FFT-based homogenisation accelerated by low-rank approximations

Jaroslav Vondřejc¹, Dishi Liu¹, Martin Ladecký², and Hermann G. Matthies³

¹Technische Universität Braunschweig, Institute of Scientific Computing, Mühlenpfordstrasse 23, 38106 Braunschweig, Germany, j.vondrejc@tu-bs.de
²Czech Technical University in Prague, Faculty of Civil Engineering, Thákurova 7/2077 166 29 Prague 6, Czech Republic

February 21, 2019

Abstract

Fast Fourier transform (FFT) based methods has turned out to be an effective computational approach for numerical homogenisation. Particularly, Fourier-Galerkin methods are computational methods for partial differential equations that are discretised with trigonometric polynomials. Its computational effectiveness benefits from efficient FFT based algorithms as well as a favourable condition number. Here this kind of methods are accelerated by low-rank tensor approximation techniques for a solution field using canonical, 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 setting 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.

Keywords: Fourier-Galerkin method, fast Fourier transform, low-rank approximations, reduced order modelling, homogenisation

Contents

1 Introduction

2 Homogenisation by Fourier-Galerkin methods

2.1 Model problem

2.2 Fourier-Galerkin methods

2.2.1 Trigonometric polynomials

2.2.2 Fourier-Galerkin method with numerical integration (FFTH-GaNi)

2.2.3 Fourier-Galerkin method with exact integration (FFTH-Ga)

2.3 Preconditioning
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 [1]. The method that has application in multiscale problems represents an alternative discretisation approach to 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 function for an auxiliary homogeneous problem. Its connection to standard variational formulation has been discovered in [2] by using the fact that the Green function is a projection on compatible fields (i.e. gradient fields in scalar elliptic problems), see [3]. It has allowed to fully remove reference conductivity tensor from the formulation and interpreted the method from the perspective of finite elements also in nonlinear problems [4, 5]. Moreover the standard primal-dual
variational formulations allow to compute guaranteed bounds on effective material properties [6] which provides tighter bounds than the Hashin-Shtrikman functional.

A 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 [7] and followed by [6,8,4,9]. Other discretisation approaches are based on pixel-wise constant basis functions [10,11], linear hexahedral elements [12], or finite differences [13,14]. The variational formulations also allowed to derive convergence of approximate solutions to the continuous one [2,9,10].

The various discretisation approaches has been studied along with the linear and non-linear solvers [15,16,7,17,11,18,4,5,19]. Other research directions focus e.g. on multiscale methods [20,21,22] or highly non-linear problems in solid mechanics [23,24,25,26].

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 two-dimensional tensors, the optimal low-rank approximation in mean square sense is based on truncated singular value decomposition (SVD). A computationally cheaper choice is Cross Approximation [27,28] which has only linear complexity in matrix size $N$. Low-rank formats in higher dimensions include canonical, Tucker, and hierarchical schemes such as tensor train and quantic tensor-train [29,30]. Nevertheless, it is preferable to avoid decompositions of full tensor, which corresponds e.g. to a solution of a discretised PDE. In [31] an efficient Proper Generalised Decomposition is adopted for the construction of low rank tensors in canonical and Tucker formats in a numerical homogenisation from high-resolution images. It is possible to compute the tensors directly in low-rank formats, which can be provided by a suitable solver [32,33,34,35]. 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 require to deal with operations such as addition, element-wise multiplication, or Fourier transformation. Since the tensors are described with fewer parameters the computational complexities are typically reduced which leads to significant speed up of computations. However, during operations with tensors in low-rank format it may happen that the representation rank of the tensors grows which calls for their truncation, i.e. their approximation or reparametrisation with fewer parameters such that the accuracy is kept [36,37,38]. This is in fact the rounding of tensors which is a generalisation to rounding of numbers occurring when working with floats. The applications of low-rank approximations are very broad e.g. for stochastic problems in higher dimensions [39,40,11,22], acceleration of solutions to PDEs, or model order reduction [43] but its application to numerical homogenisation is new.

Structure of the paper. In section 2 the state-of-the-art of Fourier-Galerkin method is described for a model homogenisation problem of scalar elliptic equation. Particularly the two discretisation methods based on numerical and exact integration are described along with their corresponding linear systems. Then in section 3 the low-rank approximation techniques are summarised and their application within Fourier-Galerkin method is discussed. In section 4 the effectiveness of the low-rank approximation is demonstrated on several numerical examples.

Notation. We will denote $d$-dimensional vectors and matrices by boldface letters: $a = (a_i)_{i=1,2,...,d} \in \mathbb{R}^d$ or $A = (A_{ij})_{i,j=1}^d \in \mathbb{R}^{d \times d}$. Matrix-matrix and matrix-vector multiplications are denoted as $C = AB$ and $c = Ab$, 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 $c = a \cdot b = \sum_i a_i b_i$, and the induced norm as $\|a\| = \sqrt{a \cdot a}$. Vectors, matrices, and tensors such as $x, b,$ and $A$ arising from discretisation will be denoted by the bold serif font in order to highlight their special structure. The compo-
ponents of a tensor will be denoted as $A = (A[k_1, \ldots, k_d])_{k_1,\ldots,k_d=1}^N \in \mathbb{R}^N$ for $N = (N, \ldots, N) \in \mathbb{N}^d$; the multiindex notation will be also incorporated to simplify the components of the tensors, e.g., $A[k] = A[k]$ for index $k = [k_1, \ldots, k_d]$.

The space of square integrable $\mathcal{Y}$-periodic integrable 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 $v : \mathcal{Y} \to \mathbb{R}^d$ with components $v_i$ from $L^2(\mathcal{Y})$. Finally, $H^1_0(\mathcal{Y}) = \{ v \in L^2(\mathcal{Y}) | \nabla v \in L^2(\mathcal{Y}; \mathbb{R}^d), \int_\mathcal{Y} v(x) \, dx = 0 \}$ denotes the Sobolev space of periodic functions with zero mean.

2 Homogenisation by Fourier-Galerkin methods

2.1 Model problem

A model problem in homogenisation [11] consists of a scalar linear elliptic variational problem defined on a unit domain $\mathcal{Y} = (-\frac{1}{2}, \frac{1}{2})^d$ in $d$-dimensional setting (considering both $d=2$ and $d=3$) with material coefficients $A : \mathcal{Y} \to \mathbb{R}^{d \times d}$ which are required to be essentially bounded, symmetric, and uniformly elliptic. This means that for almost all $x \in \mathcal{Y}$, there are constants $0 < c_A \leq C_A < +\infty$ such that

$$A(x) = A^T(x), \quad c_A \|v\|^2 \leq A(x)v \cdot v \leq C_A \|v\|^2 \quad \text{for all } v \in \mathbb{R}^d. \quad (1)$$

The homogenisation problem is focused on the computation of effective material properties $A_H \in \mathbb{R}^{d \times d}$. Its variational formulation is based on the minimisation of a microscopic energetic functional for constant vectors $E \in \mathbb{R}^d$ as

$$A_H E \cdot E = \min_{v \in H^1_0(\mathcal{Y})} a(E + \nabla v, E + \nabla v) \quad (2)$$

where $\nabla v$ is the periodically fluctuating microscopic field of $E$ and 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

$$a(e, w) := \int_\mathcal{Y} A(x) e(x) \cdot w(x) \, dx. \quad (3)$$

2.2 Fourier-Galerkin methods

Alternatively the minimisers in (2) are described by a weak formulation: find $u \in H^1_0(\mathcal{Y})$ such that

$$a(\nabla u, \nabla v) = -a(E, \nabla v) \quad \forall v \in H^1_0(\mathcal{Y}). \quad (4)$$

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

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

$$A_H E \cdot E = A_{H,11} =: A_H.$$
2.2.1 Trigonometric polynomials

The Fourier-Galerkin method, [45, 2, 8] is built on the discretisation using the space of trigonometric polynomials

\[ T_N = \left\{ \sum_{k \in \mathbb{Z}} \hat{v}[k] \varphi_k \mid \hat{v}[k] \in \mathbb{C} \text{ and } \hat{v}[k] = \hat{v}[-k] \right\}, \]

where \( \varphi_k(x) = \exp(2\pi i k \cdot x) \) are well known Fourier basis functions. The number of discretisation points \( N = [N, \ldots, N] \in \mathbb{R}^d \) in this work takes only odd values because an even \( N \) introduces Nyquist frequencies that have to be omitted to obtain a conforming approximation, see [6] for details.

