Material parameters



(a) : Square inclusion (□) in 2D **(b) :** Square (cube) inclusion (□) in 3D

Figure 6.4.1: Material coefficients (6.4.1) of the square and the cube inclusion defined by (6.4.1).



(a) : Stochastic material (S) in 2D **(b) :** Stochastic material (S) in 3D

Figure 6.4.2: One sample of the stochastic material defined by (6.4.1).

Here, we present two material examples on which we did numerical tests. The first is defined as

$$\boldsymbol{A}_{\square}(\boldsymbol{x}) = \boldsymbol{I}(1 + \rho\chi(\boldsymbol{x}))$$

where $\boldsymbol{I} \in \mathbb{R}^{d\times d}$ is the identity matrix and the parameter $\rho = 10$ corresponds to a material contrast. The function $\chi : \mathcal{Y} \to \mathbb{R}^d$ describing the topology of the inclusions is defined on a unit cell $\mathcal{Y} = (-\frac{1}{2}, \frac{1}{2})^d$ as

$$\chi(\boldsymbol{x}) = \begin{cases} 1 & \text{for } \boldsymbol{x} \text{ such that } x_i < 0.3 \text{ for } i = 1, \ldots, d, \\ 0 & \text{otherwise} \end{cases},$$

which is also depicted in 2D in Figure 6.4.1. The corresponding low-rank approximations have rank 2 for all three formats (CP, Tucker, tensor train).

As a second example, one sample of a stochastic material has been obtained using the truncated Karhunen-Loève expansion [2] of the squared exponential Matérn covariance function [89]. In order to obtain positive definite material coefficients, the exponential function has

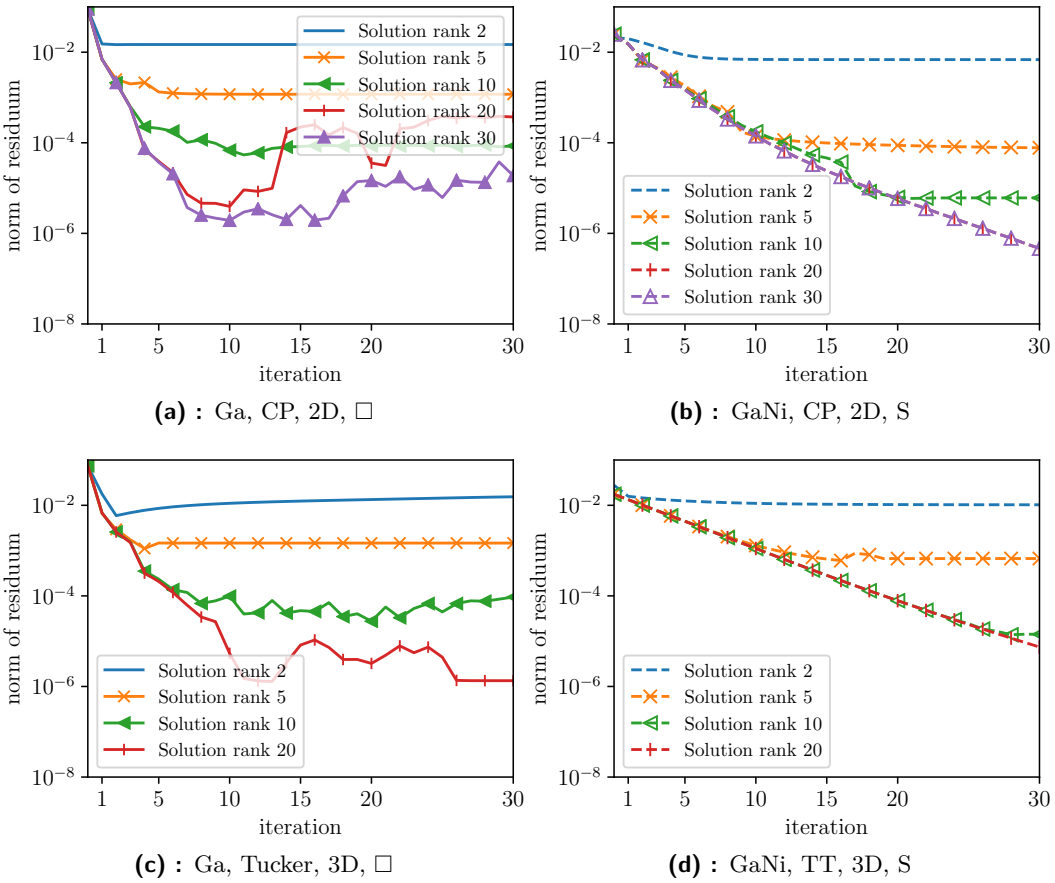
been applied on the expansion, which leads to the following form

$$\boldsymbol{A}_{\mathrm{S}}(\boldsymbol{x}) = \boldsymbol{I} \exp\Big(C + D \sum_{\boldsymbol{k} \in I} c[k] \varphi^{\boldsymbol{k}}(\boldsymbol{x})\Big).$$

The most important modes of the expansion has been selected (20 modes in $2D$ and 26 modes in 3D) and the corresponding frequencies are collected in the index set I. The coefficients $c[k]$ for $k \in I$ has been sampled from uniform distribution on the interval $[-0.5, 0.5]$. The constants C and D scales the material coefficients such that the minimal eigenvalue of $\boldsymbol{A}$ is 1, and the maximal 10. The particular sample that is used for the computation is plotted in Figure 6.4.2. The material coefficients were approximated in low-rank formats with a rank set to 10. For a comparison to the full solution, the full material coefficients have been recovered in order to compute exactly the same problem.

All the numerical problems have been computed with the same number of discretisation grids in each direction $\boldsymbol{N} = [N, \ldots, N] \in \mathbb{R}^d$.

6.4.2 Behaviour of linear systems during iterations



(a) : Ga, CP, 2D, □ **(b) :** GaNi, CP, 2D, S

(c) : Ga, Tucker, 3D, □ **(d) :** GaNi, TT, 3D, S

Figure 6.4.3: Evolution of the norm of residua during minimal residuum iteration; computed in 2D for $N = 1215$ and in 3D for $N = 135$.

The evolution of the norm during the minimal residual iteration is investigated because it describes well the character of the low-rank approximations. The numerical results in

Figure 6.4.3 depict the Euclidean norm of the residuum $\boldsymbol{r} = \boldsymbol{d} - \boldsymbol{C}\boldsymbol{u}_{(i)}$

$$\|\boldsymbol{r}\| = \Big(\sum_{\boldsymbol{k} \in \mathbb{Z}_{\boldsymbol{N}}} \big| r[\boldsymbol{k}] \big|^2 \Big)^{\frac{1}{2}}$$

because it corresponds to the L^2-norm of the corresponding trigonometric polynomial. Note that since the problem is solved in Fourier space, the residuum components agree with the Fourier coefficients of the corresponding trigonometric polynomial.

Although the truncation of the growing tensor's rank can be provided by a tolerance to an approximation error, it is difficult to set up the parameters properly during the solver. Particularly it may happen that the rank significantly increase resulting in unnecessary computational demands, especially when the tensors are far away from the solution. Therefore the truncation has been performed to a fixed rank. The solution which is from a large dimensional space $\mathbb{R}^{\boldsymbol{N}}$ with the dimension $\prod_{\alpha=1}^{d} N_\alpha$ is approximated with a significantly smaller number of parameters. Therefore there is always a residual error which can be diminished only by an increasing rank of the low-rank formats. Note that the rank-one tensors occurring in all three low-rank formats are automatically computed by a solver and are thus suboptimal global basis vectors for the particular problem. Therefore the method can be seen as a model order reduction technique.

From the results in Figure 6.4.3, we can observe that solutions with higher rank have larger potential in reducing the norm of residuum regardless the discretisation method (Ga and GaNi), material problem (□ and S), or the low-rank format (CP, Tucker, TT). This proposes a rank adapting solver that starts with a lower solution rank and increases the rank during the iterations. We also notice that the norms of residuum during iterations decrease with higher rate for the problem with the square inclusion (material □), however, the rate is more stable for the material S. Although, the material □ was systematically computed with GaNi method and material S with Ga, which is in accordance with the recommendation in [149], the discretisation method has no influence on the character of the behaviour during iterations. These finding are in agreement with [91] analysing the stochastic linear systems and solvers approximated with low-rank approximations.

Note that the computation of the Frobenius norm of tensors in Tucker format is computationally demanding. Therefore, we have used the equivalent Frobenius norm of the Tucker's core, which can be computed much faster.

6.4.3 Algebraic error of the low-rank approximations

In the Figure 6.4.4, the approximation properties of the low-rank formats are depicted. As an criterion, the relative algebraic error between the homogenised properties of low-rank solution $A_{\mathrm{H},\boldsymbol{N},r}$ and of the full solution $A_{\mathrm{H},\boldsymbol{N}}$ has been used, i.e.

$$\text{relative error} = \frac{A_{\mathrm{H},\boldsymbol{N}} - A_{\mathrm{H},\boldsymbol{N},r}}{A_{\mathrm{H},\boldsymbol{N}}}.$$

This is chosen because the error in the homogenised properties corresponds to the square of the energetic semi-norm (norm on zero-mean fields) of the algebraic error between the full solution and the low-rank approximation

$$\|u_{\boldsymbol{N}} - u_{\boldsymbol{N},r}\|_A^2 = a\Big(\nabla u_{\boldsymbol{N}} - \nabla u_{\boldsymbol{N},r}, \nabla u_{\boldsymbol{N}} - \nabla u_{\boldsymbol{N},r}\Big) = A_{\mathrm{H},\boldsymbol{N},r} - A_{\mathrm{H},\boldsymbol{N}};$$

