Received: 10 February 2024 | Revised: 30 June 2024 | Accepted: 3 July 2024

DOI: 10.1002/nme.7569

RESEARCH ARTICLE

WILEY

FFT-based computational micromechanics with Dirichlet boundary conditions on the rotated staggered grid

Lennart Risthaus[1] | **Matti Schneider**[1,2]

[1]Institute of Engineering Mathematics, University of Duisburg-Essen, Essen, Germany

[2]Fraunhofer Institute for Industrial Mathematics ITWM, Kaiserslautern, Germany

Correspondence
Matti Schneider, Institute of Engineering Mathematics, University of Duisburg-Essen, Essen, Germany.
Email: matti.schneider@uni-due.de

Funding information
European Research Council within the Horizon Europe program, Grant/Award Number: 101040238; Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Grant/Award Number: 418247895

Abstract

Imposing nonperiodic boundary conditions for unit cell analyses may be necessary for a number of reasons in applications, for example, for validation purposes and specific computational setups. The work at hand discusses a strategy for utilizing the powerful technology behind fast Fourier transform (FFT)-based computational micromechanics—initially developed with periodic boundary conditions in mind—for essential boundary conditions in mechanics, as well, for the case of the discretization on a rotated staggered grid. Introduced by F. Willot into the community, the rotated staggered grid is presumably the most popular discretization, and was shown to be equivalent to underintegrated trilinear hexahedral elements. We leverage insights from previous work on the Moulinec–Suquet discretization, exploiting a finite-strain preconditioner for small-strain problems and utilize specific discrete sine and cosine transforms. We demonstrate the computational performance of the novel scheme by dedicated numerical experiments and compare displacement-based methods to implementations on the deformation gradient.

KEYWORDS

Dirichlet boundary conditions, discrete cosine transform, discrete sine transform, FFT-based computational micromechanics, rotated staggered grid

1 | INTRODUCTION

1.1 | State of the art

Driven by the need to understand the mechanical behavior of materials with heterogeneous microstructure, computational techniques based on the fast Fourier transform (FFT) have been established as rather efficient tools for this purpose. In their seminal work, Moulinec-Suquet[1,2] proposed a numerical scheme which operates on regular grids and leverages the Lippmann-Schwinger equation[3–5] pertinent to micromechanics. The original algorithm, called basic scheme, avoided assembling system matrices—as is usual in most finite-element analysis—could handle nonlinear constitutive behavior of the constituents naturally and accounted for the periodicity of the displacement fluctuation field that is favorable for materials with periodic and stochastic microstructure.[6–8]

Subsequent developments in FFT-based computational micromechanics may be roughly classified in three categories. The first category describes extensions of the domain of applicability of the scheme, that is, going beyond the initially

considered quasi-static inelastic setting at small strain. These extensions encompass, for instance, problems at finite strains,[9–11] damage and fracture,[12–14] coupled problems,[15–17] fluid problems,[18–20] or wave propagation.[21,22] The second category of developments concerns faster solution methods. The original basic scheme converged for materials with finite material contrast, and, for fixed tolerance, required an iteration count that was proportional to the material contrast, but turned out to be bounded independently of the mesh spacing. For large material contrast—in particular for nonlinear and inelastic materials—the basic scheme requires a large number of iterations, which may be inappropriate for some applications. As a consequence, faster solvers were developed. These come in essentially three flavors. Polarization methods are based on an ingenious rewriting of the Lippmann–Schwinger equation and were introduced by Eyre–Milton[23] and Michel-Moulinec-Suquet.[24,25] Over the years, a general class of polarization methods was established[26,27] and the relationships of polarization schemes among each other and to optimization methods was uncovered.[28–30] If applicable, polarization methods turn out to be fast and robust solution methods, in particular for strongly nonlinear problems.[31–33] The second flavor of improvements of the basic scheme essentially exploit the interpretation of the basic scheme as a gradient-descent method[34,35] in case the stress operator derives from a stored energy function. Then, accelerated[36,37] and conjugate gradient methods[38–40] may be developed which significantly outperform the basic scheme in the high-contrast scenario. The third flavor involves Newton methods[34,41] and Quasi-Newton[42–44] methods. All three flavors of improved solvers have their merits, and feature in dedicated and popular FFT-based computational micromechanics codes.[45–48]