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

\[ \varphi_N^k(x) = \frac{1}{|N|} \sum_{m \in \mathbb{Z}} \omega_N^{-km} \varphi^m(x) \text{ for } x \in \mathcal{Y}, \]

of Fourier basis \( \varphi^m \) with the complex-valued weights \( \omega_N^{-km} = \exp(2\pi i \sum_{\alpha=1}^{d} \frac{m_{\alpha}k_{\alpha}}{N_{\alpha}}) \) for \( m, k \in \mathbb{Z}_N \).

The weights are from discrete Fourier transform (DFT) matrices in \( \mathbb{C}^{N \times N} \) with components

\[ \mathcal{F}_N[m, k] = \frac{1}{|N|} \omega_N^{-mk}, \quad \mathcal{F}_N^{-1}[m, k] = \omega_N^{mk} \text{ for } m, k \in \mathbb{Z}_N. \]

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

\[ v(x) = \sum_{k \in \mathbb{Z}_N} \hat{v}[k] \varphi_k(x) = \sum_{k \in \mathbb{Z}_N} v[k] \varphi_N^k(x) \quad \text{and} \quad \hat{v} = \mathcal{F}_N v. \]

Thanks to the Dirac delta property \( \varphi_N^l(x_N^k) = \delta_{kl} \) of the fundamental trigonometric polynomials on a regular grid of points \( x_N^k = \frac{k}{N} \) for \( k, l \in \mathbb{Z}_N \), the coefficients of the trigonometric polynomials are equal to the functional values on the grid points, i.e. \( v[k] = v(x_N^k) \).

Differential operators are naturally applied on trigonometric polynomials. Particularly, the gradient

\[ \nabla v(x) = \sum_{k \in \mathbb{Z}_N} \hat{v}[k] \nabla \varphi_k(x) = \sum_{k \in \mathbb{Z}_N} 2\pi i k \hat{v}[k] \varphi_k(x), \]

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

\[ (\hat{\nabla}_N^* \hat{w})[k] = \sum_{\alpha=1}^{d} -2\pi i k_\alpha \hat{w}[\alpha, k]. \]

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

\[ \nabla v(x) = \sum_k (\mathcal{F}_N^{-1} \hat{\nabla}_N \mathcal{F}_N v)[k] \varphi_N^k(x) \]

(5)
where the $d$-fold discrete Fourier transform (emphasises with bold) $\mathcal{F}_N : \mathbb{C}^{d \times N} \rightarrow \mathbb{C}^{d \times N}$ acts individually on each component of the vector field $(\mathcal{F}_N w)[\alpha] = \mathcal{F}_N w[\alpha]$ for $\alpha = 1, \cdots, d$.

The numerical treatment of the weak formulation (4) or a corresponding Galerkin approximation requires to use numerical integrations. In this manuscript we incorporate two versions: an exact integration [8] as described in sub-section 2.2.3 and a numerical integration as described in sub-section 2.2.2.

### 2.2.2 Fourier-Galerkin method with numerical integration (FFTH-GaNi)

This numerical integration corresponds to the original Moulinec-Suquet algorithm as the resulting discrete solution vectors fully coincide. This approach applied on the bilinear form (3) on regular grids reads

$$a(e, w) \approx a_N(e_N, w_N) = \sum_{k \in \mathbb{Z}} A(x_N^k) e_N(x_N^k) \cdot w_N(x_N^k) = (\tilde{A} e, w)_{\mathbb{R}^{d \times N}},$$

where $e$ and $w$ store the values on the grid (e.g. $e[\alpha, k] = e_N, \alpha(x_N^k)$) and $\tilde{A} \in \mathbb{R}^{d \times d \times N \times N}$ is a block diagonal tensor with components

$$\tilde{A}[\alpha, \beta, k, l] = \delta_{kl} A_{\alpha \beta}(x_N^k);$$

in reality, one needs to store only the diagonals which can be done in the multidimensional array of shape $d \times d \times N$.

The integration leads to an approximate formulation of (4):

$$\text{find } u \in \mathcal{F}_N : a_N(\nabla u_N, \nabla v_N) = -a_N(E, \nabla v_N), \quad \forall v_N \in \mathcal{F}_N;$$

note the approximation is exact for a constant material coefficients $A$. This formulation that can be seen also as a collocation method [7] is equivalent to the original Moulinec and Suquet formulation [1] in the sense that the solution vectors coincide [2]. However, the formulation here builds on the variational formulation [2] solved for the potential field (instead of gradient one) with a preconditioner according to [46].

The combination of the numerical integration and the differentiation of trigonometric polynomials [5] allows to approximate the bilinear form with respect to the nodal values of potential fields

$$a_N(\nabla u_N, \nabla v_N) = (\tilde{A} \mathcal{F}_N^{-1} \nabla_N \mathcal{F}_N u, \mathcal{F}_N^{-1} \nabla_N \mathcal{F}_N v)_{\mathbb{R}^{d \times N}}.$$

In order to deduce the linear system, all operators at test vectors $v_N$ are moved to trial vector $u_N$ as adjoint operators to reveal the linear system in original space

$$\mathcal{F}_N^{-1} \nabla^* \nabla_N \mathcal{F}_N \tilde{A} \mathcal{F}_N^{-1} \nabla_N \mathcal{F}_N u = -\mathcal{F}_N^{-1} \nabla^* \nabla_N \mathcal{F}_N \tilde{A} E,$$

where $E \in \mathbb{R}^{d \times N}$ is a constant with components $E[\alpha, k] = E_\alpha$. We can 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

$$\nabla^*_N \mathcal{F}_N \tilde{A} \mathcal{F}_N^{-1} \nabla_N \tilde{u} = -\nabla^*_N \mathcal{F}_N \tilde{A} E.$$

(6)
2.2.3 Fourier-Galerkin method with exact integration (FFTH-Ga)

For many types of material coefficients \([1]\) and basis functions, there is a possibility to integrate the bilinear forms in weak formulation exactly, which leads to Galerkin approximation with exact integration

\[
\text{find } u \in \mathcal{T}_N : \quad a(\nabla u_N, \nabla v_N) = a(E, \nabla v_N) \quad \forall v_N \in \mathcal{T}_N.
\]

However, the direct integration of the Fourier-Galerkin formulation, contrary to FEM, leads to a full-linear system, which can be overcome with a double-grid integration with projection (DoGIP) \([6,8]\). The DoGIP is a general method applicable also within finite element method \([47]\). The original evaluation of material law on a grid \(N\) is reformulated as an evaluation on a double grid \(2N - 1\) with modified material coefficients; they can be expressed as a modification of original material coefficients.