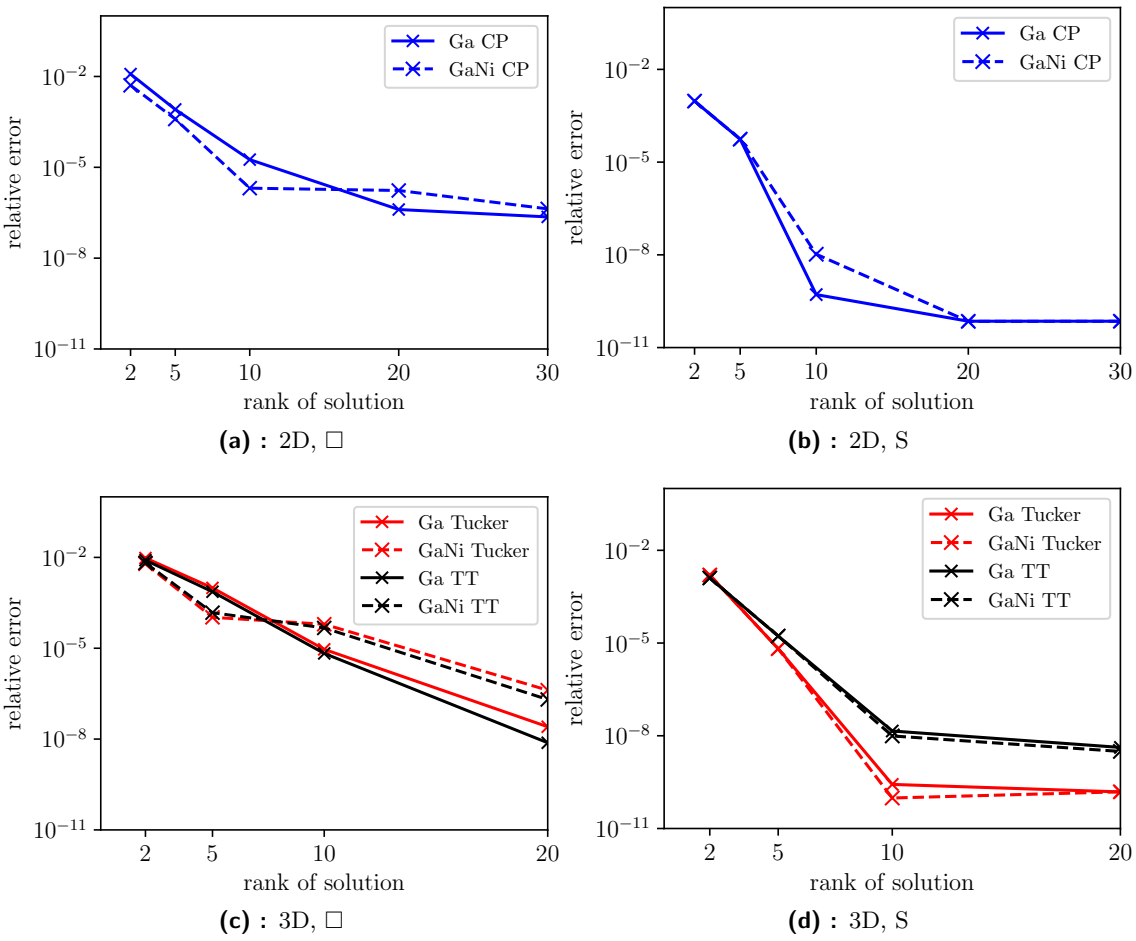


Figure 6.4.4: Relative errors (6.4.3) of low-rank solutions computed in 2D for $N = 1215$ and in 3D for $N = 135$.

for the derivation see [149, Appendix D]. We also note that the full solution $u_{\boldsymbol{N}}$ has been computed using conjugate gradients with high accuracy (tolerance 10^{-8} on the norm of the residuum) to obtain a solution that is close to the exact one. The low-rank solution has been obtained from minimal residual iteration, which was stopped when the residuum failed to be decreased. The minimal residual iteration was used to provide low-rank solution with the minimal norm of residuum.

We can observe that the results are again similar regardless of the discretisation method (Ga and GaNi), material problem (□ and S), or the low-rank format (CP, Tucker, TT). An increase in the solution rank leads to a significant reduction of the relative error. However, the low-rank approximations of the material S reach the threshold error corresponding to the full approximate solution obtained from the conjugate gradients. It also shows that the low-rank method is more accurate for a problem with continuous material property (material S) than for the one with discontinuous coefficients (material □).

6.4.4 Memory and computational efficiencies

Here, we discuss the computational and memory requirements to resolve the linear system using low-rank approximations. Additionally to the previous examples, the CPU times and

Formats \ Operations	Element-wise product	FFT_d	Truncation
full	N^d	$\mathcal{O}(N^d \log N\ )$	—
CP	$dNrs$	$\mathcal{O}(dNr \log N)$	$\mathcal{O}(dNr^2)$
Tucker	$dNrs + r^d s^d$	$\mathcal{O}(dNr \log N)$	$\mathcal{O}(dNr^2 + r^{d+1})$
TT	$dNr^2 s^2$	$\mathcal{O}(dNr^2 \log N)$	$\mathcal{O}(dNr^3)$

Table 6.4.1: Asymptotic computational complexities in terms of floating point multiplications. The operations are performed on full tensors of order d and shape $(N, \ldots, N)$, and on the same tensors in their CP, Tucker, and tensor-train (TT) formats with maximum rank r and s (s for the second operand in a binary operation).

format	memory requirements
full	N^d
CP	dNr
Tucker	$dNr + r^d$
TT	$2Nr + (d-2)Nr^2$

Table 6.4.2: Memory requirements to store tensors of order d with shape $(N, \ldots, N)$ for full, CP, Tucker, and tensor-train (TT) formats with maximum rank r.

approximation properties of low-rank formats were tested for an anisotropic material. The heterogeneous material coefficients $\boldsymbol{A}_\square$ and $\boldsymbol{A}_\mathrm{S}$ were modified by adding a spatially constant anisotropic material tensor $\boldsymbol{B}$, i.e.

$$\widetilde{\boldsymbol{A}}_\bullet(\boldsymbol{x}) = \boldsymbol{A}_\bullet(\boldsymbol{x}) + \boldsymbol{B},$$

where the matrices

$$\boldsymbol{B} = \begin{pmatrix} 5.5 & -4.5 \\ -4.5 & 5.5 \end{pmatrix}, \qquad \boldsymbol{B} = \begin{pmatrix} 4.25 & -3.25 & -1.25\sqrt{2} \\ -3.25 & 4.25 & 1.25\sqrt{2} \\ -1.25\sqrt{2} & 1.25\sqrt{2} & 7.5 \end{pmatrix}$$

have eigenvalues $(1, 10)$ in 2D and $(1, 5, 10)$ in 3D.

	2D, $\square$				3D, $\square$				
N	45	135	405	1215	5	15	45	135	175
r (isotropic)	3	3	5	7	3	3	3	5	5
r (anisotropic)	5	11	21	31	3	3	5	11	11

Table 6.4.3: Rank r of low-rank solutions that reach the same accuracy as the full solution for various values of N, for isotropic and anisotropic material $\square$. The full solver has been computed with grid size $(N, \ldots, N)$ while sparse solver with $(3N, \ldots, 3N)$. The stopping criterion of conjugate gradients for the full solver was set to 10^{-6} on the norm of residuum.

As we are using several low-rank formats and several operations on them, the computational complexities and memory requirements are summarised in Tables 6.4.1 and 6.4.2. The memory requirements of the FFT-based systems are controlled by memory requirements for material coefficients, preconditioner, solution vector, and possibly other vectors needed to store as a requirement of the linear solver. Provided that the ranks are kept small, the memory of low-rank solvers scales linearly with N, while full solver scales with N^d, which makes the method effective particularly for tensor with high order.

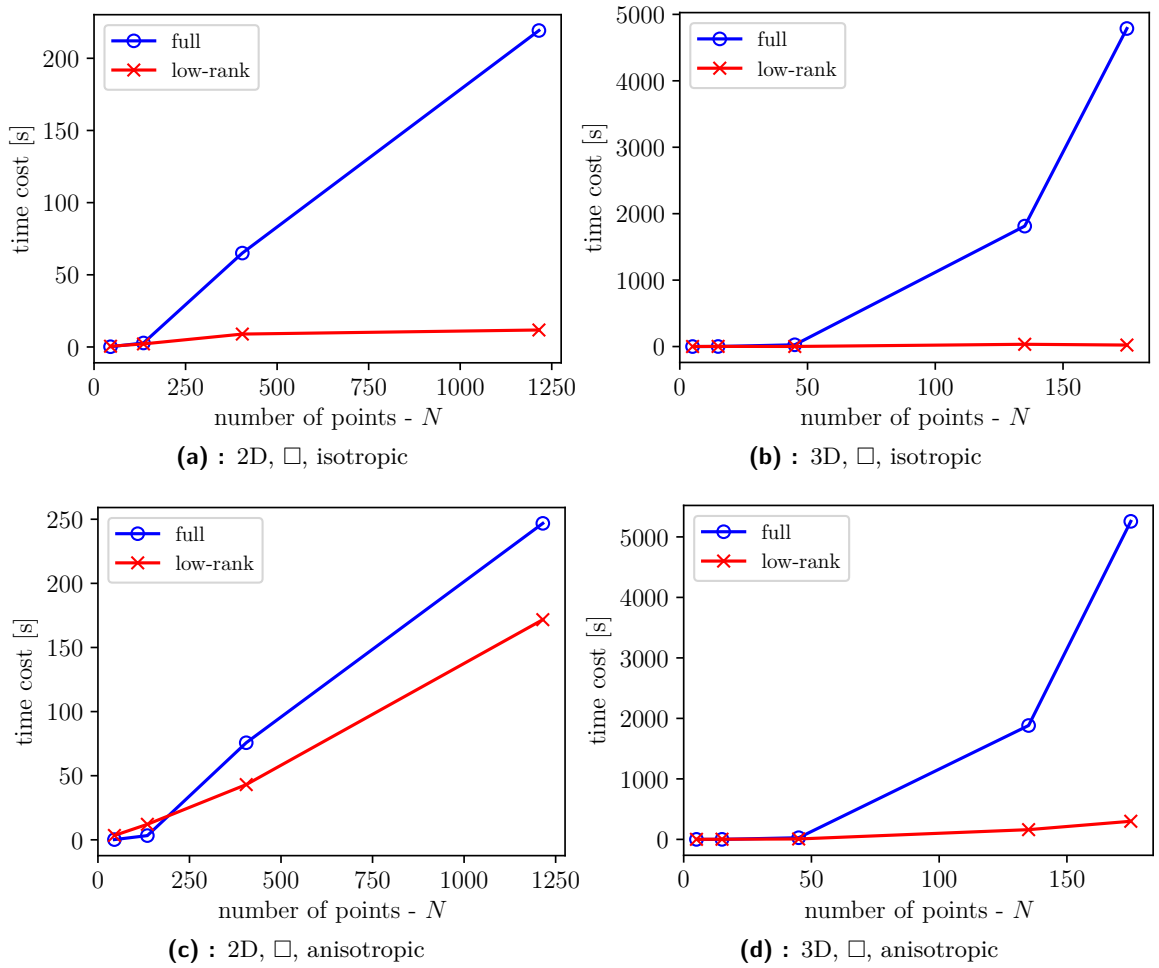


Figure 6.4.5: The CPU time of Ga solver to solve the problem with isotropic (1st row) and anisotropic (2nd row) material □. The full solution has been computed on a grid of size $(N, \ldots, N)$ while the low-rank solution on the grid $(3N, \ldots, 3N)$ with various solution ranks to achieve the same level of accuracy as the full scheme. The stopping criterion of conjugate gradients for the full solver was set to 10^{-6} on the norm of residuum.

We compare the CPU time of full and low-rank solvers for homogenisation with exact integration (Ga) on the same level of accuracy measured by the energetic norm. It is achieved by the following procedure. The reference full solution was computed using conjugate gradient method on the regular grid $(N, \ldots, N)$ with the tolerance 10^{-6} on the norms of residua. In order to achieve the same accuracy as the full solution, the low-rank solver was run on a bigger grid $(\alpha N, \ldots, \alpha N)$ with the multiplier $\alpha = 3$. The rank of low-rank approximations was increased step-by-step until it achieved a required error tolerance defined in (6.4.3). The iterations of the low-rank solver (for a given rank) are stopped when the residuum fails to decrease. This procedure, which creates a great possibility for a rank reduction in the low-rank solution, is applicable only for problems that allow an exact integration of material coefficients (here material □).