The third general direction of improvements over Moulinec–Suquet's initial contribution are concerned with the utilized discretization of the cell problem. The original scheme[1,2] naturally emerges as a nonconforming Fourier–Galerkin method,[49] that is, utilizing trigonometric polynomials as the ansatz functions and considering an approximation of the exact stored energy via quadrature by the trapezoidal rule. This type of discretization comes with three characteristics which may be considered disadvantageous. For a start, the solution fields show characteristic ringing artifacts, a consequence of Gibbs' phenomenon for Fourier series approximations. The second disadvantage concerns the violation of the Galerkin framework implied by the use of quadrature. Last but not least, the discretization was exposed to imply numerical ill-conditioning in the presence of pores, i.e., leading to a lack of convergence of all solvers for certain microstructures.[50] To battle the ringing artifacts, finite-difference,[51–53] finite-element,[54–56] and finite-volume[33,57] discretizations on a regular grid were shown to be compatible with the Lippmann–Schwinger framework, some of which introduce different artifacts. The Galerkin framework could be preserved by using either Fourier–Galerkin[58–60] or finite-element methods.[61] The ill-conditioning for infinite contrast could be fixed by using low-order finite difference or finite-element discretizations.[62]

In parallel to the improvements in discretization, the accuracy of computational approaches on regular grids for applications to micromechanics were coming under scrutiny. The voxelized representation of the microstructure precludes resolving the interfaces between different constituent materials, leading to artifacts in the vicinity of the interface for both the strain and the stress field, essentially independent of the discretization used. Early contributions[49,54,63] established quantitative convergence under mesh refinement of the Moulinec–Suquet discretization.[1,2] The seminal article of Ye-Chung[64] provided a number of insights. For a start, they showed quantitative convergence under mesh refinement for the discretization on a rotated staggered grid.[53,65,66] Moreover, they analyzed trilinear finite elements on a regular grid, and provided a qualitative convergence result for the strain field in finite-contrast linear elasticity. Additionally, they showed that the effective stresses converge with twice the rate of the local strain field in this setting, that is, the averaging that is necessary for computing effective properties also improves the convergence rate upon grid refinement.[67] This superconvergence phenomenon was subsequently also shown to hold for a class of nonlinear materials[68] and for the original Moulinec–Suquet discretization.[69] Essentially, it turned out that the Moulinec–Suquet discretization shares the same convergence rate, for both local and effective stresses, as trilinear finite elements on a regular grid.

For more details on the improvements and a broader view we refer to the available review articles.[70–73]

Periodic boundary conditions for the displacement are intrinsically linked to FFT-based computational micromechanics. However, there are applications when different boundary conditions, for example, essential boundary conditions or stress boundary conditions are required. For instance, when reproducing a compression experiment of a microstructured material, essential boundary conditions need to be enforced on two opposing sides of the specimen, and the remaining faces of the cell should be stress-free. For thermal homogenization problems, there is a simple way to impose Dirichlet (zero temperature) or Neumann (zero flux) boundary conditions—by using sine or cosine series, that is, upon discretization, dedicated discrete sine and discrete cosine transforms. Such strategies have been known for a long time (sec. 19.7 of Reference 74) for homogeneous problems, and were recently applied to heterogeneous problems.[75–78] Unfortunately, these approaches run into difficulties for small-strain mechanics. In a nutshell, when applying the differential operator of small-strain elasticity, certain shear terms appear which mix sine and cosine terms, precluding the emergence of

a corresponding Green's operator. Only in case of a specific combination of essential and stress boundary conditions, a Green's operator may be found.[79,80] For this reason, different techniques were exploited. These approaches all share the insight that a field on a rectangular box which vanishes on the faces of the box is a special case of a periodic field. Thus, essential boundary conditions may be imposed by considering periodic problems and interpreting the essential boundary conditions as *additional constraints*. These constraints may either be imposed via a penalty technique, for example, via a "stiff layer,"[81,82] or in terms of suitable Lagrange multipliers.[22,83,84] In any case, these techniques lead to a deterioration of the numerical performance of the solvers. In case of stiff layers, the material contrast is increased significantly, whereas the use of Lagrange multipliers leads to indefinite systems of equations which are, in general, less efficient to solve than symmetric positive definite ones (sec. 5.2 of Reference 85).