The main idea relies on expressing gradients of the trial and a test function together

\[
\nabla u_N(x) \otimes \nabla v_N(x) = e_N(x) \otimes w_N(x) = \sum_{k \in \mathbb{Z}_{2N-1}} e[; k] \otimes w[; k] \varphi^k_{2N-1}(x)
\]

with respect to the basis of the double grid space consisting of trigonometric polynomials with doubled frequencies \(\mathcal{T}_{2N-1}\); the arrays \(e\) and \(w\) stores the values of polynomials on a double grid, e.g. \(e[\alpha, k] = e_N(\alpha x_{2N-1}^k)\) for \(\alpha \in \{1, \ldots, d\}\) and \(k \in \mathbb{Z}_{2N-1}\). Then the bilinear form can be expressed on a double grid

\[
a(e_N, w_N) = \sum_{k \in \mathbb{Z}_{2N-1}} \int_Y A(x) \varphi^k_{2N-1}(x) \, dx : e_{2N-1}[; k] \otimes w_{2N-1}[; k] = (A e, w)_{\mathbb{R}^{d \times (2N-1)}}
\]

where : is a double contraction between two matrices of size \(d \times d\) and the material coefficients are defined as

\[
A[\alpha, \beta, k, l] = \delta_{kl} \int_Y A_{\alpha\beta}(x) \varphi^k_{2N-1}(x) \, dx \quad \text{for } \alpha, \beta \in \{1, \ldots, d\} \text{ and } k, l \in \mathbb{Z}_{2N-1}.
\]

This integration is allowed exactly for a large class of material coefficients. Particularly in \([3,6]\), the 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 \(\mathcal{T}_N \subset \mathcal{T}_M\) for \(N < M\) (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 operator \(I : \mathbb{C}^{d \times N} \to \mathbb{C}^{d \times (2N-1)}\) defined as

\[
(I \tilde{w})[; k] = \begin{cases} 
\tilde{w}[; k], & \text{for } k \in \mathbb{Z}_N \\
0, & \text{for } k \in \mathbb{Z}_{2N-1} \setminus \mathbb{Z}_N.
\end{cases}
\]

Its adjoint operator \(I^* : \mathbb{C}^{d \times (2N-1)} \to \mathbb{C}^{d \times N}\) just removes the frequencies \(k \in \mathbb{Z}_{2N-1} \setminus \mathbb{Z}_N\).

This allows us to deduce the linear system with exact integration

\[
\hat{\nabla}_N I^* \mathcal{F}_{2N-1} A \mathcal{F}^{-1}_{2N-1} \hat{\nabla}_N \hat{u} = -\hat{\nabla}_N I^* \mathcal{F}_{2N-1} A E,
\]

which has very similar structure compared to the scheme based on numerical integration \([6]\).
2.3 Preconditioning

Following the recent paper [46], the preconditioning of both linear systems (6) and (7) is based on a Laplacian expressed in a Fourier domain

\[ \hat{P}[k, l] = \delta_{kl} k \cdot k, \]

which is a simple diagonal preconditioner. Its inverse is given by the Moore-Penrose pseudoinverse

\[ \hat{P}^{-1}[k, l] = \delta_{kl} \frac{1}{k} \text{ for } k \in \mathbb{Z}_N \setminus \{0\} \text{ and } \hat{P}^{-1}[0, 0] = 0; \]

the latter condition enforces the zero-mean property of the approximated vectors. The preconditioned systems are explicitly stated for both discretisation schemes

\[ \hat{P}^{-1} \hat{\nabla}_N^* \mathcal{F}_N \hat{A} \mathcal{F}^{-1}_N \hat{\nabla}_N \hat{u} = -\hat{P}^{-1} \hat{\nabla}_N^* \mathcal{F}_N \hat{A} \mathcal{E}, \] (8a)

\[ \hat{P}^{-1} \hat{\nabla}_N^* I_2^* \mathcal{F}_2 \mathcal{F}^{-1}_2 \hat{A} \mathcal{F}^{-1}_2 \mathcal{F}^* \hat{\nabla}_N \hat{u} = -\hat{P}^{-1} \hat{\nabla}_N^* I_2^* \mathcal{F}_2 \mathcal{E}. \] (8b)

3 FFT-based methods with low-rank approximations

Applying low rank approximation techniques is of particular interest for problems with huge number of degrees of freedom. The approximations could 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. The latter needs some operations such as additions, element-wise multiplication, and fast Fourier transform(FFT), to be implemented on tensors in low rank format. In this section we introduce an FFT-based solution process incorporated with low rank representations and operations of tensor. In the following sub-section 3.1, the low-rank approximation formats are summarised along the corresponding operations; the details can be found in textbooks or in appendix A. Then the application of low-rank approximation for Fourier-Galerkin method is described and discussed in sub-section 3.2 along with the possible linear solvers in sub-section 3.3.

3.1 Overview of low-rank formats

Here we give a brief introduction of three types of low rank tensor that is applied in this work, they are of canonical, Tucker, and tensor train format respectively. The canonical format is only used in 2-dimensional case for its intrinsic difficulty in finding optimal approximation in higher dimensionality. 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 A.

3.1.1 Canonical format

A canonical (or canonocal polyadic(CP), or \( r \)-term) approximation a tensor \( \mathbf{v} \in \mathbb{K}^{N_1 \times \cdots \times N_d} \) (field \( \mathbb{K} \) is \( \mathbb{R} \) or \( \mathbb{C} \)) is a sum of \( r \) rank-1 tensors. In this work the canonical format is only used for tensors with order 2 \( (d = 2) \), in this case the representation has the form:

\[ \mathbf{v} \approx \tilde{\mathbf{v}} = \sum_{i=1}^{r} c[i] \mathbf{b}^{(1)}[i] \otimes \mathbf{b}^{(2)}[i], \]

where \( c \in \mathbb{R}^r \) stores the coefficients with respect to basis vectors \( \mathbf{b}^{(j)} \in \mathbb{K}^{r \times N_j} \) in spatial direction \( j \); operator \( \otimes \) denotes tensor product. Such a representation for order-2 tensors (matrices) can be obtained by various matrix factorizing methods among which Singular Value Decomposition (SVD)
is prominent since it provides a factorization that minimizes the mean square error of an $r$-term approximation. The level of compression (reduction of memory requirements) depends on the rank $r$. When one is interested to find a solution in such a form, it is necessary to use several operations occurring in Fourier Galerkin method, particularly FFT and element-wise multiplication.

The linearity of Fourier transform facilitates to express $d$-dimensional FFT of a tensor as sum of tensor products of 1-dimensional FFTs, i.e.,

$$
\mathcal{F}_N(\tilde{v}) = \sum_{i=1}^r c[i] \mathcal{F}_{N_1}(b_{(1)}^{(1)}[i]) \otimes \mathcal{F}_{N_2}(b_{(2)}^{(2)}[i]).
$$

For the same number of discretisation points in all spatial directions $N_j = N$, this FFT algorithm has a complexity $O(drN \log N)$ which is much better than full $O(dN^d \log N)$ when the rank $r$ is kept low. Note that this operation keeps a rank of 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 case of the canonical format it is computed as:

$$
\tilde{v} + \tilde{w} = \sum_{i=1}^r c_v[i] \left( b_{(1)}^{(1)}[i] \otimes b_{(2)}^{(2)}[i] \right) + \sum_{k=1}^s c_w[k] \left( b_{(1)}^{(1)}[k] \otimes b_{(2)}^{(2)}[k] \right),
$$

$$
\tilde{v} \odot \tilde{w} = \sum_{i=1}^r \sum_{k=1}^s c_v[i] c_w[k] \left( b_{(1)}^{(1)}[i] \odot b_{(1)}^{(1)}[k] \right) \otimes \left( b_{(2)}^{(2)}[i] \odot b_{(2)}^{(2)}[k] \right).
$$

While addition of two tensor costs no floating point operations, the element-wise multiplication has a complexity $rsdN$ which is significantly cheaper than $N^d$ operations for full tensors, especially when $r$ and $s$ are much smaller than $N$.