The results in Figure 6.4.5 shows that the CPU time scales as N^d for a full solution, and almost linearly for low-rank solutions on the isotropic material □. In the anisotropic cases the time costs of low-rank solutions are relatively higher but still cheaper than that of the

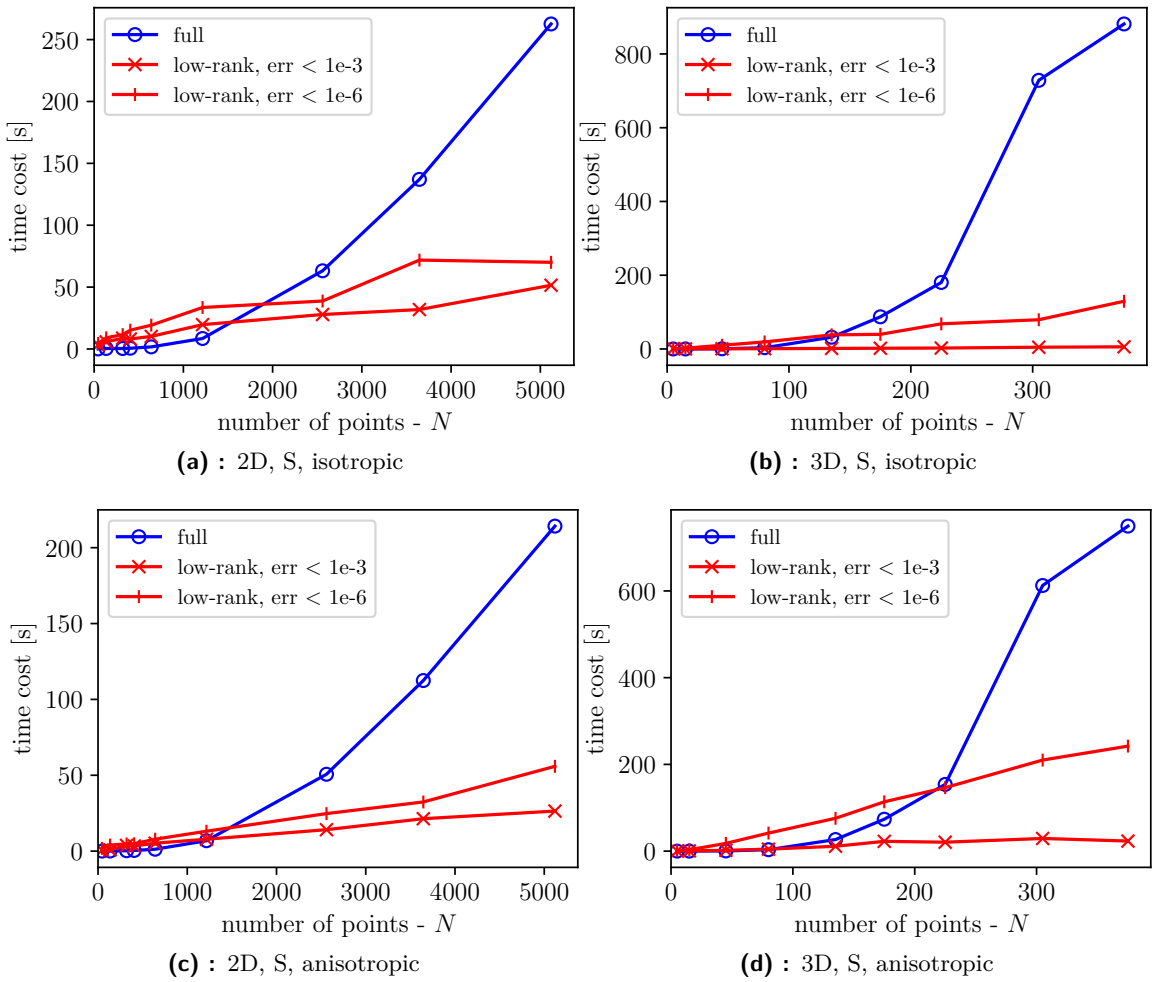


Figure 6.4.6: The CPU time of GaNi solver to solve the problem with isotropic (1st row) and anisotropic (2nd row) material S. Both the full and low-rank solution are computed on the same grid. The stopping criterion of conjugate gradients for the full solver was set to 10^{-6} on the norm of residuum. The minimal residuum iteration for the low-rank approximation was computed with various rank to achieve a required error tolerance defined in (6.4.3).

full solution. The difference in these two cases is due to the different ranks of the low-rank solutions. For isotropic material □, the solution rank increases only slowly with N , while for its anisotropic counterpart the rank increases at a faster rate (as tabulated in the Table 6.4.3). In general, the results show that the low-rank solver are significantly faster for larger N, despite being run on a larger computational grid.

We also did the comparison of both solvers for the isotropic and anisotropic material S. However, the smooth material S is better suited for the homogenisation with the numerical integration (GaNi), see the comparison in [149]. Therefore, the comparison of the solvers is run on the same discretisation grid. The ranks of the low-rank solution are chosen such that it achieves a relative error (as defined in (6.4.3)) below 10^{-3} or 10^{-6}. The ranks remain stable when N increases, which makes the CPU time of the low-rank solver almost linear in N, as shown in Figure 6.4.6.

6.5 Conclusion

This paper is focused on the acceleration of Fourier–Galerkin methods using low-rank tensor approximations for spatially 2-dimensional and 3-dimensional problems of numerical homogenisation. The efficiency of this approach builds on incorporation of the fast Fourier transform (FFT) and low-rank tensor approximation into the iterative linear solvers. The computational complexity is reduced to be quasilinear in the size of the discretisation and linear in spatial dimension d, since on a low-rank tensor of order d, the d-dimensional FFT can be performed as a series of one-dimensional FFTs. In this paper three formats — canonical polyadic (CP), Tucker, and tensor train (TT) — have been considered, and all of them show similar advantage in saving the computational cost.

The main results are summarised as the following:

- The incorporation of low-rank tensor approximations lead to a significant reduction of memory and computational cost in the solution of the homogenisation problems.
- The method is more suitable for material coefficients with relatively smaller rank. The low-rank approximation solvers computationally benefits from the better asymptotic behaviour, see Table 6.4.1 and 6.4.2. The advantage is accentuated for problems of a higher spatial dimension d leading to tensors with order d.
- The low-rank approximation can be seen as a model order reduction technique.

Since the low-rank approximation provides a significant memory reduction it allows to compute the solution on a finer grid. Therefore, the proposed method based on low-rank approximation may provide more accurate solution than the conventional method based on full tensors, especially when the material is of a relatively small rank.

6.A Low-rank tensor approximations

Here we provide more details of the low-rank tensor approximations techniques utilized in this paper. This includes the approximation in CP, Tucker and tensor train formats.

6.A.1 The canonical polyadic format

A canonical polyadic (CP) or r-term representation $\boldsymbol{v}_r$ of a tensor $\boldsymbol{v} \in \mathbb{K}^{N_1 \times \cdots \times N_d}$ ($\mathbb{K}$ is either $\mathbb{R}$ or $\mathbb{C}$) is a sum of r rank-1 tensors, i.e.

$$\boldsymbol{v} \approx \boldsymbol{v}_r = \sum_{i=1}^{r} c[i] \bigotimes_{j=1}^{d} \boldsymbol{b}^{(j)}[i]$$

with $\boldsymbol{b}^{(j)} \in \mathbb{K}^{r \times N_j}$ and $\bigotimes$ denotes tensor product. This format has linear storage size $r \sum_{j=1}^{d} N_j$. But for $d \geq 3$ and a given r, the construction of an error minimizing $\boldsymbol{v}_r$ is not always feasible [53, Proposition 9.10] because the space of CP format tensor with fixed r is not closed [53, Lemma 9.11].

6.A.1.1 Element-wise multiplication

The element-wise (Hadamard) product of two tensors of ranks r and s in CP format is computed as:

$$\boldsymbol{v}_r \odot \boldsymbol{w}_s = \sum_{i=1}^{r}\sum_{k=1}^{s} c_{\boldsymbol{v}}[i]c_{\boldsymbol{w}}[k] \bigotimes_{j=1}^{d} \left(\boldsymbol{b}_{\boldsymbol{v}}^{(j)}[i] \odot \boldsymbol{b}_{\boldsymbol{w}}^{(j)}[k] \right).$$

This operation has complexity $rs\sum_{j=1}^{d} N_j$ and the product has a new rank rs.

6.A.1.2 Fourier transform

Due to the linearity and tensor structure of the Fourier transform $\mathcal{F}_{\boldsymbol{N}}$ of a size $\boldsymbol{N} \in \mathbb{N}^d$, a d-dimensional Fourier transform of a CP tensor is broken down to a series of 1-d Fourier transform, i.e.,

$$\mathcal{F}_{\boldsymbol{N}}(\boldsymbol{v}_r) = \sum_{i=1}^{r} c[i] \bigotimes_{j=1}^{d} \mathcal{F}_{N_j}(\boldsymbol{b}^{(j)}[i]).$$

Hence a FFT on a CP tensor has complexity $drN \log N$.

6.A.1.3 Rank truncation

Operations (e.g. element-wise multiplication) applied on tensors in CP format usually inflate the representation rank. This calls for a truncation to a prescribed rank or error tolerance.

For $d = 2$, this reduction is done by rank truncation based on QR decomposition and singular value decomposition(SVD). Let the matrices $\boldsymbol{B}^{(j)} \in \mathbb{K}^{N_j \times r}$ collect the vectors $\{\boldsymbol{b}^{(j)}[i]\}_{i=1}^{r}$ for the j-th dimension, we have their re-orthogonalisations $\boldsymbol{B}^{(1)} = \boldsymbol{Q}^{(1)}\boldsymbol{R}^{(1)}$ and $\boldsymbol{B}^{(2)} = \boldsymbol{Q}^{(2)}\boldsymbol{R}^{(2)}$ by QR decompositions. A SVD $\boldsymbol{R}^{(1)}\boldsymbol{R}^{(2)} = \boldsymbol{U}^{(1)}\boldsymbol{\Sigma}(\boldsymbol{U}^{(2)})^{\top}$ facilitates the truncation. Suppose $\boldsymbol{U}_k^{(1)}$, $\boldsymbol{U}_k^{(2)}$ and $\boldsymbol{\Sigma}_k$ are the truncated ones with rank $k \leq r$, the truncated form of the CP representation (6.A.1) is

$$\boldsymbol{v}_k = \sum_{i=1}^{k} c[i]\widehat{\boldsymbol{b}}^{(1)}[i] \otimes \widehat{\boldsymbol{b}}^{(2)}[i]$$

where $\widehat{\boldsymbol{b}}^{(1)}[i]$, $\widehat{\boldsymbol{b}}^{(2)}[i]$ are the columns of $\boldsymbol{Q}^{(1)}\boldsymbol{U}_k^{(1)}$, $\boldsymbol{Q}^{(2)}\boldsymbol{U}_k^{(2)}$ respectively, and $c[i]$ are the diagonal entries of $\boldsymbol{\Sigma}_k$.

For $d \geq 3$, the k-rank form could be obtained by numerical error minimizing procedures [53], e.g. Alternative Least-Squares method. But there is no guarantee that the procedures would converge, and if they would, there is no guarantee that they converge to the global optimum. This is due to the non-closedness of the set of rank-r CP tensors with $d \geq 3$.