1.2 | Contributions

Recently, the authors introduced a workaround for imposing essential boundary conditions in FFT-based computational micromechanics.[86] The fundamental idea is to use a finite-strain framework also for problems at small strains. With this way of solving the problem, the different shear components of the deformation gradient and the Cauchy stress are handled by different discrete sine and cosine series, avoiding the problems when constructing a small-strain preconditioner. With this trick, worked out for the original Moulinec-Suquet discretization,[1,2] the full arsenal of Lippmann–Schwinger solvers was unlocked for essential boundary conditions. The price to pay for this advantage was a slightly increased iteration count (factor 2 for the basic scheme and factor $\sqrt{2}$ for the conjugate gradient method) and a delicate implementation, requiring suitable discrete sine and cosine transforms as well as weighted quadrature.

The work at hand is concerned with extending the Lippmann–Schwinger framework to the case of a finite difference discretization with vanishing essential boundary conditions. The general case for nonzero Dirichlet boundary conditions can be handled by additively decomposing the displacement field into a part which is fixed but satisfies the Dirichlet datum and a fluctuating part with vanishing displacement on the boundary which needs to be determined. We refer to Morin-Paux (Eq. (36) of Reference 77) for an application of this idea in the context of thermal conductivity. In the periodic setting, the introduction of finite-difference and finite-element solvers turned out to be a game changer, as it permitted to solve some problems with infinite material contrast, for example, porous materials or damaging materials, with Lippmann–Schwinger solvers in a robust manner, also reducing the ringing artifacts pertinent to the Moulinec–Suquet discretization. In the work at hand, we focus on the discretization on a rotated staggered grid,[65,66] introduced into FFT-based computational micromechanics by Willot[53] alongside a suitable Eshelby–Green operator. As of today, the rotated staggered grid is allegedly the most popular discretization in FFT-based computational micromechanics. It was shown that the discretization also emerges when considering trilinear finite elements with one-point quadrature in the center of the voxel.[55,87]

To develop Lippmann–Schwinger solvers for finite-difference discretizations and essential boundary conditions, a number of difficulties need to be overcome. For a start, in contrast to the Moulinec–Suquet discretization,[1,2] the displacements and the deformation gradient reside on *different grids*, see Section 2 for details. Even if the idea to use discrete sine and cosine transforms is clear, it may be nontrivial to select the proper transform among the different possibilities. We introduce a systematic technique for detecting the appropriate transform to choose. The essential idea is to change coordinates for the discrete displacement field by representing the discrete field as a suitable sine polynomial. With this trick, the discrete differential operators involved may be written naturally in block diagonal form upon appropriate discrete sine and cosine transforms, see Section 3.2 for details. We also expose a surprisingly simple formula for the Eshelby–Green operator. Unexpectedly, the formula requires a special shift operation that is hard to guess unless a systematic approach is used. In contrast to the Moulinec–Suquet discretization, the discretization on a rotated staggered grid may be implemented on the displacement fluctuation instead of the traditional formulation based on the strain or the deformation gradient. This insight, put to fruition in the periodic setting,[56,61,87] may also be exploited for essential boundary conditions. Interestingly, a displacement-based implementation permits to save two thirds of the memory, as only the three scalar components of the displacement field need to be stored instead of the nine components required for storing the deformation gradient. Moreover, applying the Green's operator for the displacement is conceptually simpler than working with the Eshelby–Green operator for the deformation gradient. The former applies the same transform (the DST-I) to all components of the displacement field and in all spatial directions, whereas the Eshelby–Green operator requires a delicate mix of discrete sine and cosine transforms to be applied to different components of the deformation gradient, alongside an appropriate shift operator. We refer to Section 3.3 for details.

Interestingly, the displacement-based approach to imposing Dirichlet boundary conditions turns out to be faster than the deformation-gradient implementation, but not significantly so. Also, the runtimes are comparable to displacement-based implementations in the periodic setting. The details comprise Section 4.