### 3.1.2 Tucker format

The decomposition of higher order tensors have many variants. A Tucker format representation is linked to the definition of 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 basis vectors $\{ b^{(j)}[i] | i = 1, \ldots, r_j \}$ in spatial direction $j$. The Tucker format is then a linear combination of tensor products of all possible combinations of basis vectors in different directions, i.e.

$$
\mathbf{v} \approx \sum_{i_1=1}^{r_1} \cdots \sum_{i_d=1}^{r_d} c[i_1, \ldots, i_d] \bigotimes_{j=1}^d b^{(j)}[i_j],
$$

where the core $c \in \mathbb{R}^r$ is a tensor of order $d$. The canonical 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 spatial directions.

### 3.1.3 Tensor train (TT) format

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 individual spatial dimensions. In 3-dimensional case, the decomposition of the tensor of size $N \times N \times N$ is provided in two steps. Using the standard SVD algorithm the decomposition is provided 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 leads to such an equivalent expression:

$$v = \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} b^{(1)}[1, :, i_1] \otimes b^{(2)}[i_1, :, i_2] \otimes b^{(3)}[i_2, :, 1]$$  \hspace{1cm} (9)

where the vectors $b^{(j)}[i_j-1, :, i_j] \in \mathbb{R}^{N_j}$ are then basis vectors in spatial direction $j$. The tensor’s components can be explicitly written as

$$v[k] = \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} b^{(1)}[1, k_1, i_1]b^{(2)}[i_1, k_2, i_2]b^{(3)}[i_2, k_3, 1].$$

The tensor train format in (9) 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 FFT on the train *carriages* along the second axis, i.e. applied on $b^{(j)}[i_j-1, :, i_j] \in \mathbb{K}^{N_j}$.

### 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 could inflate the representation rank $r$, potentially at a very fast rate, which hinder the computation. On the other hand in the resulted representation a large part of the $r$ terms are not essential and could be given up without loss of accuracy if done correctly. Further truncations are also necessary to trade accuracy for efficiency, especially in the cases of using iterative solvers.

Rank truncations of tensors in the three low rank formats are all based on SVD or high order SVD (HOSVD) which provide optimal or suboptimal truncations and error estimates. We supplement a more detailed introduction about them in the Appendix A.

### 3.2 Linear systems in low-rank formats

Here, the application of low-rank formats on linear systems (8a) and (8b) is discussed. The solution vector $\hat{u}$ is searched in a low-rank format which require to solve the linear systems

$$\mathcal{T}\hat{P}^{-1}I\hat{\nabla}_N^*\mathcal{F}_N\mathcal{T}\hat{A}\mathcal{F}^{-1}_N\hat{\nabla}_N\hat{u} = -\mathcal{T}\hat{P}^{-1}\hat{\nabla}_N^*\mathcal{F}_N\hat{A}\mathcal{E},$$

$$\mathcal{T}\hat{P}^{-1}I\hat{\nabla}_N^*\mathcal{F}_N\mathcal{T}\hat{A}\mathcal{F}^{-1}_N\hat{\nabla}_N\hat{u} = -\mathcal{T}\hat{P}^{-1}\hat{\nabla}_N^*\mathcal{F}_N\mathcal{T}\hat{A}\mathcal{F}^{-1}_N\hat{\nabla}_N\hat{u} = -\mathcal{T}\hat{P}^{-1}\hat{\nabla}_N^*\mathcal{F}_N\mathcal{T}\hat{A}\mathcal{F}^{-1}_N\hat{\nabla}_N\hat{u} = -\mathcal{T}\hat{P}^{-1}\hat{\nabla}_N^*\mathcal{F}_N\mathcal{T}\hat{A}\mathcal{F}^{-1}_N\hat{\nabla}_N\hat{u},$$

using an iterative solver introduced in 3.3. These systems are supplemented with the rounding (truncation) $\mathcal{T}$ to keep the $r$ at an affordable level in the low rank formats. These systems involve several operations like differentiation, divergence, and Fourier transform etc. They are tabulated in the Table I together with the corresponding implementations in low rank format and their impact on the rank $r$. 

10
| Operation                               | Low rank tensor implementation | Rank $r$ |
|----------------------------------------|--------------------------------|----------|
| Differentiation (gradient)             | element-wise multiplication    | remain unchanged |
| Divergence                             | element-wise multiplication and addition | increased |
| Evaluation of material law             | element-wise multiplication    | increased |
| $d$-dimensional FFT                    | series of 1D FFT               | remain unchanged |
| Preconditioning                        | element-wise multiplication    | increased |

Table 1: Operations and their implementations in low-rank formats

Since the material coefficients $\tilde{A}, A$ and the preconditioner $P^{-1}$ 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 maintain reasonable accuracy in the solution.

The application of the gradient and divergence in Fourier space is also implemented as element-wise multiplications. The differentiating operator for Fourier series is by nature a rank-1 tensor in the form

$$\hat{\nabla}_N = [2\pi i K_1 \otimes 1 \otimes 1, 1 \otimes 2\pi i K_2 \otimes 1, 1 \otimes 1 \otimes 2\pi i K_3]$$

in the 3D setting, where $K_\alpha = (k \in \mathbb{Z}; |k| < N/2)$ is a vector of all discrete frequencies in spatial direction $\alpha$. So the corresponding element-wise multiplication keeps the rank of tensors unchanged. However for divergence the contraction along the first component of $\hat{\nabla}_N$ is provided by operation addition of two low-rank formats, which increases the rank hence a truncation has to be performed.

The last operation that occurs in the system is $d$-dimensional fast Fourier transform (FFT) which is efficiently evaluated using 1-dimensional FFTs. Moreover the rank of the tensor is kept unchanged in this operation.

### 3.3 Linear solvers

For the full solver we have used Conjugate gradients which is considered to be the best available solver for FFT-based homogenisation [17]. However, the linear systems accelerated by low-rank approximations require solvers that are insensitive to the small perturbations as the matrix-vector product is computed only approximately, due to the rounding of tensors. Therefore the conjugate gradients that build on the orthogonalisation of Krylov subspace vectors using short-term recurrence relation are inappropriate.

The systems with low-rank approximations are solved here with Minimal Residual iteration [48] 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

$$Cu = d,$$

by the iteration

$$u_{(i+1)} = u_{(i)} + \omega \left( d - Cu_{(i)} \right) = (I - \omega C) u_{(i)} + \omega d.$$
The parameter $\omega$ is chosen such that the iterative matrix $(I - \omega C)$ has a norm smaller than one to guarantee convergence. In Richardson iteration a fixed optimal $\omega$ value is set on the basis of apriori knowledge about the eigenvalues of the system matrix $C$, i.e.

$$\omega = \frac{\lambda_{\text{min}}(C) + \lambda_{\text{max}}(C)}{2},$$

because it satisfies the minimal norm of the iterative matrix as proposed in [49] for FFT-based homogenisation. Here $\lambda_{\text{min}}(C)$ denotes the smallest positive eigenvalue as the system matrix is only positive semidefinite. Particularly, the linear systems in [8] 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 satisfies that the solution is delivered on compatible fields.

In Minimal Residual approach the $\omega$ is chosen at each iteration as the minimizer of the next residual $r_{(i+1)}$ over all increments of $u$ in the direction of $r_{(i)}$, i.e.

$$u_{(i+1)} = u_{(i)} + \omega_{(i)} r_{(i)}, \quad \text{with} \quad \omega_{(i)} = \frac{(C r_{(i)}, r_{(i)})}{\|C r_{(i)}\|^2}.$$

We adopt the latter method in this work, because of our observation that the Minimal Residual iteration has a higher accuracy than Richardson iteration. For a low-rank approximation of a solution vector, note that the solver has to deal with the matrix vector product $Cu_{(i)}$, which is provided only approximately to limit the growth of the solution rank. The rank also grow by the operation addition during the iteration. Therefore, a truncation is provided at each step of the low-rank variant of the Minimal Residual iteration, i.e.

$$u_{(i+1)} = T[u_{(i)} + \omega_{(i)} (d - Cu_{(i)})].$$

4 Numerical results

The numerical results that are here presented were calculated using software FFTHomPy, which is freely available at https://github.com/vondrejc/FFTHomPy.