6.A.2 Tucker format

A Tucker format representation (or tensor subspace representation) of a tensor $\boldsymbol{v} \in \mathbb{K}^{N_1 \times \cdots \times N_d} \in \mathcal{V}$ is a linear combination of frames (usually orthogonal bases) of the tensor space $\mathcal{V}$. Suppose $\mathcal{V} = \bigotimes_{j=1}^{d} \mathcal{V}^j$, the subspace $\mathcal{V}^j$ has basis vectors $\{\boldsymbol{b}^{(j)}[i_j] \in \mathbb{K}^{N_j} : 1 \leq i_j \leq r_j\}$ with ranks $\boldsymbol{r} = (r_1, \ldots, r_d)$. The tensors $\bigotimes_{j=1}^{d} \boldsymbol{b}^{(j)}[i_j]$ for all $1 \leq i_j \leq r_j$ form the bases of the space $\mathcal{V}$. Then we have a unique coefficient $c[i_1, i_2, \ldots, i_d]$ for every $\boldsymbol{v} \in \mathcal{V}$ such that

$$\boldsymbol{v} \approx \boldsymbol{v}_{\boldsymbol{r}} = \sum_{i_1=1}^{r_1} \cdots \sum_{i_d=1}^{r_d} c_{\boldsymbol{v}}[i_1, i_2, \ldots, i_d] \bigotimes_{j=1}^{d} \boldsymbol{b}_{\boldsymbol{v}}^{(j)}[i_j],$$

where $\boldsymbol{c} \in \mathbb{K}^{r_1 \times \cdots \times r_d}$ is called the core tensor. Given any prescribed rank vector $\boldsymbol{r}$, an error minimizing approximation $\boldsymbol{v}_{\boldsymbol{r}}$ can be found by a *high-order singular value decomposition* (HOSVD) [26]. When the vectors $\{\boldsymbol{b}^{(j)}[i_j] \in \mathbb{K}^{N_j} : 1 \le i_j \le r_j\}$ form only a frame of the subspace $\mathcal{V}$ (e.g. after addition of two tensors), the core tensor is not unique, however, a representation with orthogonal bases can be obtained by applying QR decomposition to the frames and HOSVD to the accordingly updated core.

6.A.2.1 Element-wise multiplication

Let another Tucker format tensor with rank $\boldsymbol{s}$ be defined as

$$\boldsymbol{w}_{\boldsymbol{s}} = \sum_{\boldsymbol{k}}^{\boldsymbol{s}} \boldsymbol{c}_{\boldsymbol{w}}[\boldsymbol{k}] \bigotimes_{j=1}^{d} \boldsymbol{b}_{\boldsymbol{w}}^{(j)}[k_j]$$

the element-wise (Hadamard) product of $\boldsymbol{v}_{\boldsymbol{r}}$ and $\boldsymbol{w}_{\boldsymbol{s}}$ has also a Tucker format

$$\boldsymbol{v}_{\boldsymbol{r}} \odot \boldsymbol{w}_{\boldsymbol{s}} = \sum_{\boldsymbol{l}}^{\boldsymbol{t}} \boldsymbol{c}[\boldsymbol{l}] \bigotimes_{j=1}^{d} \boldsymbol{b}^{(j)}[l_j]$$

where $\boldsymbol{t} = \boldsymbol{r} \odot \boldsymbol{s}$ and $\boldsymbol{c} = \boldsymbol{c}_{\boldsymbol{v}} \otimes \boldsymbol{c}_{\boldsymbol{w}}$, i.e. the Kronecker product of the two coefficient tensors. So for any $1 \le j \le d$, the index l_j is related to i_j and k_j by $l_j = \overline{i_j k_j} = i_j r_j + k_j$, and u is obtained from v and w through

$$u_{l_j}^{(j)} = u_{\overline{i_j k_j}}^{(j)} = v_{i_j}^{(j)} \odot w_{k_j}^{(j)} \quad \text{for} \quad 1 \le i_j \le r_j,\ 1 \le k_j \le s_j$$

Let $N = \max_i N_i$, $r = \max_i r_i$ and $s = \max_i s_i$, the computational complexity of the element-wise product is bounded by $dNrs + r^d s^d$, in which the first term is the cost for computing $\{u_{l_j}^{(j)} : 1 \le l_j \le R_j\}_{j=1}^{d}$, and the second for the Kronecker product of coefficient tensors.

6.A.2.2 Fourier transform

The Fourier transform of $\boldsymbol{v}_{\boldsymbol{r}}$ is

$$\mathcal{F}_{\boldsymbol{N}}(\boldsymbol{v}_{\boldsymbol{r}}) = \sum_{\boldsymbol{i}}^{\boldsymbol{r}} \boldsymbol{c}[\boldsymbol{i}] \bigotimes_{j=1}^{d} \mathcal{F}_{N_j}(\boldsymbol{b}^{(j)}[i_j])$$

which only involves the basis vectors. If FFT is applied, the complexity is of order $\mathcal{O}(drN \log N)$.

6.A.2.3 Rank truncation

The Tucker representation (6.A.2) can be obtained either by a HOSVD applied on a full tensor or by an operation (e.g. element-wise multiplication) over other Tucker operands. In the first case, an error minimizing rank truncation is readily available due to the property of HOSVD:

$$\sigma_1^{(j)} \ge \sigma_2^{(j)} \ge \cdots \ge \sigma_{r_j}^{(j)}, \quad \text{for } j = 1, \cdots, d,$$

where $\sigma_{i_j}^{(j)}$ is the 2-norm of the i_j-th slice of the core tensor $\boldsymbol{c}$ cut on the j-th dimension. If the truncation rank is $k_j < r_j$, the error of the truncated representation $\boldsymbol{v}_{\boldsymbol{k}}$ is bounded by

$$\|\boldsymbol{v}_{\boldsymbol{r}} - \boldsymbol{v}_{\boldsymbol{k}}\| \le \left[\sum_{j=1}^{d} \sum_{i=k_j+1}^{r_j} (\sigma_i^{(j)})^2 \right]^{1/2}.$$

In the second case the bases $\{\boldsymbol{b}^{(j)}[i_j]\}_{i_j=1}^{r_j}$ have to be re-orthogonalised first, and then a HOSVD of the updated core tensor is to be made to facilitate the truncation as in the first case. This procedure [53, as detailed in] is analogues to the re-orthogonalisation and SVD for the 2D CP format representations, but with higher tensor order.

6.A.3 Tensor train format

A tensor train(TT) representation [115] of a tensor $\boldsymbol{v} \in \mathbb{K}^{N_1 \times \cdots \times N_d}$ can be expressed as a series of consecutive contractions of tensors $\boldsymbol{b}^{(j)} \in \mathbb{K}^{r_{j-1} \times N_j \times r_j}$ of order 3 for $j = 1, \cdots, d$, which are the *carriages* of the tensor train. An equivalent expression in the form of tensor products is

$$\boldsymbol{v} \approx \boldsymbol{v}_{\boldsymbol{r}} = \sum_{i_1=1}^{r_1} \cdots \sum_{i_{d-1}=1}^{r_{d-1}} \boldsymbol{b}_{\boldsymbol{v}}^{(1)}[1, :, i_1] \otimes \boldsymbol{b}_{\boldsymbol{v}}^{(2)}[i_1, :, i_2] \otimes \cdots \otimes \boldsymbol{b}_{\boldsymbol{v}}^{(d)}[i_{d-1}, :, 1]$$

$\boldsymbol{r}$ is the TT-rank of $\boldsymbol{v}$ with a constrain $r_0 = r_d = 1$ to keep the elements of $\boldsymbol{v}$ scalars. The TT format is stable in the sense that for any prescribed $\boldsymbol{r}$ an error minimizing $\boldsymbol{v}_{\boldsymbol{r}}$ can always be constructed by a series of SVDs on consecutive matricisations of $\boldsymbol{v}$.

6.A.3.1 Element-wise multiplication

Let another TT format tensor with rank $\boldsymbol{s}$ be defined as

$$\boldsymbol{w}_{\boldsymbol{s}} = \sum_{i_1=1}^{s_1} \cdots \sum_{i_{d-1}=1}^{s_{d-1}} \boldsymbol{b}_{\boldsymbol{w}}^{(1)}[1, :, i_1] \otimes \boldsymbol{b}_{\boldsymbol{w}}^{(2)}[i_1, :, i_2] \otimes \cdots \otimes \boldsymbol{b}_{\boldsymbol{w}}^{(d)}[i_{d-1}, :, 1]$$

with $\boldsymbol{b}_{\boldsymbol{w}}^{(j)} \in \mathbb{K}^{s_{j-1} \times N_j \times s_j}$. The element-wise product of $\boldsymbol{v}_{\boldsymbol{r}}$ and $\boldsymbol{w}_{\boldsymbol{s}}$ can also be expressed in TT format:

$$\boldsymbol{v}_{\boldsymbol{r}} \odot \boldsymbol{w}_{\boldsymbol{s}} = \sum_{i_1=1}^{t_1} \cdots \sum_{i_{d-1}=1}^{t_{d-1}} \boldsymbol{b}^{(1)}[1, :, i_1] \otimes \boldsymbol{b}^{(2)}[i_1, :, i_2] \otimes \cdots \otimes \boldsymbol{b}^{(d)}[i_{d-1}, :, 1]$$

where $\boldsymbol{t} = \boldsymbol{r} \odot \boldsymbol{s}$ and $\boldsymbol{b}^{(j)} = \boldsymbol{b}_{\boldsymbol{v}}^{(j)} * \boldsymbol{b}_{\boldsymbol{w}}^{(j)}$. Here the $*$ denotes one type of Khatri–Rao product [65] which makes Kronecker product only in the first and third dimensions, i.e. it yields an order 3 tensor $\boldsymbol{b}^{(j)} \in \mathbb{K}^{r_{j-1} s_{j-1} \times N_j \times r_j s_j}$. The complexity of the element-wise product is of order $\mathcal{O}(dNr^2s^2)$ with N, r and s as defined in the subsection 6.A.2.

6.A.3.2 Fourier transform

The Fourier transform of $\boldsymbol{v}_{\boldsymbol{r}}$ can also be carried out by doing 1-D transforms on each *carriage*:

$$\mathcal{F}_{\boldsymbol{N}}(\boldsymbol{v}_{\boldsymbol{r}}) = \sum_{i_1=1}^{r_1} \cdots \sum_{i_{d-1}=1}^{r_{d-1}} \mathcal{F}_{N_1}(\boldsymbol{b}_{\boldsymbol{v}}^{(1)}[1, :, i_1]) \otimes \mathcal{F}_{N_2}(\boldsymbol{b}_{\boldsymbol{v}}^{(2)}[i_1, :, i_2]) \otimes \cdots \otimes \mathcal{F}_{N_d}(\boldsymbol{b}_{\boldsymbol{v}}^{(d)}[i_{d-1}, :, 1])$$

in which the $\mathcal{F}_{N_j}(\cdot)$ is made on the fibres along the second mode. If FFT is applied here, the number of operations is of order $\mathcal{O}(dr^2 N \log N)$.

6.A.3.3 Rank truncation

The tensor train representation (6.3.1.3) can be obtained either by transforming a full tensor into tensor train format by using $d-1$ sequential SVDs applied on auxiliary matrices of the tensor (known as TT-SVD) [115], or as a result of operations (e.g. additions or multiplications) over tensor train operands. In the first case, an error minimising rank truncation could be directly carried out in the TT-SVD process. The truncation has an error bound $(\sum_{k=1}^{d-1} \epsilon_k^2)^{1/2}$, where ϵ_k is the Frobenious norm error introduced by the truncation of the k-th SVD. In the second case, a re-orthogonalisation has to be done in the first place, this is followed by $d-1$ sequential SVDs on unfolded *carriages*. This process is known as TT-truncation (also called rounding).