We wish to emphasize that our strategy is to reduce the mechanical cell problem to a (vectorized version of the) thermal conductivity problem. For the discretization on a rotated staggered grid, the thermal conductivity case was discussed by Gélébart[78] in a comprehensive way, covering a multitude of applicable boundary conditions. Gélebart uses symmetry and antisymmetry arguments to uncover the proper transforms and "Fourier" coefficients. In the paper at hand, we develop a different, but equivalent, strategy to treat Dirichlet boundary conditions for thermal and mechanical homogenization problems based on a first-principle approach based on sine series. Unfortunately, applying symmetry ideas to small-strain mechanics leads to a set of boundary conditions[80] different from pure Dirichlet boundary conditions. However, the strategy[78] to devise Green's operator for other boundary conditions from the periodic Green's operator may be investigated in more detail.

2 | CELL PROBLEMS WITH ESSENTIAL BOUNDARY CONDITIONS

2.1 | The continuous setting

We are concerned with a problem on the unit cell

$$Y = [0, L_1] \times [0, L_2] \times [0, L_3], \tag{1}$$

and we suppose that a (heterogeneous and potentially nonlinear) stress function

$$\boldsymbol{S} : Y \times \mathbb{R}^{3\times 3} \to \mathbb{R}^{3\times 3}, \quad (\boldsymbol{x}, \boldsymbol{F}) \mapsto \boldsymbol{S}(\boldsymbol{x}, \boldsymbol{F}), \tag{2}$$

is given, which encodes the stress response to an applied deformation $\boldsymbol{F} \in \mathbb{R}^{3\times 3}$ at a particular point $\boldsymbol{x} \in Y$ in the cell (1). We actually work at small strains, but choose the finite-strain formulation (2) with the computational resolution in mind, compare Risthaus-Schneider[86] or Section 3 below for more details. Working at small strains entails the condition

$$\boldsymbol{S}(\boldsymbol{x}, \boldsymbol{F}) = \boldsymbol{0} \quad \text{for all} \quad \boldsymbol{x} \in Y \quad \text{and any skew-symmetric} \quad \boldsymbol{F} \in \mathbb{R}^{3\times 3}. \tag{3}$$

Moreover, we suppose that the stress function (2) satisfies both a Lipschitz and a strong monotonicity condition on the symmetric part of the deformation gradient uniformly in the unit cell (sec. 2.1 of Reference 86).

For a prescribed average strain

$$\overline{\boldsymbol{\varepsilon}} \in \mathrm{Sym}(3) \equiv \left\{ \boldsymbol{F} \in \mathbb{R}^{3\times 3} \;\middle|\; \boldsymbol{F}^T = \boldsymbol{F} \right\}, \tag{4}$$

we seek the displacement fluctuation

$$\boldsymbol{u} \in H^1_0(Y; \mathbb{R}^3) \equiv \left\{ \boldsymbol{u} \in H^1(Y; \mathbb{R}^3) \;\middle|\; \boldsymbol{u} \equiv \boldsymbol{0} \quad \text{on} \quad \partial Y \right\}, \tag{5}$$

vanishing on the boundary ∂Y of the cell (1), which satisfies the equilibrium equation

$$\mathrm{div}\, \boldsymbol{S}(\cdot, \overline{\boldsymbol{\varepsilon}} + \nabla \boldsymbol{u}) = \boldsymbol{0}. \tag{6}$$

With a finite-element discretization in mind, the latter equation (6) is typically rephrased in *weak form*

$$\int_Y \langle \nabla \boldsymbol{v}, \boldsymbol{S}(\cdot, \overline{\boldsymbol{\varepsilon}} + \nabla \boldsymbol{u}) \rangle \, d\boldsymbol{x} = 0 \quad \text{for all} \quad \boldsymbol{v} \in H^1_0(Y; \mathbb{R}^3), \tag{7}$$

involving the Frobenius inner product

$$\langle \cdot, \cdot \rangle : \mathbb{R}^{3\times 3} \times \mathbb{R}^{3\times 3} \to \mathbb{R}, \quad \langle \boldsymbol{F}_1, \boldsymbol{F}_2 \rangle \equiv \operatorname{tr}(\boldsymbol{F}_2^T \boldsymbol{F}_1). \tag{8}$$

Under suitable assumptions (sec. 2.1 of Reference 86), existence and uniqueness of solutions to the small-strain problem (6) without inertial forces may be established (sec. 2.1 of Reference 86). Subsequently, the averaged stress

$$\overline{\boldsymbol{\sigma}} \equiv \fint_Y \boldsymbol{\mathcal{S}}(\cdot, \overline{\boldsymbol{\varepsilon}} + \nabla \boldsymbol{u}) \, d\boldsymbol{x}, \tag{9}$$

is the primary quantity of interest.