4.1 Material parameters

Here, we present two material examples on which we did numerical tests. The first is defined as

$$A_\square(x) = I(1 + \rho f_\square(x))$$

where $I \in \mathbb{R}^{d \times d}$ is the identity matrix and the parameter $\rho = 10$ corresponds to a material contrast. The function $f_\square : \mathcal{Y} \to \mathbb{R}^d$ describing the topology of the inclusions is defined on a unit cell $\mathcal{Y} = (-\frac{1}{2}, \frac{1}{2})^d$ as

$$f_\square(x) = \begin{cases} 1 & \text{for } x \text{ such that } x_i < 0.3 \text{ for } i = 1, \ldots, d, \\ 0 & \text{otherwise} \end{cases},$$

which is also depicted in 2D in Figure 1. The corresponding low-rank approximation have rank 2 for all three formats (canonical, Tucker, tensor train).
As a second example one sample of a stochastic material has been considered using the Karhunen-Loève expansion of a square exponential covariance function. It is a linear combination of basis functions, which are trigonometric polynomials $\varphi^k$, of a corresponding covariance operator. Since the linear combination may fail to be positive definite the exponential function is incorporated because it maps the random variable with the range on real domain to positive numbers. Therefore the following material coefficients

$$A_S(x) = I \exp(C + \sum_{k \in I} c[k] \varphi^k(x)),$$

(11)

which are plotted in Figure 2 have been considered and approximated in a low-rank format. The constants $C$ and $c$ have been chosen such that the minimal value of $A(x)$ is 1 and maximal 10. The rank of the approximation has been fixed to 10. For a comparison to 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 spatial direction $N = [N, \ldots, N] \in \mathbb{R}^d$.

### 4.2 Behaviour of linear systems during iterations

The evolution of the norm during the Minimal Residual iteration is investigated because it well describes the character of low-rank approximations. The numerical results depicted in Figure 3 are
Figure 3: Evolution of the norm of residua during Minimal Residuum iterations; computed in 2D for $N = 1215$ and in 3D for $N = 135$.

computed with respect to the norm of residuum $r = d - Cu(i)$ in iteration $i$ as

$$\|r\| = \left( \sum_{k \in \mathbb{Z}_N} |r[k]|^2 \right)^{\frac{1}{2}}$$

because it corresponds to the $L^2$-norm of the corresponding trigonometric polynomial. Note that since problem is solved in Fourier space, the residuum components agree with the Fourier coefficients of the corresponding trigonometric polynomial.

Although the rounding of the growing tensor 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 tensor rank significantly increase resulting in unnecessary computational demands, especially when the tensors are far away from the solution. Therefore the truncation has been performed always to a fixed rank. The solution which is from a large dimensional space $\mathbb{R}^N$ 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.

Similar results in Figure 3 have been obtained regardless the discretisation method (Ga and GaNi), material problem (□ and S), or the low-rank format (canonical, Tucker, TT). We can also observe for solutions with different ranks that the norms of residua are reduced at almost the same rate until the residuum reaches a threshold error and start to stagnate. This proposes a rank adapting solver that starts with a lower solution rank and increases the rank during the iterations.

4.3 Discretisation error of low-rank approximation

![Graphs showing relative errors for low-rank solutions in 2D and 3D for different discretisation methods.]

Figure 4: Relative errors of low-rank solutions computed in 2D for $N = 1215$ and in 3D for $N = 135$.

In the Figure 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 $A_{H,N,r}$ and of the full solution $A_{H,N}$ has been used, i.e.

$$\text{relative error} = \frac{A_{H,N} - A_{H,N,r}}{A_{H,N}}.$$  \hspace{1cm} (12)
This is chosen because the error in homogenised properties corresponds to the square of energetic semi-norm (norm on zero-mean fields) of the algebraic error between full solution and low rank solution

$$\|u_N - u_{N,r}\|^2_A = a(\nabla u_N - \nabla u_{N,r}, \nabla u_{N} - \nabla u_{N,r}) = A_{H,N,r} - A_{H,N};$$

for the derivation see [50, Appendix D]. We also note that the full solution $u_N$ has been computed using Conjugate gradients with high accuracy (tolerance $10^{-10}$ on norm of residuum) to obtain a solution that is close to the exact one. The low-rank solution has been obtained from Minimal Residual iteration, which has been stopped when the residuum failed to be decreased.

We can observe that the results are again similar regardless the discretisation method (Ga and GaNi), material problem (□ and S), or the low-rank format (canonical, Tucker, TT). The increase in the solution rank leads to a significant reduction of the relative error. Note that since the stochastic material is smooth there is almost no difference between the discretisation using exact integration (Ga) and numerical integration (GaNi). It also shows the low rank method is more accurate for problem with continuous material property (S) than for the one with discontinuous coefficients (□).

### 4.4 Memory and computational efficiencies

| Format   | Element-wise product | FFT$_d$  | Truncation       |
|----------|----------------------|----------|------------------|
| full     | $N^d$                | $O(N^d \log N)$ | —                |
| canonical| $dNr$                | $O(drN \log N)$  | $O(dNr^2)$       |
| Tucker   | $dNr + r^d s^d$      | $O(drN \log N)$  | $O(dNr^2 + r^{d+1})$ |
| TT       | $dNr^2 s^2$          | $O(dNr^2 \log N)$ | $O(dNr^3)$       |

Table 2: Asymptotic computational complexities in terms of floating point multiplications performed on $d$-dimensional arrays of shape $(N, \ldots, N)$ considering canonical, Tucker, and tensor-train (TT) formats.

| format   | memory requirements         |
|----------|-------------------------------|
| full     | $N^d$                         |
| canonical| $dNr$                         |
| Tucker   | $dNr + r^d$                   |
| TT       | $2Nr + (d-2)Nr^2$             |

Table 3: Memory requirements to store $d$-dimensional array of shape $(N, \ldots, N)$ for full, canonical, Tucker, and tensor-train (TT) formats.

Here, we discuss the computational and memory requirements to resolve the linear system using low-rank approximations. As we are using several low-rank formats and several operations on them, the computational complexities and memory requirements are summarised in Tables 2 and 3. The memory requirements of the FFT-based systems are directed by memory requirements for material coefficients, preconditioner, solution vector, and possibly other vectors needed to store as a requirement of a linear solver. Provided that the ranks are kept small, the memory of sparse solvers scales linearly with $N$ while full solvers with $N^d$, which makes the method effective particularly for higher dimensional problems.
Therefore the computational requirements for full and sparse solver has been compared on the same level of accuracy. As a criterion the relative error (12) has been considered with the reference full solution that has been calculated on the regular grid \((N, \ldots, N)\) with the tolerance \(10^{-6}\) on the norms of residua. In order to achieve the same level of accuracy with the sparse solver, the sparse solver has been computed on a bigger grid \((\alpha N, \ldots, \alpha N)\) with some multiplier \(\alpha \in \mathbb{N}\). This creates a great possibility for a rank reduction in the sparse solution. The results in Figure ?? shows that the computational time scales as \(N^d\) for a full solution and almost linearly for sparse solvers. The latter is therefore significantly faster for higher \(N\)’s despite it is calculated on a larger computational grid. The solution rank increases with the increased accuracy only slowly, reaching the value ??? for \(N = ???\).

![Figure 5: The computational time at the same level of accuracy for the scheme with exact integration. The full solution has been calculated on a grid of size \((N, \ldots, N)\) while the sparse solution on the grid \((3N, \ldots, 3N)\) with a solution rank to achieve the same level of accuracy as full scheme.](image)

5 Conclusion

This paper is focused on acceleration of Fourier–Galerkin method using low-rank approximations for problems of numerical homogenisations. The efficiency of this approach builds on incorporation of the fast Fourier transform (FFT) algorithm into the system matrix which makes it efficient for iterative linear solvers. The complexity of full computation is based on FFT algorithm which is also very natural for low-rank formats as the \(d\)-dimensional FFT is transformed into the series of one-dimensional FFTs. In this paper three formats — canonical, Tucker, and tensor train (TT) — has been considered.