For the first case, the complexity of truncation is the same as that for the TT-SVD, which is of order $\mathcal{O}(N^{d+1})$. A cheaper alternative for TT-SVD is TT-cross approximation as introduced in [114]. The complexity of TT-truncation in the second case is of order $\mathcal{O}(dNr^3)$.

Chapter 7

Conclusions

This thesis aimed at contributing to a deeper understanding and development of spectral methods for computational homogenization of periodic microstructures. In particular, we focused on three major research objectives.

(i) Understanding the effect of discrete Green's operator preconditioning.

(ii) Minimization of discretization artifacts of spectral methods.

(iii) Reduction of computational requirements of spectral methods.

We discussed these topics in the form of the collection of five manuscripts adapted to **Chapters 2-6**.

In the first part, related to objective (i), we focused on the effect of the discrete Green's operator preconditioner on the mesh-independent convergence rate of the conjugate gradient (CG) method. The CG method benefits from the clustering of eigenvalues (at least in exact arithmetic). We inspect the spectrum of the preconditioned linear system arising from discretization of homogenization problems, i.e., elliptic partial differential equations. In **Chapter 2**, we analyze the distribution of eigenvalues of general diffusion or elasticity problems, discretized by the conforming FE method and preconditioned by the discrete Green's operator of the reference problem. We provided a constructive proof that bounds on these eigenvalues are defined by the local properties of the material data and the reference material data. Bounds are obtained from the data on supports of FE basis functions. Therefore, these eigenvalue bounds are independent of the characteristic element size, which suggests that the condition number (ratio of the biggest and smallest eigenvalue) of the preconditioned linear system is independent of the problem size. FE basis functions (the corresponding degrees of freedom) are connected to the same eigenvalues if their supports are inside a subdomain where the material and the reference material do not change. Therefore, mesh refinement in the interior of a homogeneous subdomain does not generate additional distinct eigenvalues (in exact arithmetic).

We proposed an algorithm that provides guaranteed, two-sided, easily accessible eigenvalues bounds. For pixel/voxel representation of geometry with element-wise constant data, our approach provides eigenvalue bounds in linear complexity, i.e., $O(n)$ arithmetic operations must be performed. This affordable guaranteed lower bound on the smallest eigenvalue gives access to better control of solution precision over the iterative process, provided by accurate algebraic error estimates; see, e.g., [94]. Additionally, we investigated how the choice of the reference material affects the convergence of the CG method. The provided proof of the direct correspondence between the reference material, material of the problem, and the resulting eigenvalues renders the optimization of the reference material data more accessible. The

closer the reference material is to the real material of the sample, the better conditioning the preconditioned discretized problem has and the spectrum contains a small number of clusters, recall, e.g., Section 2.7. However, the FFT technique is, up to now, restricted to homogeneous reference problems. We also experimentally observed that the weighted mean values reference material can reduce the number of CG iterations compared to the conservative choice with identity matrix; recall, e.g., Section 4.6.3.

The mesh-size independent CG iteration count is observable also for other, non-Galerkin, discretization schemes. Therefore, in **Chapter 3**, we extended the results of **Chapter 2** beyond the Galerkin approach with FE basis functions. We used the assumption that the global matrix of the linear system can be obtained as a sum of local symmetric positive semidefinite matrices. In these cases, the eigenvalue bounds depend solely on local material data and on connections between the degrees of freedom, i.e. on the properties of the discretization. We demonstrated the approach of obtaining the eigenvalues bounds for the finite difference method, the stochastic Galerkin FE method, and the method of algebraic multilevel preconditioning.

In the second part, related to objective (ii), we focused on minimizing discretization artifacts of FFT-based methods. In **Chapter 4**, we derived a micromechanical solver in a standard FE manner, enhanced by a discrete Green's operator preconditioner. We provided a linear algebra-based viewpoint on the discretization of the micromechanical problems, i.e. elliptic PDEs. We explained how regular FE discretization preserves the efficient block-circulant structure of the preconditioner, which allows us to use the FFT technique for its efficient inversion and application. Thanks to FFT, FE discretization maintains the quasilinear computational complexity typical of spectral homogenization solvers.

Using an exactly integrated FE scheme delivered ringing-free results. However, this accuracy comes at the cost of higher memory consumption because the exactly integrated trilinear element needs 8 quadrature points (or at least 5 quadrature points for 5 linear tetrahedral elements). This expands the memory requirements of the FE scheme in comparison to standard spectral schemes with 1 quadrature points. However, the scheme is derived in a flexible manner and allows for the use of an arbitrary integration rule. We tested an under-integrated trilinear FE with 1 quadrature point per element/voxel. However, this more memory efficient scheme does not completely eliminate discretization artifacts, as was explained later in **Chapter 5**. This observation suggests that the price of higher memory cost has to be paid for simulations of localized phenomena.

Additionally, we extended the range of FFT-based methods from simple regular grids (one discretization node per pixel/voxel) to more general regular grids (multiple discretization nodes per pixel/voxel), recall Fig. 4.2. This, e.g., allows discretizations with equilateral triangles with minimal mesh anisotropy, which is useful for modeling of crack propagation in the concrete. We also showed the equivalence between our displacement-based scheme and the strain-based scheme with the FE projection operator, used in **Chapter 5**. This readily extends the application range of our method for obtaining the eigenvalue bounds discussed in **Chapter 2**.

Chapter 5 followed the discussion on ringing artifacts that naturally appear in the solutions of the Fourier-Galerkin discretization. In this chapter, we used the strain-based formulation that considers the deformation gradient as the primary degree of freedom, and enforce the compatibility of gradient fields with the compatibility projection operator, recall Section 5.2.1. We derived a general formulation of the projection operator for arbitrary gradient stencils, and described the derivation of finite-difference stencils, a least-square stencil, and a FE stencil.

We showed in Section 5.3.4 that the ringing phenomena pollutes the solution stress-strain field and cannot be used for modeling of problems with localized deformations. Beside

the Gibbs ringing of Fourier basis, the solution is prone to ringing artifacts arising from missing degrees of freedom in the formulation of the deformation gradient. This phenomenon occurs because a single deformation gradient tensor is not enough to describe all admissible deformations of a pixel/voxel. One of the main results of the **Chapter 5** is the observation that for full elimination of all ringing artifacts we need gradient stencils that are equivalent to FE discretization. This speaks in favor of the FE scheme from **Chapter 4**.

The third part, related to objective (iii), was dedicated to the reduction of computational costs of spectral methods. High-resolution 3D microstructures are extremely memory demanding, and handling such datasets is still unaffordable for widespread use.

In **Chapter 6**, we explored the potential of low-rank tensor approximation techniques in the context of spectral solvers. Low-rank tensors can sparsely express d-dimensional fields as the outer products of d vectors. This makes them suitable for spectral methods with the regular discretization grids, which also have an outer product structure. Standard arithmetic operations such as addition, element-wise multiplication, or FFT can be performed with low-rank tensors, thus the whole numerical solution process can be performed directly in a compressed, low-rank format. Performing such operations can increase the representation rank of the tensors, which requires rank truncation, i.e. their reparametrisation with a smaller rank while keeping a reasonable accuracy [115, 13].

We studied performances of the canonical polyadic, Tucker, and Tensor-Train low-rank tensors format [54, 72] on the series of scalar linear elliptic homogenization problems. We showed that the memory and computational costs of the FFT-based methods can be significantly reduced while keeping rounding errors at an acceptable level. In this chapter, we showed the potential of an efficient reduced-order modeling that may be attractive for large-scale engineering problems.

7.1 Perspectives for future research

We showed in **Chapter 2** and **Chapter 4** that the proper choice of the reference material can lead to a significant decrease in the number of CG iterations. The change of the reference material in the displacement-based scheme is straightforward. However, in the strain-based scheme a simple change of reference material in a discrete Green's operator can significantly slow down convergence, if the method is not implemented properly. Therefore, we are currently working on manuscript that discusses this problem and suggests an alternative implementation strategy for strain-based scheme, solved by the CG method.

Combining the FE scheme of **Chapter 4** with the low-rank tensor technique of **Chapter 6** can deliver promising results and is in the scope of my near-future work. Furthermore, in **Chapter 6**, we observed that Krylov solvers improve the solution with a rank-one update in every iteration. This feature can be utilized to speed up the solver by an adaptive rank strategy. I intend to explore and exploit this observation to speed up the low-rank technique.

An effective homogeneous algorithmic tangent can be obtained from solutions to micromechanical problems. Guaranteed upper and lower bounds on effective tangents are necessary to quantify the quality of obtained numerical results. The upper bound is easily accessible (based on minimization principles) from the solutions of the primal (original) homogenization problems. However, the lower bound requires solutions to the dual homogenization problems. In an upcoming manuscript, we discuss the problem of guaranteed upper and lower bounds on effective tangents for FE discretization in 3D. We derive a technique for the construction of proper FE approximation subspaces that are necessary for solving the dual problems.

Bibliography

[1] N. Aage, E. Andreassen, B. S. Lazarov, and O. Sigmund. Giga-voxel computational morphogenesis for structural design. *Nature*, 550(7674):84–86, 2017. DOI: 10.1038/nature23911.

[2] R. J. Adler and J. E. Taylor. *Random fields and geometry*. Springer, New York, 2009.

[3] O. Axelsson. *Iterative solution methods*. Cambridge University Press, 1996. DOI: 10.1017/CBO9780511624100.

[4] O. Axelsson and J. Karátson. Equivalent operator preconditioning for elliptic problems. *Numerical Algorithms*, 50(3):297–380, 2009. DOI: 10.1007/s11075-008-9233-4.

[5] I. Babuška, R. Tempone, and G. E. Zouraris. Galerkin finite element approximations of stochastic elliptic partial differential equations. *SIAM Journal on Numerical Analysis*, 42(2):800–825, 2004. DOI: 10.1137/S0036142902418680.

[6] J. Ballani and L. Grasedyck. A projection method to solve linear systems in tensor format. *Numerical Linear Algebra with Applications*, 20(1):27–43, 2013. DOI: 10.1002/nla.1818.

[7] M. Bebendorf. Approximation of boundary element matrices. *Numerische Mathematik*, 86(4):565–589, 2000. DOI: 10.1007/PL00005410.

[8] T. Belytschko, W. K. Liu, B. Moran, and K. I. Elkhodary. *Nonlinear finite elements for continua and structures*. Wiley, 2014. ISBN 9781118632703.

[9] A. Bensoussan, J.-L. Lions, and G. Papanicolaou. *Asymptotic analysis for periodic structures*. North Holland, Amsterdam, 1978. ISBN 0080875262.

[10] N. Bertin and L. Capolungo. A FFT-based formulation for discrete dislocation dynamics in heterogeneous media. *Journal of Computational Physics*, 355:366–384, 2018. DOI: 10.1016/J.JCP.2017.11.020.

[11] A. Bespalov, D. Loghin, and R. Youngnoi. Truncation preconditioners for stochastic Galerkin finite element discretizations. *SIAM Journal on Scientific Computing*, 43(5): S92–S116, 2021. DOI: 10.1137/20M1345645.

[12] B. Bialecki and A. Karageorghis. Finite difference schemes for the Cauchy–Navier equations of elasticity with variable coefficients. *Journal of Scientific Computing*, 62(1): 78–121, 2015. DOI: 10.1007/s10915-014-9847-8.

[13] D. Bigoni, A. P. Engsig-Karup, and Y. M. Marzouk. Spectral tensor-train decomposition. *SIAM Journal on Scientific Computing*, 38(4):A2405–A2439, 2016. DOI: 10.1137/15M1036919.