2.2 | The rotated staggered grid

The discretization on a rotated staggered grid[65,66] is a finite-difference discretization on a regular grid. It was introduced to FFT-based computational micromechanics in the periodic setting by Willot.[53] More precisely, he presented an accompanying Lippmann–Schwinger solution framework which preserves the original computational prowess of Moulinec–Suquet's "basic scheme,"[1,2] but fixes some shortcomings of the original scheme. Subsequently, the discretization found its way into a multitude of implementations and became one of the most popular frameworks used in FFT-based computational micromechanics, see also the review articles.[70–72] In addition to Willot's original article,[53] the discretization emerged also as an underintegrated finite-element integration on a regular grid[55] and a nonconforming $\mathrm{P1}$-finite element discretization.[88] To keep things simple, we stick with the interpretation as a finite-difference method.

We consider $N_i > 0$ voxels in each coordinate direction ($i = 1, 2, 3$) and take a look at the nodal grid

$$Y_{\boldsymbol{N}}^{\mathrm{nod}} = \left\{ \boldsymbol{x} \in Y \;\middle|\; x_i = \frac{j_i L_i}{N_i} \quad \text{for some} \quad j_i \in \{0, 1, \dots, N_i\} \quad \text{and all} \quad i = 1, 2, 3 \right\}, \tag{10}$$

with the abbreviation $\boldsymbol{N} \equiv (N_1, N_2, N_2)$ and the quadrature grid

$$Y_{\boldsymbol{N}}^{\mathrm{qu}} = \left\{ \boldsymbol{x} \in Y \;\middle|\; x_i = \left(j_i + \frac{1}{2}\right) \frac{L_i}{N_i} \quad \text{for some} \quad j_i \in \{0, 1, \dots, N_i - 1\} \quad \text{and all} \quad i = 1, 2, 3 \right\}. \tag{11}$$

The nodal grid (10) concerns degrees of freedom which live on the corners of the voxels and comprises $(N_1 + 1)(N_2 + 1)(N_3 + 1)$ points, see Figure 1A. In contrast, the quadrature grid (11) is located at the *voxel centers*, see Figure 1A, and consists of $N_1 N_2 N_3$ points, the same as the number of voxels in the discretized microstructure.

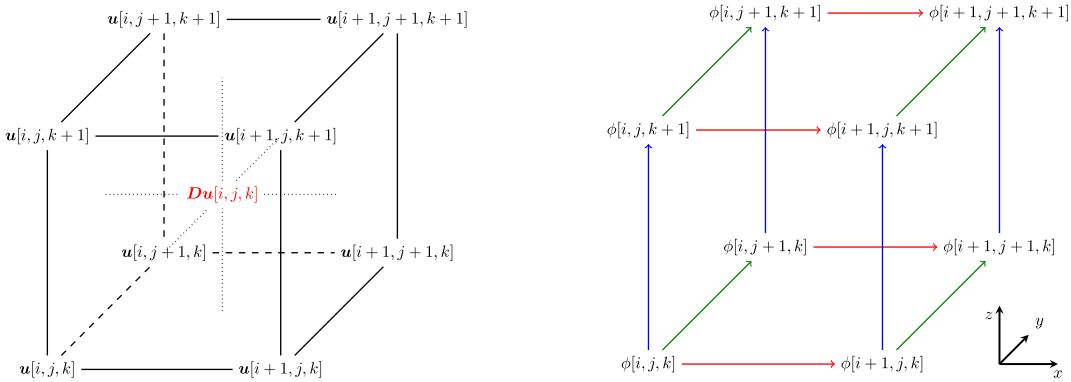


FIGURE 1 Illustration of the grids and the "derivative" operators for the rotated staggered grid. (A) Illustration of the displacement grid (10), in black, and the quadrature grid (11), in red, for each voxel. (B) Illustration of the finite difference operators D_1, D_2, and D_3 defined in Equations (15)–(17).