The main results are summarised:

- The method is suitable for material coefficients with relatively smaller rank compared to the size of the discretisation grid. For a fixed material rank the low-rank approximation will be always beneficial for large discretisation grids because of better scalability of the algorithms.

- Low-rank approximations lead to a significant memory and computational cost reduction.
For very complex materials, the low-rank approximation could be used as a preconditioner or as a model order reduction technique.

Since the low-rank approximation provides a significant memory reduction it can allow to compute the problems on a finer grid compared to the full solution. It means that with fixed memory demands the low-rank approximation technique can provide better accuracy than full solutions.

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 canonical, Tucker and tensor train formats.

A.1 Canonical format

A canonical (or canonical polyadic(CP), or r-term) representation \( \mathbf{v}_r \) of a tensor \( \mathbf{v} \in \mathbb{K}^{N_1 \times \cdots \times N_d} \) (\( \mathbb{K} \) is either \( \mathbb{R} \) or \( \mathbb{C} \)) is a sum of \( r \) rank-1 tensors, i.e.

\[
\mathbf{v} \approx \mathbf{v}_r = \sum_{i=1}^{r} c[i] \bigotimes_{j=1}^{d} \mathbf{b}^{(j)}[i].
\]

with \( \mathbf{b}^{(j)} \in \mathbb{K}^{r \times N_j} \) and \( \bigotimes \) denotes tensor product. This format has a 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 \( \mathbf{v}_r \) is not always feasible [51, Proposition 9.10].

A.1.1 Element-wise multiplication

The element-wise (Hadamard) product of two tensors in canonical format is computed as:

\[
\mathbf{v}_r \odot \mathbf{w}_s = \sum_{i=1}^{r} \sum_{k=1}^{s} c_v[i] c_w[k] \bigotimes_{j=1}^{d} (\mathbf{b}_{v}^{(j)}[i] \odot \mathbf{b}_{w}^{(j)}[k]).
\]

This operation has a complexity \( rs \sum_{j=1}^{d} N_j \). The product has a new rank \( rs \).

A.1.2 Fourier transform

Due to the linearity of Fourier transform \( \mathcal{F} \), a \( d \)-dimensional Fourier transform of a canonical tensor is broken down to a series of 1-d Fourier transform, i.e.,

\[
\mathcal{F}_N(\mathbf{v}_r) = \sum_{i=1}^{r} c[i] \bigotimes_{j=1}^{d} \mathcal{F}_{N_j}(\mathbf{b}_{v}^{(j)}[i]).
\]

Hence a FFT on canonical tensor has a complexity \( dr N \log N \).

A.1.3 Rank truncation

Operations (e.g. element-wise multiplication) applied on tensors in canonical format could inflate the representation rank. This calls for a model reduction up 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 matrix $B^{(j)} \in \mathbb{K}^{N_j \times r}$ collects the vectors $\{b^{(j)}[i]\}_{i=1}^r$ for the $j$-th dimension, we have their re-orthogonalisations $B^{(1)} = Q^{(1)}R^{(1)}$ and $B^{(2)} = Q^{(2)}R^{(2)}$ by QR decompositions. A SVD $R^{(1)}R^{(2)} = U^{(1)}\Sigma(U^{(2)})^\top$ facilitates the truncation. Suppose $U^{(1)}_k$, $U^{(2)}_k$ and $\Sigma_k$ are the truncated ones with rank $k \leq r$, the truncated form of the canonical representation is

$$v_k = \sum_{i=1}^k c[i]\hat{b}^{(1)}[i] \otimes \hat{b}^{(2)}[i]$$

where $\hat{b}^{(1)}[i]$, $\hat{b}^{(2)}[i]$ are the columns of $Q^{(1)}U^{(1)}_k$, $Q^{(2)}U^{(2)}_k$ respectively, and $c[i]$ are the diagonal entries of $\Sigma_k$.

For $d \geq 3$, the $k$-rank form could be obtained by numerical error minimizing procedures \[51\], 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.

### A.2 Tucker format

A Tucker format representation (or tensor subspace representation) of a tensor $v \in \mathbb{K}^{N_1 \times \cdots \times N_d} \in \mathcal{V}$ is a linear combination of 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 $\{b^{(j)}[i_j] \in \mathbb{K}^{N_j} : 1 \leq i_j \leq r_j\}$. The tensors $\bigotimes_{j=1}^d b^{(j)}[i_j]$ for all $1 \leq i_j \leq r_j$ form the bases of the space $\mathcal{V}$. Then we have unique coefficient $c[i_1, i_2, \ldots, i_d]$ for every $v \in \mathcal{V}$ such that

$$v \approx v_r = \sum_{i_1=1}^{r_1} \cdots \sum_{i_d=1}^{r_d} c[i_1, i_2, \ldots, i_d] \bigotimes_{j=1}^d b^{(j)}[i_j],$$

where $c \in \mathbb{K}^{r_1 \times \cdots \times r_d}$ is called a core tensor. Given any prescribed rank vector $r$, an error minimizing approximation $v_r$ can be found by a high-order singular value decomposition (HOSVD) \[52\].

#### A.2.1 Element-wise multiplication

Let another Tucker format tensor with rank $s$ be defined as

$$w_s = \sum_k c_w[k] \bigotimes_{j=1}^d b^{(j)}[k_j]$$

the element-wise (Hadamard) product of $v_r$ and $w_s$ has also a Tucker format

$$v_r \odot w_s = \sum_l c[l] \bigotimes_{j=1}^d b^{(j)}[l_j]$$

where $t = r \odot s$ and $c = c_v \otimes c_w$, i.e. the Kronecker product of the two coefficient tensors. So for any $1 \leq j \leq d$, the index $l_j$ is related to $i_j$ and $k_j$ by $l_j = i_jk_j = i_jr_j + k_j$, and $u$ is obtained from $v$ and $w$ through

$$u^{(j)}_{i_j} = u^{(j)}_{i_jk_j} = u^{(j)}_{i_j} \odot u^{(j)}_{k_j} \quad \text{for} \quad 1 \leq i_j \leq r_j, \ 1 \leq k_j \leq s_j$$

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 $dNr^s + r^d s^d$ in which the first term is the cost for computing $\{u^{(j)}_{i_j} : 1 \leq l_j \leq R_j\}_{j=1}^d$ and the second for the Kronecker product of coefficient tensors.
A.2.2 Fourier transform

The Fourier transform of \( v_r \) is

\[
F_N(v_r) = \sum_i^r c[i] \bigotimes_{j=1}^d F_N_j(b^{(j)}[i_j])
\]

which only involves the basis vectors. If FFT is applied the complexity is \( drN \log N \).

A.2.3 Rank truncation

The Tucker representation (13) 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:

\[
\sigma_1^{(j)} \geq \sigma_2^{(j)} \geq \cdots \geq \sigma_r^{(j)}, \text{ for } j = 1, \cdots, d,
\]

where \( \sigma_{ij}^{(j)} \) is the 2-norm of the \( ij \)-th slice of the core tensor \( c \) cut on the \( j \)-th dimension.

If the truncation rank is \( k_j < r_j \), the error of the truncated representation \( v_k \) is bounded by

\[
\|v_r - v_k\| \leq \left[ \sum_{j=1}^d \sum_{i=k_j+1}^{r_j} (\sigma_{ij}^{(j)})^2 \right]^{1/2}.
\]

In the second case the bases \( \{b^{(j)}[i_j]\}_{i_j=1}^{r_j} \) 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 (as detailed in [51]) is analogous to the re-orthogonalisation and SVD for the 2D canonical format representations, but in higher dimensionality.