[14] R. Blaheta. Displacement decomposition—incomplete factorization preconditioning techniques for linear elasticity problems. *Numerical Linear Algebra with Applications*, 1 (2):107–128, 1994. DOI: 10.1002/nla.1680010203.

[15] R. Blaheta, S. Margenov, and M. Neytcheva. Uniform estimate of the constant in the strengthened CBS inequality for anisotropic non-conforming FEM systems. *Numerical Linear Algebra with Applications*, 11(4):309–326, 2004. DOI: 10.1002/nla.350.

[16] M. Boeff, F. Gutknecht, P. S. Engels, A. Ma, and A. Hartmaier. Formulation of nonlocal damage models based on spectral methods for application to complex microstructures. *Engineering Fracture Mechanics*, 147:373–387, 2015. DOI: 10.1016/j.engfracmech.2015.06.030.

[17] J. P. Boyd. *Chebyshev and Fourier spectral methods*. Dover Publications, New York, 2000.

[18] S. Brisard and L. Dormieux. FFT-based methods for the mechanics of composites: A general variational framework. *Computational Materials Science*, 49(3):663–671, 2010. DOI: 10.1016/j.commatsci.2010.06.009.

[19] S. Brisard and L. Dormieux. Combining Galerkin approximation techniques with the principle of Hashin and Shtrikman to derive a new FFT-based numerical method for the homogenization of composites. *Computer Methods in Applied Mechanics and Engineering*, 217-220:197–212, 2012. DOI: 10.1016/j.cma.2012.01.003.

[20] G. F. Cagey and W. F. Spotz. Higher-order compact mixed methods. *Communications in Numerical Methods in Engineering*, 13(7):553–564, 1997. DOI: 10.1002/(SICI)1099-0887(199707)13:7<553::AID-CNM80>3.0.CO;2-O.

[21] P. G. Ciarlet. *Mathematical elasticity, three-dimensional elasticity*. Elsevier Science Publishers B. V., 1988. ISBN 0444-702598. DOI: 10.1002/crat.2170250509.

[22] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19:297–301, 1965. DOI: 10.1090/S0025-5718-1965-0178586-1.

[23] A. J. Crowder and C. E. Powell. CBS constants & their role in error estimation for stochastic Galerkin finite element methods. *Journal of Scientific Computing*, 77(2):1030–1054, 2018. DOI: 10.1007/s10915-018-0736-4.

[24] T. W. J. de Geus, R. H. J. Peerlings, and M. G. D. Geers. Competing damage mechanisms in a two-phase microstructure: How microstructure and loading conditions determine the onset of fracture. *International Journal of Solids and Structures*, 97:687–698, 2016. DOI: 10.1016/j.ijsolstr.2016.03.029.

[25] T.W.J. de Geus, J. Vondřejc, J. Zeman, R.H.J. Peerlings, and M.G.D. Geers. Finite strain FFT-based non-linear solvers made simple. *Computer Methods in Applied Mechanics and Engineering*, 318:412–430, 2017. DOI: 10.1016/j.cma.2016.12.032.

[26] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000. DOI: 10.1137/S0895479896305696.

[27] M. K. Deb, I. M. Babuška, and J.T. Oden. Solution of stochastic partial differential equations using Galerkin finite element techniques. *Computer Methods in Applied Mechanics and Engineering*, 190(48):6359–6372, 2001. DOI: 10.1016/S0045-7825(01)00237-7.

[28] F. Dietrich, D. Merkert, and B. Simeon. *Derivation of higher-order terms in FFT-based numerical homogenization*, volume 126. Springer International Publishing, 2019. DOI: 10.1007/978-3-319-96415-7_25.

[29] S. V. Dolgov. TT-GMRES: Solution to a linear system in the structured tensor format. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 28(2):149–172, 2013. DOI: 10.1515/rnam-2013-0009.

[30] C. Dávila, Ch. Rose, and P. Camanho. A procedure for superposing linear cohesive laws to represent multiple damage mechanisms in the fracture of composites. *International Journal of Fracture*, 158:211–223, 2009. DOI: 10.1007/s10704-009-9366-z.

[31] V. Eijkhout and P. Vassilevski. The role of the strengthened Cauchy-Buniakowskii-Schwarz inequality in multilevel methods. *SIAM Rev.*, 33:405–419, 1991. DOI: 10.1137/1033098.

[32] A. Ern and J.-L. Guermond. *Theory and practice of finite elements*. Applied Mathematical Sciences. Springer, New York, 2004. ISBN 978-0-387-20574-8. DOI: 10.1007/978-1-4757-4355-5.

[33] J. D. Eshelby. The determination of the elastic field of an ellipsoidal inclusion, and related problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 241(1226):376–396, 1957. DOI: 10.1098/rspa.1957.0133.

[34] J. D. Eshelby. The elastic field outside an ellipsoidal inclusion. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 252(1271):561–569, 1959. DOI: 10.1098/rspa.1959.0173.

[35] M. Espig, W. Hackbusch, A. Litvinenko, H. G. Matthies, and P. Wähnert. Efficient low-rank approximation of the stochastic Galerkin matrix in tensor formats. *Computers and Mathematics with Applications*, 67(4):818–829, 2014. DOI: 10.1016/j.camwa.2012.10.008.

[36] D. J. Eyre and G. W. Milton. A fast numerical scheme for computing the response of composites using grid refinement. *The European Physical Journal Applied Physics*, 6 (1):41–47, 1999.

[37] J. H. Ferziger and M. Perić. *Computational methods for fluid dynamics*. Springer, 2002. ISBN 978-3-540-42074-3. DOI: 10.1007/978-3-642-56026-2.

[38] J. Fish, G. J. Wagner, and S. Keten. Mesoscopic and multiscale modelling in materials. *Nature Materials*, 20:774–786, 2021. DOI: 10.1038/s41563-020-00913-0.

[39] I. Fried. Bounds on the spectral and maximum norms of the finite element stiffness, flexibility and mass matrices. *International Journal of Solid Structures*, 9:1013–1034, 1973.

[40] M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. DOI: 10.1109/JPROC.2004.840301.

[41] S. Gajek, M. Schneider, and T. Böhlke. On the micromechanics of deep material networks. *Journal of the Mechanics and Physics of Solids*, 142:103984, 2020. DOI: 10.1016/j.jmps.2020.103984.

[42] C. Garcia-Cardona, R. Lebensohn, and M. Anghel. Parameter estimation in a thermoelastic composite problem via adjoint formulation and model reduction. *International Journal for Numerical Methods in Engineering*, 112(6):578–600, 2017. DOI: 10.1002/nme.5530.

[43] L. Gélébart and R. Mondon-Cancel. Non-linear extension of FFT-based methods accelerated by conjugate gradients to evaluate the mechanical behavior of composite materials. *Computational Materials Science*, 77:430–439, 2013. DOI: 10.1016/j.commatsci.2013.04.046.

[44] T. Gergelits and Z. Strakoš. Composite convergence bounds based on Chebyshev polynomials and finite precision conjugate gradient computations. *Numerical Algorithms*, 65(4):759–782, 2014. DOI: 10.1007/s11075-013-9713-z.

[45] T. Gergelits, K.-A. Mardal, B. F. Nielsen, and Z. Strakoš. Laplacian preconditioning of elliptic PDEs: Localization of the eigenvalues of the discretized operator. *SIAM Journal on Numerical Analysis*, 57(3):1369–1394, 2019. DOI: 10.1137/18M1212458.

[46] T. Gergelits, B. F. Nielsen, and Z. Strakoš. Generalized spectrum of second order differential operators. *SIAM Journal on Numerical Analysis*, 58(4):2193–2211, 2020. DOI: 10.1137/20M1316159.

[47] R. G. Ghanem and P. D. Spanos. *Stochastic finite elements: A spectral approach*. Springer New York, 1991. DOI: 10.1007/978-1-4612-3094-6.

[48] L. Giraldi, A. Nouy, G. Legrain, and P. Cartraud. Tensor-based methods for numerical homogenization from high-resolution images. *Computer Methods in Applied Mechanics and Engineering*, 254:154–169, 2013. DOI: 10.1016/j.cma.2012.10.012.

[49] F. S. Göküzüm and M. A. Keip. An algorithmically consistent macroscopic tangent operator for FFT-based computational homogenization. *International Journal for Numerical Methods in Engineering*, 113(4):581–600, 2018. DOI: 10.1002/nme.5627.

[50] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 2013. ISBN 9781421407944.

[51] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra and Its Applications*, 261(1-3):1–21, 1997. DOI: 10.1016/S0024-3795(96)00301-1.

[52] D. Gottlieb and C.-W. Shu. On the Gibbs phenomenon and its resolution. *SIAM Review*, 39(4):644–668, 1997. DOI: 10.1137/S0036144596301390.

[53] W. Hackbusch. *Tensor spaces and numerical tensor calculus*. Springer Science and Business Media, Berlin, Heidelberg, 2012. ISBN 9783540773986. DOI: 10.1007/978-3-540-78862-1.

[54] W. Hackbusch. Numerical tensor calculus. *Acta numerica*, 23:651–742, 2014. DOI: 10.1017/S0962492914000087.

[55] R. W. Hamming. *Numerical methods for scientists and engineers*. International Series in Pure and Applied Mathematics. McGraw-Hill, 1962.

[56] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49:409–435, 1952. DOI: 10.6028/JRES.049.044.

[57] S. Margenov J. Kraus. *Robust algebraic multilevel methods and algorithms*. Walter de Gruyter, 2009.

[58] T. D. B. Jacobs, T. Junge, and L. Pastewka. Quantitative characterization of surface topography using spectral analysis. *Surface Topography: Metrology and Properties*, 5 (1):013001, 2017. DOI: 10.1088/2051-672x/aa51f8.

[59] C. Johnson. *Numerical solution of partial differential equations by the finite element method*. Cambridge University Press, 1995.

[60] T. Junge et al. µSpectre: An open-source platform for efficient FFT-based continuum mesoscale modelling., 2018. URL gitlab.com/muspectre/muspectre.

[61] M. Kabel, T. Böhlke, and M. Schneider. Efficient fixed point and Newton–Krylov solvers for FFT-based homogenization of elasticity at large deformations. *Computational Mechanics*, 54:1497–1514, 2014. DOI: 10.1007/s00466-014-1071-8.

[62] M. Kadic, G. W. Milton, M. van Hecke, and M. Wegener. 3D metamaterials. *Nature Reviews Physics*, 1(3):198–210, 2019. DOI: 10.1038/s42254-018-0018-y.

[63] S. Karaa. High-order difference schemes for 2D elliptic and parabolic problems with mixed derivatives. *Numerical Methods for Partial Differential Equations*, 23(2):366–378, 2007. DOI: 10.1002/num.20181.

[64] S. Kaßbohm, W. H. Müller, and R. Feßler. Improved approximations of Fourier coefficients for computing periodic structures with arbitrary stiffness distribution. *Computational Materials Science*, 37(1-2):90–93, 2006. DOI: 10.1016/j.commatsci.2005.12.010.

[65] C. G. Khatri and C. R. Rao. Solutions to some functional equations and their applications to characterization of probability distributions. *Sankhyā: The Indian Journal of Statistics, Series A*, 30(2):167–180, 1968. DOI: 10.2307/25049527.