We define the discrete boundary of the nodal grid (10)

$$\partial Y_{\boldsymbol{N}}^{\mathrm{nod}} = \left\{ \boldsymbol{x} \in Y_{\boldsymbol{N}}^{\mathrm{nod}} \;\middle|\; x_i = 0 \quad \text{or} \quad x_i = L_i \quad \text{for some} \quad i = 1,2,3 \right\}. \tag{12}$$

The discretization on a rotated staggered grid is concerned with discrete displacements in the space

$$\boldsymbol{H}_{\boldsymbol{N}}^{3} = \left\{ \boldsymbol{u} : Y_{\boldsymbol{N}}^{\mathrm{nod}} \to \mathbb{R}^3 \;\middle|\; \boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{0} \quad \text{for all} \quad \boldsymbol{x} \in \partial Y_{\boldsymbol{N}}^{\mathrm{nod}} \right\}, \tag{13}$$

with vanishing boundary datum, and we consider the space

$$\boldsymbol{V}_{\boldsymbol{N}}^{3\times 3} = \left\{ \boldsymbol{T} : Y_{\boldsymbol{N}}^{\mathrm{qu}} \to \mathbb{R}^{3\times 3} \right\}, \tag{14}$$

to contain the deformation gradient and stress fields.

For any nodal field $\phi : Y_{\boldsymbol{N}}^{\mathrm{nod}} \to \mathbb{R}$, we define the partial difference operators D_i $(i = 1,2,3)$ to produce fields on the quadrature grid (11) via

$$\begin{aligned} D_1\phi[j_1,j_2,j_3] = \frac{1}{4h_1}(&\phi[j_1+1,j_2,j_3] - \phi[j_1,j_2,j_3] \\ &+ \phi[j_1+1,j_2+1,j_3] - \phi[j_1,j_2+1,j_3] \\ &+ \phi[j_1+1,j_2,j_3+1] - \phi[j_1,j_2,j_3+1] \\ &+ \phi[j_1+1,j_2+1,j_3+1] - \phi[j_1,j_2+1,j_3+1]), \end{aligned} \tag{15}$$

as well as

$$\begin{aligned} D_2\phi[j_1,j_2,j_3] = \frac{1}{4h_2}(&\phi[j_1,j_2+1,j_3] - \phi[j_1,j_2,j_3] \\ &+ \phi[j_1+1,j_2+1,j_3] - \phi[j_1+1,j_2,j_3] \\ &+ \phi[j_1,j_2+1,j_3+1] - \phi[j_1,j_2,j_3+1] \\ &+ \phi[j_1+1,j_2+1,j_3+1] - \phi[j_1+1,j_2,j_3+1]), \end{aligned} \tag{16}$$

and

$$\begin{aligned} D_3\phi[j_1,j_2,j_3] = \frac{1}{4h_3}(&\phi[j_1,j_2,j_3+1] - \phi[j_1,j_2,j_3] \\ &+ \phi[j_1+1,j_2,j_3+1] - \phi[j_1+1,j_2,j_3] \\ &+ \phi[j_1,j_2+1,j_3+1] - \phi[j_1,j_2+1,j_3] \\ &+ \phi[j_1+1,j_2+1,j_3+1] - \phi[j_1+1,j_2+1,j_3]). \end{aligned} \tag{17}$$

To keep these formulas reasonably short, we rely on the shorthand notations

$$D_k\phi[j_1,j_2,j_3] \equiv D_k\phi\left(\left(j_1+\frac{1}{2}\right)h_1, \left(j_2+\frac{1}{2}\right)h_2, \left(j_3+\frac{1}{2}\right)h_3\right), \quad k = 1,2,3, \tag{18}$$

with $h_k = L_k/N_k$ $(k = 1,2,3)$ and

$$\phi[j_1,j_2,j_3] \equiv \phi(j_1h_1, j_2h_2, j_3h_3). \tag{19}$$

The finite-difference operators (15)–(17) average the corresponding (forward) finite-differences on the edges of the voxel under consideration and pointing in the direction of differentiation, see Figure 1B.