A.3 Tensor train format

A tensor train (TT) representation \([37]\) of a tensor \( v \in \mathbb{K}^{N_1 \times \cdots \times N_d} \) can be expressed as a series of consecutive contractions of order-3 tensors \( b^{(j)} \in \mathbb{K}^{r_j-1 \times N_j \times r_j} \) for \( j = 1, \cdots, d \) which are the carriages of the tensor train. An equivalent expression in the form of tensor products is

\[
v \approx v_r = \sum_{i_1=1}^{r_1} \cdots \sum_{i_{d-1}=1}^{r_{d-1}} b^{(1)}_v[1, \cdots, i_1] \otimes b^{(2)}_v[i_1, \cdots, i_2] \otimes \cdots \otimes b^{(d)}_v[i_{d-1}, \cdots, 1]
\]

\( r \) is the TT-rank of \( v \) with a constrain \( r_0 = r_d = 1 \) to keep the elements of \( v \) scalars. The TT format is stable in the sense that for any prescribed \( r \) an error minimizing \( v_r \) can always be constructed by a series of SVDs on consecutive matricizations of \( v \).

A.3.1 Element-wise multiplication

Let another TT format tensor with rank \( s \) be defined as

\[
w_s = \sum_{i_1=1}^{s_1} \cdots \sum_{i_{d-1}=1}^{s_{d-1}} b^{(1)}_w[1, \cdots, i_1] \otimes b^{(2)}_w[i_1, \cdots, i_2] \otimes \cdots \otimes b^{(d)}_w[i_{d-1}, \cdots, 1]
\]
with \( b_w^{(j)} \in \mathbb{K}^{s_j - 1 \times N_j \times s_j} \). The element-wise product of \( \mathbf{v}_r \) and \( \mathbf{w}_s \) can also be expressed in TT format:

\[
\mathbf{v}_r \odot \mathbf{w}_s = \sum_{i_1=1}^{t_1} \cdots \sum_{i_{d-1}=1}^{t_{d-1}} \mathbf{b}^{(1)}[1,:,i_1] \otimes \mathbf{b}^{(2)}[i_1,:,i_2] \otimes \cdots \otimes \mathbf{b}^{(d)}[i_{d-1},:,1]
\]

where \( t = r \odot s \) and \( \mathbf{b}^{(j)} = \mathbf{b}_w^{(j)} * \mathbf{b}_w^{(j)} \). Here the \( \ast \) denotes one type of Khatri–Rao product that in this case yields an order-3 tensor \( \mathbf{b}^{(j)} \in \mathbb{K}^{r_j - 1 \times N_j \times r_j s_j} \). The complexity of the element-wise product is bounded by \( dN^2 r^2 s^2 \) operations with \( N, r \) and \( s \) as defined in the subsection [A.2].

### A.3.2 Fourier transform

The Fourier transform of \( \mathbf{v}_r \) can also be carried out by doing 1-D transforms on each carriage:

\[
\mathcal{F}_N(\mathbf{v}_r) = \sum_{i_1=1}^{r_1} \cdots \sum_{i_{d-1}=1}^{r_{d-1}} \mathcal{F}_{N_1}(\mathbf{b}_v^{(1)}[1,:,i_1]) \otimes \mathcal{F}_{N_2}(\mathbf{b}_v^{(2)}[i_1,:,i_2]) \otimes \cdots \otimes \mathcal{F}_{N_d}(\mathbf{b}_v^{(d)}[i_{d-1},:,1])
\]

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 the order \( dr^2 N \log N \).

### A.3.3 Rank truncation

The tensor train representation [9] 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) [37], or as a result of operations (e.g. additions or multiplications) over tensor train operands. In the first case, an error minimizing rank truncation could be directly carried out in the TT-SVD process. The truncation has an error bound \( \left( \sum_{k=1}^{d-1} \epsilon_k^2 \right)^{1/2} \), where \( \epsilon_k \) 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-rounding.

For the first case, the complexity of truncation is the same as that for the TT-SVD, which is of order \( O(N^{d+1}) \). A cheaper alternative for TT-SVD is TT-cross Approximation as introduced in [36]. The complexity of TT-rounding in the second case is of order \( O(dN^3 r^3) \).

### References

[1] H. Moulinec, 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 (11) (1994) 1417–1423.

[2] J. Vondřejc, J. Zeman, I. Marek, An FFT-based Galerkin method for homogenization of periodic media, Computers & Mathematics with Applications 68 (3) (2014) 156–173. [arXiv:1311.0089](https://arxiv.org/abs/1311.0089), [doi:10.1016/j.camwa.2014.05.014](https://doi.org/10.1016/j.camwa.2014.05.014).

[3] G. W. Milton, The Theory of Composites, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, UK, 2002.

[4] J. Zeman, T. W. J. de Geus, J. Vondřejc, R. H. J. Peerlings, M. G. D. Geers, A finite element perspective on non-linear FFT-based micromechanical simulations, International Journal for
[5] T. W. J. de Geus, J. Vondřejc, J. Zeman, R. H. J. Peerlings, M. G. D. Geers, Finite strain FFT-based non-linear solvers made simple, Computer Methods in Applied Mechanics and Engineering 318 (2017) 412–430. arXiv:1603.08893 doi:10.1016/j.cma.2016.12.032.

[6] J. Vondřejc, J. Zeman, I. Marek, Guaranteed upper-lower bounds on homogenized properties by FFT-based Galerkin method, Computer Methods in Applied Mechanics and Engineering 297 (2015) 258–291. arXiv:1404.3614 doi:10.1016/j.cma.2015.09.003.

[7] J. Zeman, J. Vondřejc, J. Novák, I. Marek, Accelerating a FFT-based solver for numerical homogenization of periodic media by conjugate gradients, Journal of Computational Physics 229 (21) (2010) 8065–8071. doi:10.1016/j.jcp.2010.07.010.

[8] 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) (2016) 1106–1135. arXiv:1412.2033 doi:10.1002/nme.5199.

[9] M. Schneider, Convergence of FFT-based homogenization for strongly heterogeneous media, Mathematical Methods in the Applied Sciences 38 (13) (2014) 2761–2778. doi:10.1002/mma.3259.

[10] S. Brisard, 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 (2012) 197–212. doi:10.1016/j.cma.2012.01.003.

[11] S. Brisard, L. Dormieux, FFT-based methods for the mechanics of composites: A general variational framework, Computational Materials Science 49 (3) (2010) 663–671. doi:doi:10.1016/j.commatsci.2010.06.009.

[12] M. Schneider, D. Merkert, M. Kabel, FFT-based homogenization for microstructures discretized by linear hexahedral elements, International Journal for Numerical Methods in Engineering 109 (10) (2017) 1461–1489. arXiv:1010.1724, doi:10.1002/nme.5336.

[13] F. Willot, Fourier-based schemes for computing the mechanical response of composites with accurate local fields, Comptes Rendus Mécanique 343 (2015) 232–245. doi:10.1016/j.crme.2014.12.005.

[14] F. Willot, B. Abdallah, 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) (2014) 518–533. arXiv:1307.1015 doi:10.1002/nme.4641.

[15] D. J. Eyre, G. W. Milton, A fast numerical scheme for computing the response of composites using grid refinement, The European Physical Journal Applied Physics 6 (1) (1999) 41–47.