[66] B. Khoromskij and S. Repin. Rank structured approximation method for quasi-periodic elliptic problems. *Computational Methods in Applied Mathematics*, 17(3):457–477, 2017. DOI: 10.1515/cmam-2017-0014.

[67] B. N. Khoromskij and S. I. Repin. A fast iteration method for solving elliptic problems with quasiperiodic coefficients. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 30(6):329–344, 2015. DOI: 10.1515/rnam-2015-0030.

[68] B. N. Khoromskij and Ch. Schwab. Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs. *SIAM Journal on Scientific Computing*, 33(1): 364–385, 2011. DOI: 10.1137/100785715.

[69] M. Khorrami, J. R. Mianroodi, P. Shanthraj, and B. Svendsen. Development and comparison of spectral algorithms for numerical modeling of the quasi-static mechanical behavior of inhomogeneous materials. *arXiv:2009.03762*, 2020.

[70] J. Kochmann, B. Svendsen, S. Reese, L. Ehle, S. Wulfinghoff, and J. Mayer. Efficient and accurate two-scale FE-FFT-based prediction of the effective material behavior of elasto-viscoplastic polycrystals. *Computational Mechanics*, 61(6):751–764, 2017. DOI: 10.1007/s00466-017-1476-2.

[71] J. Kochmann, K. Manjunatha, C. Gierden, S. Wulfinghoff, B. Svendsen, and S. Reese. A simple and flexible model order reduction method for FFT-based homogenization problems using a sparse sampling technique. *Computer Methods in Applied Mechanics and Engineering*, 347:622–638, 2019. DOI: 10.1016/j.cma.2018.11.032.

[72] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009. DOI: 10.1137/07070111X.

[73] D. Kressner and Ch. Tobler. Low-rank tensor krylov subspace methods for parametrized linear systems. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1288–1316, 2011. DOI: 10.1137/100799010.

[74] M. Kubínová and I. Pultarová. Block preconditioning of stochastic Galerkin problems: New two-sided guaranteed spectral bounds. *SIAM/ASA J. Uncertain. Quantification*, 8 (1):88–113, 2020. DOI: 10.1137/19M125902X.

[75] N. Lahellec, J. C. Michel, H. Moulinec, and P. Suquet. Analysis of inhomogeneous materials at large strains using fast Fourier transforms. In *IUTAM Symposium on Computational Mechanics of Solid Materials at Large Strains*, pages 247–258, Dordrecht, 2003. Springer Netherlands. ISBN 978-94-017-0297-3.

[76] H. P. Langtangen and S. Linge. *Finite difference computing with PDEs: A modern software approach*. Springer, 2017. ISBN 978-3319554556.

[77] R. A. Lebensohn and A. Needleman. Numerical implementation of non-local polycrystal plasticity using fast Fourier transforms. *Journal of the Mechanics and Physics of Solids*, 97(SI):333–351, 2016. DOI: 10.1016/j.jmps.2016.03.023.

[78] S. T. Lee, J. Liu, and H.-W. Sun. Combined compact difference scheme for linear second-order partial differential equations with mixed derivative. *Journal of Computational and Applied Mathematics*, 264:23–37, 2014. DOI: 10.1016/j.cam.2014.01.004.

[79] M. Leuschner and F. Fritzen. Fourier-Accelerated Nodal Solvers (FANS) for homogenization problems. *Computational Mechanics*, 62(3):359–392, 2018. DOI: 10.1007/s00466-017-1501-5.

[80] R. J. Leute, **M. Ladecký**, A. Falsafi, I. Jödicke, I. Pultarová, J. Zeman, T. Junge, and L. Pastewka. Elimination of ringing artifacts by finite-element projection in FFT-based homogenization. *Journal of Computational Physics*, 453:110931, 2022. DOI: 10.1016/j.jcp.2021.110931.

[81] R. J. LeVeque. *Finite difference methods for ordinary and partial differential equations: Steady-state and time-dependent problems*. Society for Industrial and Applied Mathematics, 2007. ISBN 9780898717839.

[82] J. Liesen and Z. Strakos. *Krylov subspace methods: Principles and analysis*. Oxford University Press, Oxford, 2012. ISBN 9780199655410. DOI: 10.1093/acprof:oso/9780199655410.001.0001.

[83] J. LLorca, C. González, J. M. Molina-Aldareguía, J. Segurado, R. Seltzer, F. Sket, M. Rodríguez, S. Sádaba, R. Muñoz, and L. P. Canal. Multiscale modeling of composite materials: A roadmap towards virtual testing. *Advanced Materials*, 23(44):5130–5147, 2011. DOI: 10.1002/adma.201101683.

[84] S. Lucarini, M. V. Upadhyay, and J. Segurado. FFT based approaches in micromechanics: Fundamentals, methods and applications. *Modelling and Simulation in Materials Science and Engineering*, 30(2):023002, 2021. DOI: 10.1088/1361-651x/ac34e1.

[85] S. Lucarini, L. Cobian, A. Voitus, and J. Segurado. Adaptation and validation of FFT methods for homogenization of lattice based materials. *Computer Methods in Applied Mechanics and Engineering*, 388:114223, 2022. DOI: 10.1016/j.cma.2021.114223.

[86] X. Ma, M. Shakoor, D. Vasiukov, S. V. Lomov, and Chung Hae Park. Numerical artifacts of fast Fourier transform solvers for elastic problems of multi-phase materials: Their causes and reduction methods. *Computational Mechanics*, 2021. DOI: 10.1007/s00466-021-02013-5.

[87] E. Maire and P. J. Withers. Quantitative X-ray tomography. *International Materials Reviews*, 59(1):1–43, 2014. DOI: 10.1179/1743280413Y.0000000023.

[88] B. B. Mandelbrot, W. H. Freeman, and Company. *The fractal geometry of nature*. Einaudi paperbacks. Henry Holt and Company, 1983. ISBN 9780716711865.

[89] B. Matérn. *Spatial variation*, volume 36. Springer New York, New York, NY, 1986. ISBN 978-0-387-96365-5. DOI: 10.1007/978-1-4615-7892-5.

[90] K. Matouš, M. G. D. Geers, V. G. Kouznetsova, and A. Gillman. A review of predictive nonlinear theories for multiscale modeling of heterogeneous materials. *Journal of Computational Physics*, 330:192–220, 2017. DOI: 10.1016/j.jcp.2016.10.070.

[91] H. G. Matthies and E. Zander. Solving stochastic systems with low-rank tensor compression. *Linear Algebra and its Applications*, 436(10):3819–3838, 2012. DOI: 10.1016/j.laa.2011.04.017.

[92] C. Meng, W. Heltsley, and D. D. Pollard. Evaluation of the Eshelby solution for the ellipsoidal inclusion and heterogeneity. *Computational Geoscience*, 40:40–48, 2012. DOI: 10.1016/j.cageo.2011.07.008.

[93] G. Meurant and Z. Strakoš. The Lanczos and conjugate gradient algorithms in finite precision arithmetic. *Acta Numerica*, 15:471–542, 2006. DOI: 10.1017/S096249290626001X.

[94] G. Meurant and P. Tichy. On computing quadrature-based bounds for the A-norm of the error in conjugate gradients. *Numerical Algorithms*, 62(2):163–191, 2013.

[95] J.-C. Michel, H. Moulinec, and P. Suquet. Effective properties of composite materials with periodic microstructure: A computational approach. *Computer Methods in Applied Mechanics and Engineering*, 172(1–4):109–143, 1999. DOI: 10.1016/S0045-7825(98)00227-8.

[96] G. W. Milton. *The theory of composites*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge, UK, 2002.

[97] G. W. Milton and R. V. Kohn. Variational bounds on the effective moduli of anisotropic composites. *Journal of the Mechanics and Physics of Solids*, 36(6):597–629, 1988. DOI: 10.1016/0022-5096(88)90001-4.

[98] N. Mishra, J. Vondřejc, and J. Zeman. A comparative study on low-memory iterative solvers for FFT-based homogenization of periodic media. *Journal of Computational Physics*, 321:151–168, 2016. DOI: 10.1016/j.jcp.2016.05.041.

[99] **M. Ladecký**, I. Pultarová, and J. Zeman. Guaranteed two-sided bounds on all eigenvalues of preconditioned diffusion and elasticity problems solved by the finite element method. *Applications of Mathematics*, 66(1):21–42, 2021. DOI: 10.21136/AM.2020.0217-19.

[100] **M. Ladecký**, A. Falsafi, R. J. Leute, I. Pultarová, J. Zeman, T. Junge, and L. Pastewka. On the equivalence of the displacement- and strain-based computational homogenisation schemes. *in preparation*, 2022.

[101] **M. Ladecký**, J. R. Leute, A. Falsafi, I. Pultarová, L. Pastewka, T. Junge, and J. Zeman. Optimal FFT-accelerated finite element solver for homogenisation. 2022. DOI: 10.48550/arXiv.2203.02962.

[102] H. Moulinec and P. Suquet. A fast numerical method for computing the linear and nonlinear mechanical properties of composites. *Comptes Rendus de l'Académie des sciences. Série II. Mécanique, physique, chimie, astronomie*, 318(1–2):1417–1423, 1994.

[103] H. Moulinec and P. Suquet. A numerical method for computing the overall response of nonlinear composites with complex microstructure. *Computer Methods in Applied Mechanics and Engineering*, 157(1–2):69–94, 1998. DOI: 10.1016/S0045-7825(97)00218-1.

[104] H. Moulinec, P. Suquet, and G. W. Milton. Convergence of iterative methods based on Neumann series for composite materials: Theory and practice. *International Journal for Numerical Methods in Engineering*, 114(10):1103–1130, 2018. DOI: 10.1002/nme.5777.

[105] T. Mura. *Micromechanics of defects in solids*. Kluwer Academic Publishers Group, 1982. ISBN 978-94-011-8548-6. DOI: 10.1007/978-94-011-9306-1.

[106] J. Málek and Z. Strakoš. *Preconditioning and the conjugate gradient method in the context of solving PDEs*. SIAM Spotlight Series. Society for Industrial and Applied Mathematics, 2015. ISBN 978-1-611973-83-9.

[107] W. H. Müller. Mathematical vs. experimental stress analysis of inhomogeneities in solids. *J. Phys. IV France*, 06(C1):139–148, 1996. DOI: 10.1051/jp4:1996114.

[108] M. A. Najafgholipour, S. M. Dehghan, A. Dooshabi, and A. Niroomandi. Finite element analysis of reinforced concrete beam-column connections with governing joint shear failure mode. *Latin American Journal of Solids and Structures*, 14:1200–1225, 2017. DOI: 10.1590/1679-78253682.

[109] J. Nečas and I. Hlaváček. *Mathematical theory of elastic and elasto-plastic bodies: An introduction*. Elsevier, Amsterdam, 1981.

[110] B. F. Nielsen, A. Tveito, and W. Hackbusch. Preconditioning by inverting the Laplacian: An analysis of the eigenvalues. *IMA Journal of Numerical Analysis*, 29(1):24–42, 2009. DOI: 10.1093/imanum/drm018.

[111] A. Nouy. *Low-rank methods for high-dimensional approximation and model order reduction*. Society for Industrial and Applied Mathematics, 2015. DOI: 10.1007/978-3-319-11259-6_21-1.

[112] A. Nouy. *Low-Rank Tensor Methods for Model Order Reduction*. Springer International Publishing, Cham, 2015. DOI: 10.1007/978-3-319-11259-6_21-1.