[16] H. Moulinec, P. Suquet, G. W. Milton, Convergence of iterative methods based on Neumann series for composite materials: theory and practice. arXiv:1711.05880.
[17] N. Mishra, J. Vondřejc, J. Zeman, A comparative study on low-memory iterative solvers for FFT-based homogenization of periodic media, Journal of Computational Physics 321 (2016) 151–168. arXiv:1508.02045 doi:10.1016/j.jcp.2016.05.041

[18] M. Kabel, T. Böhlke, M. Schneider, Efficient fixed point and Newton–Krylov solvers for FFT-based homogenization of elasticity at large deformations, Computational Mechanics 54 (6) (2014) 1497–1514. doi:10.1007/s00466-014-1071-8

[19] M. Schneider, An FFT-based fast gradient method for elastic and inelastic unit cell homogenization problems, Computer Methods in Applied Mechanics and Engineering 315 (2017) 846–866. doi:10.1016/j.cma.2016.11.004

[20] J. Kochmann, S. Wulfinghoff, L. Ehle, J. Mayer, B. Svendsen, S. Reese, Efficient and accurate two-scale FE-FFT-based prediction of the effective material behavior of elasto-viscoplastic polycrystals, Computational Mechanics (2017) 1–14 doi:10.1007/s00466-017-1476-2

[21] F. S. Göküzüm, M.-A. Keip, An Algorithmically Consistent Macrosopic Tangent Operator for FFT-based Computational Homogenization, International Journal for Numerical Methods in Engineering doi:10.1002/nme.5627

[22] F. Dietrich, D. Merkert, B. Simeon, Derivation of higher-order terms in FFT-based numerical homogenization arXiv:1712.05145.

[23] N. Bertin, L. Capolungo, A FFT-based formulation for discrete dislocation dynamics in heterogeneous media, Journal of Computational Physics 355 (2018) 366–384. doi:10.1016/J.JCP.2017.11.020

[24] T. de Geus, R. Peerlings, 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 (2016) 687–698. doi:10.1016/j.ijsolstr.2016.03.029

[25] M. Boeff, F. Gutknecht, P. S. Engels, A. Ma, A. Hartmaier, Formulation of nonlocal damage models based on spectral methods for application to complex microstructures, Engineering Fracture Mechanics 147 (2015) 373–387. doi:10.1016/j.engfracmech.2015.06.030

[26] J. Segurado, R. A. Lebensohn, J. LLorca, Computational Homogenization of Polycrystals arXiv:1804.02538.

[27] S. A. Goreinov, E. E. Tyrtyshnikov, N. L. Zamarashkin, A theory of pseudoskeleton approximations, Linear Algebra and Its Applications 261 (1-3) (1997) 1–21. doi:10.1016/S0024-3795(96)00301-1

[28] M. Bebendorf, Approximation of boundary element matrices, Numerische Mathematik 86 (4) (2000) 565–589. doi:10.1007/PL00005410

[29] W. Hackbusch, Numerical tensor calculus, Acta numerica 23 (2014) 651–742. doi:10.1017/S0962492914000087.

[30] T. G. Kolda, B. W. Bader, Tensor Decompositions and Applications, SIAM Review 51 (3) (2009) 455–500. doi:10.1137/07070111X.
[31] L. Giraldi, A. Nouy, G. Legrain, P. Cartraud, Tensor-based methods for numerical homogenization from high-resolution images, Computer Methods in Applied Mechanics and Engineering 254 (2013) 154–169. doi:10.1016/j.cma.2012.10.012

[32] D. Kressner, C. Tobler, Low-Rank Tensor Krylov Subspace Methods for Parametrized Linear Systems, SIAM Journal on Matrix Analysis and Applications 32 (4) (2011) 1288–1316. doi:10.1137/100799010.

[33] C. Tobler, Low-rank tensor methods for linear systems and eigenvalue problems, Ph.D. thesis, ETH Zürich (2012).

[34] 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) (2013) 149–172. arXiv:1206.5512, doi:10.1515/rnam-2013-0009.

[35] J. Ballani, L. Grasedyck, A projection method to solve linear systems in tensor format, Numerical Linear Algebra with Applications 20 (1) (2013) 27–43. doi:10.1002/nla.1818.

[36] I. Oseledets, E. Tyrtyshnikov, TT-cross approximation for multidimensional arrays, Linear Algebra and its Applications 432 (1) (2010) 70–88. doi:10.1016/J.LAA.2009.07.024.

[37] I. V. Oseledets, Tensor-Train Decomposition, SIAM Journal on Scientific Computing 33 (5) (2011) 2295–2317. doi:10.1137/090752286.

[38] D. Bigoni, A. P. Engsig-Karup, Y. M. Marzouk, Spectral Tensor-Train Decomposition, SIAM Journal on Scientific Computing 38 (4) (2016) A2405–A2439. arXiv:1405.5713, doi:10.1137/15M1036919.

[39] M. Espig, W. Hackbusch, A. Litvinenko, H. G. Matthies, P. Wähnert, Efficient low-rank approximation of the stochastic Galerkin matrix in tensor formats, Computers & Mathematics with Applications 67 (4) (2014) 818–829. doi:10.1016/j.camwa.2012.10.008.

[40] H. G. Matthies, E. Zander, Solving stochastic systems with low-rank tensor compression, Linear Algebra and its Applications 436 (10) (2012) 3819–3838. doi:10.1016/j.laa.2011.04.017.

[41] A. Nouy, Low-rank methods for high-dimensional approximation and model order reduction, in: Model Reduction and Approximation: Theory and Algorithms, 2015, pp. 1–73. arXiv:1511.01554, doi:10.1007/978-3-319-11259-6_21-1.

[42] B. N. Khoromskij, C. Schwab, Tensor-Structured Galerkin Approximation of Parametric and Stochastic Elliptic PDEs, SIAM Journal on Scientific Computing 33 (1) (2011) 364–385. doi:10.1137/100785715.

[43] A. Nouy, Low-Rank Tensor Methods for Model Order Reduction, in: Handbook of Uncertainty Quantification, Springer International Publishing, Cham, 2015, pp. 1–26. doi:10.1007/978-3-319-11259-6_21-1.

[44] A. Bensoussan, J.-L. Lions, G. Papanicolaou, Asymptotic Analysis for Periodic Structures, North Holland, Amsterdam, 1978.

[45] J. Saranen, G. Vainikko, Periodic Integral and Pseudodifferential Equations with Numerical Approximation, Springer Monographs Mathematics, Berlin, Heidelberg, 2002.
[46] M. Ladecký, I. Pultarová, J. Vondřejc, J. Zeman, Preconditioning the spectral Fourier method for homogenization of periodic media (2019). [https://mat.fsv.cvut.cz/nales/preprints/preprinty/2019/preprinty2019.html](https://mat.fsv.cvut.cz/nales/preprints/preprinty/2019/preprinty2019.html)

[47] J. Vondřejc, Double-grid quadrature with interpolation-projection (DoGIP) as a novel discretisation approach: An application to FEM on simplices (2017). [arXiv:1710.09913](https://arxiv.org/abs/1710.09913).

[48] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd Edition, SIAM, Philadelphia, PA, USA, 2003.

[49] H. Moulinec, 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) (1998) 69–94. [doi:10.1016/S0045-7825(97)00218-1](https://doi.org/10.1016/S0045-7825(97)00218-1).

[50] J. Vondřejc, T. de Geus, Energy-based comparison between Fourier-Galerkin and finite element method within numerical homogenisation (2017). [arXiv:1709.08477](https://arxiv.org/abs/1709.08477).

[51] W. Hackbusch, Tensor Spaces and Numerical Tensor Calculus, Springer Science & Business Media, Berlin, Heidelberg, 2012. [doi:10.1007/978-3-540-78862-1](https://doi.org/10.1007/978-3-540-78862-1).

[52] L. De Lathauwer, B. De Moor, J. Vandewalle, A Multilinear Singular Value Decomposition, SIAM Journal on Matrix Analysis and Applications 21 (4) (2000) 1253–1278. [arXiv:34249873259 doi:10.1137/S0895479896305696](https://doi.org/10.1137/S0895479896305696).