[113] A. Osama, Al-A. Abdulrahman, A. Wadea, and H. M. Syed. Additive manufacturing: Challenges, trends, and applications. *Advances in Mechanical Engineering*, 11(2), 2019. DOI: 10.1177/1687814018822880.

[114] I. Oseledets and E. Tyrtyshnikov. TT-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*, 432(1):70–88, 2010. DOI: 10.1016/J.LAA.2009.07.024.

[115] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011. DOI: 10.1137/090752286.

[116] I. Pultarová. Hierarchical preconditioning for the stochastic Galerkin method: Upper bounds to the strengthened CBS constants. *Computers and Mathematics with Applications*, 71(4):949–964, 2016. DOI: 10.1016/j.camwa.2016.01.006.

[117] I. Pultarová. Block and multilevel preconditioning for stopchastic Galerkin problems with lognormally distributed parameters and tensor product polynomials. *International Journal for Uncertainty Quantification*, 7(5):441–462, 2017.

[118] I. Pultarová and **M. Ladecký**. Two-sided guaranteed bounds to individual eigenvalues of preconditioned finite element and finite difference problems. *Numerical Linear Algebra with Applications*, 28(5):e2382, 2021. DOI: 10.1002/nla.2382.

[119] S. B. Ramisetti, C. Campañá, G. Anciaux, J.-F. Molinari, M. H. Müser, and M. O. Robbins. The autocorrelation function for island areas on self-affine surfaces. *Journal of Physics: Condensed Matter*, 23(21):215004, 2011. DOI: 10.1088/0953-8984/23/21/215004.

[120] C. Ramos, A. Isabel, C. Roux-Langlois, C. F. Dunant, M. Corrado, and J.-F. Molinari. HPC simulations of alkali-silica reaction-induced damage: Influence of alkali-silica gel properties. *Cement Concrete Research*, 109:90–102, 2018. DOI: 10.1016/j.cemconres.2018.03.020.

[121] P. Rauwoens, J. Vierendeels, and B. Merci. A solution for the odd–even decoupling problem in pressure-correction algorithms for variable density flows. *Journal of Computational Physics*, 227(1):79–99, 2007. DOI: 10.1016/j.jcp.2007.07.010.

[122] F. Roters, M. Diehl, P. Shanthraj, P. Eisenlohr, C. Reuber, S. L. Wong, T. Maiti, A. Ebrahimi, T. Hochrainer, H. O. Fabritius, S. Nikolov, M. Friák, N. Fujita, N. Grilli, K. G. F. Janssens, N. Jia, P. J. J. Kok, D. Ma, F. Meier, E. Werner, M. Stricker, D. Weygand, and D. Raabe. DAMASK – the Düsseldorf Advanced Material Simulation Kit for modeling multi-physics crystal plasticity, thermal, and damage phenomena from the single crystal up to the component scale. *Computational Materials Science*, 158: 420–478, 2019. DOI: 10.1016/j.commatsci.2018.04.030.

[123] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, second edition, 2003. ISBN 978-0-898715-34-7. DOI: 10.1137/1.9780898718003.

[124] A. A. Samarskii, P. P. Matus, V. I. Mazhukin, and I. E. Mozolevski. Monotone difference schemes for equations with mixed derivatives. *Computers and Mathematics with Applications*, 44(3):501–510, 2002. DOI: 10.1016/S0898-1221(02)00164-5.

[125] J. Saranen and G. Vainikko. *Periodic integral and pseudodifferential equations with numerical approximation*. Springer Monographs Mathematics, Berlin, Heidelberg, 2002.

[126] M. Schneider. Convergence of FFT-based homogenization for strongly heterogeneous media. *Mathematical Methods in the Applied Sciences*, 38(13):2761–2778, 2015. DOI: 10.1002/mma.3259.

[127] M. Schneider. An FFT-based fast gradient method for elastic and inelastic unit cell homogenization problems. *Computer Methods in Applied Mechanics and Engineering*, 315:846–866, 2017. DOI: 10.1016/j.cma.2016.11.004.

[128] M. Schneider. Lippmann-Schwinger solvers for the computational homogenization of materials with pores. *International Journal for Numerical Methods in Engineering*, 121 (22):5017–5041, 2020. DOI: 10.1002/nme.6508.

[129] M. Schneider. A review of nonlinear FFT-based computational homogenization methods. *Acta Mechanica*, 232(6):2051–2100, 2021. DOI: 10.1007/s00707-021-02962-1.

[130] M. Schneider, F. Ospald, and M. Kabel. Computational homogenization of elasticity on a staggered grid. *International Journal for Numerical Methods in Engineering*, 105 (9):693–720, 2016. DOI: 10.1002/nme.5008.

[131] M. Schneider, D. Merkert, and M. Kabel. FFT-based homogenization for microstructures discretized by linear hexahedral elements. *International Journal for Numerical Methods in Engineering*, 109(10):1461–1489, 2017. DOI: 10.1002/nme.5336.

[132] J. Segurado, R. A. Lebensohn, and J. Llorca. Computational homogenization of polycrystals. *Advances in Applied Mechanics*, 51:1–114, 2018. DOI: 10.1016/bs.aams.2018.07.001.

[133] D. Serre. *Matrices: Theory and applications*. Springer, 2010.

[134] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996. DOI: 10.1073/pnas.93.4.1591.

[135] P. Shanthraj, P. Eisenlohr, M. Diehl, and F. Roters. Numerically robust spectral methods for crystal plasticity simulations of heterogeneous materials. *International Journal of Plasticity*, 66:31–45, 2015. DOI: 10.1016/j.ijplas.2014.02.006.

[136] A. van der Sluis and H. A. van der Vorst. The rate of convergence of conjugate gradients. *Numerische Mathematik*, 48:543–560, 1986.

[137] B. Sonon, B. Francois, and T. J. Massart. A unified level set based methodology for fast generation of complex microstructural multi-phase RVEs. *Comput. Method. Appl. M.*, 223:103–122, 2012. DOI: 10.1016/j.cma.2012.02.018.

[138] B. Sonon, K. Ehab Moustafa Kamel, and T. J. Massart. *Advanced geometry representations and tools for microstructural and multiscale modeling*, volume 54. Elsevier, 2021. DOI: 10.1016/bs.aams.2020.12.001.

[139] B. Sousedik and R. Ghanem. Truncated hierarchical preconditioning for the stochastic Galerkin FEM. *International Journal for Uncertainty Quantification*, 4(4):333–348, 2014.

[140] Z. Strakoš. On the real convergence rate of the conjugate gradient method. *Linear Algebra and its Applications*, 154-156:535–549, 1991. DOI: 10.1016/0024-3795(91)90393-B.

[141] K. Terada, T. Miura, and N. Kikuchi. Digital image-based modeling applied to the homogenization analysis of composite materials. *Computational Mechanics*, 20:331–346, 1997. DOI: 10.1007/s004660050255.

[142] Ch. Tobler. *Low-rank tensor methods for linear systems and eigenvalue problems*. PhD thesis, ETH Zürich, 2012.

[143] H. A. van der Vorst. *Iterative Krylov methods for large linear systems*. Cambridge University Press, 2003.

[144] B. van Es, B. Koren, and H. J. de Blank. Finite-difference schemes for anisotropic diffusion. *Journal of Computational Physics*, 272:526–549, 2014. DOI: 10.1016/j.jcp.2014.04.046.

[145] T. Vejchodský. Three methods for two-sided bounds of eigenvalues—A comparison. *Numerical Methods for Partial Differential Equations*, 34(4):1188–1208, 2018. DOI: 10.1002/num.22251.

[146] A. Vidyasagar, W. L. Tan, and D. M. Kochmann. Predicting the effective response of bulk polycrystalline ferroelectric ceramics via improved spectral phase field methods. *Journal of the Mechanics and Physics of Solids*, 106:133–151, 2017. DOI: 10.1016/j.jmps.2017.05.017.

[147] J. Vondřejc. Improved guaranteed computable bounds on homogenized properties of periodic media by the Fourier–Galerkin method with exact integration. *International Journal for Numerical Methods in Engineering*, 107(13):1106–1135, 2016. DOI: 10.1002/nme.5199.

[148] J. Vondřejc. Double-grid quadrature with interpolation-projection (DoGIP) as a novel discretisation approach: An application to FEM on simplexes. *Computers and Mathematics with Applications*, 78(11):3501–3513, 2019. DOI: 10.1016/j.camwa.2019.05.021.

[149] J. Vondřejc and T. W. J. de Geus. Energy-based comparison between the Fourier–Galerkin method and the finite element method. *Journal of Computational and Applied Mathematics*, 374:112585, 2020. DOI: 10.1016/j.cam.2019.112585.

[150] J. Vondřejc, J. Zeman, and I. Marek. An FFT-based Galerkin method for homogenization of periodic media. *Computers and Mathematics with Applications*, 68(3):156–173, 2014. DOI: 10.1016/j.camwa.2014.05.014.

[151] J. Vondřejc, J. Zeman, and I. Marek. Guaranteed upper-lower bounds on homogenized properties by FFT-based Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 297:258–291, 2015. DOI: 10.1016/j.cma.2015.09.003.

[152] J. Vondřejc, D. Liu, **M. Ladecký**, and H. G. Matthies. FFT-based homogenisation accelerated by low-rank tensor approximations. *Computer Methods in Applied Mechanics and Engineering*, 364:112890, 2020. DOI: 10.1016/j.cma.2020.112890.

[153] J. Vondřejc et al. FFTHomPy: Numerical software for evaluating guaranteed upper-lower bounds on homogenized properties., 2020. URL github.com/vondrejc/FFTHomPy.

[154] F. Willot. Fourier-based schemes for computing the mechanical response of composites with accurate local fields. *C. R. Mécanique*, 343(3):232–245, 2015. DOI: 10.1016/j.crme.2014.12.005.

[155] F. Willot, B. Abdallah, and Y.-P. Pellegrini. Fourier-based schemes with modified Green operator for computing the electrical response of heterogeneous media with accurate local fields. *International Journal for Numerical Methods in Engineering*, 98(7):518–533, 2014. DOI: 10.1002/nme.4641.

[156] J. Wu, O. Sigmund, and J. P. Groen. Topology optimization of multi-scale structures: A review. *Structural and Multidisciplinary Optimization*, 63(3):1455–1480, 2021. DOI: 10.1007/s00158-021-02881-8.

[157] D. Xiu. *Numerical methods for stochastic computations: A spectral method approach*. Princeton University Press, 2010. ISBN 9780691142128.

[158] J. Zeman, J. Vondřejc, J. Novák, and I. Marek. Accelerating a FFT-based solver for numerical homogenization of periodic media by conjugate gradients. *Journal of Computational Physics*, 229(21):8065–8071, 2010. DOI: 10.1016/j.jcp.2010.07.010.

[159] J. Zeman, T. W. J. de Geus, J. Vondřejc, R. H. J. Peerlings, and M. G. D. Geers. A finite element perspective on nonlinear FFT-based micromechanical simulations. *International Journal for Numerical Methods in Engineering*, 111(10):903–926, 2017. DOI: 10.1002/nme.5481.

[160] I. Šebestová and T. Vejchodský. Two-sided bounds for eigenvalues of differential operators with applications to Friedrichs, Poincaré, trace, and similar constants. *SIAM Journal on Numerical Analysis*, 52(1):308–329, 2014. DOI: 10.1137/13091467X.