With the operators (15)–(17) at hand, we define the discrete deformation gradient

$$\boldsymbol{D} : \boldsymbol{H}_{\boldsymbol{N}}^{3} \to \boldsymbol{V}_{\boldsymbol{N}}^{3\times 3}, \tag{20}$$

via the usual rule

$$\boldsymbol{Du} = \begin{bmatrix} D_1u_1 & D_1u_2 & D_1u_3 \\ D_2u_1 & D_2u_2 & D_2u_3 \\ D_3u_1 & D_3u_2 & D_3u_3 \end{bmatrix} \quad \text{for a field} \quad \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}. \tag{21}$$

The discrete equivalent of the equilibrium equation for the discretization on a rotated staggered grid is to find a displacement-fluctuation field $\boldsymbol{u} \in \boldsymbol{H}_{\boldsymbol{N}}^3$, s.t. the discrete weak form

$$\sum_{\boldsymbol{x} \in Y_{\boldsymbol{N}}^{\mathrm{qu}}} \langle \boldsymbol{Dv}(\boldsymbol{x}), \boldsymbol{S}(\boldsymbol{x}, \overline{\boldsymbol{\varepsilon}} + \boldsymbol{Du}(\boldsymbol{x})) \rangle = 0, \tag{22}$$

is satisfied for all $\boldsymbol{v} \in \boldsymbol{H}_{\boldsymbol{N}}^3$. After solving this Equation (22), the discrete average stress is computed by computing the mean of the stress field on the quadrature grid (11)

$$\overline{\boldsymbol{\sigma}}_{\boldsymbol{N}} \equiv \frac{1}{N_1 N_2 N_3} \sum_{\boldsymbol{x} \in Y_{\boldsymbol{N}}^{\mathrm{qu}}} \boldsymbol{S}(\boldsymbol{x}, \overline{\boldsymbol{\varepsilon}} + \boldsymbol{Du}(\boldsymbol{x})). \tag{23}$$

3 | SOLVERS OF LIPPMANN–SCHWINGER TYPE

3.1 | General framework

Lippmann–Schwinger solvers may be developed for problems of the form (sec. 3.1 of Reference 76)

$$\boldsymbol{D}^* \boldsymbol{S}(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{Du}) = \boldsymbol{0}. \tag{24}$$

Here, we suppose that two Hilbert spaces $\boldsymbol{H}$ and $\boldsymbol{V}$ are given, together with a bounded linear operator

$$\boldsymbol{D} \in L(\boldsymbol{H}, \boldsymbol{V}), \tag{25}$$

s.t. the bilinear form

$$\boldsymbol{H} \times \boldsymbol{H} \to \mathbb{R}, \quad (\boldsymbol{u}_1, \boldsymbol{u}_2) \mapsto \langle \boldsymbol{Du}_1, \boldsymbol{Du}_2 \rangle_{\boldsymbol{V}}, \tag{26}$$

defines an (equivalent) inner product on the Hilbert space $\boldsymbol{H}$. In the expression (26), the angular brackets $\langle \cdot, \cdot \rangle_{\boldsymbol{V}}$ encode the inner product on the Hilbert space $\boldsymbol{V}$.

With these data at hand, the "divergence operator" (up to a minus sign) $\boldsymbol{D}^*$ is defined as the (mixed) adjoint of the linear operator (25)

$$\boldsymbol{D}^* \in L(\boldsymbol{V}, \boldsymbol{H}'), \quad \boldsymbol{D}^* \boldsymbol{\tau}[\boldsymbol{v}] = \langle \boldsymbol{\tau}, \boldsymbol{Dv} \rangle_{\boldsymbol{V}} \quad \text{for} \quad \boldsymbol{\tau} \in \boldsymbol{V}, \quad \boldsymbol{v} \in \boldsymbol{H}, \tag{27}$$

where $\boldsymbol{H}'$ denotes the continuous dual space of the Hilbert space $\boldsymbol{H}$, that is, the space of continuous linear functionals on $\boldsymbol{H}$, and the rectangular brackets stand for the evaluation of a linear functional on a vector.

To make sense of Equation (24) we moreover suppose that a (possibly nonlinear) mapping

$$\boldsymbol{S} : \boldsymbol{V} \to \boldsymbol{V}, \tag{28}$$

is given, together with an element $\overline{\boldsymbol{\varepsilon}} \in \boldsymbol{V}$ with vanishing divergence, that is, which satisfies the condition

$$\boldsymbol{D}^* \overline{\boldsymbol{\varepsilon}} = \boldsymbol{0}. \tag{29}$$

Notice that the definition (27) of the adjoint operator $\boldsymbol{D}^*$ enables us to rewrite the original equation in "weak form"

$$\langle \boldsymbol{D}\boldsymbol{v}, \boldsymbol{S}(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u})\rangle_{\boldsymbol{V}} = 0 \quad \text{for all} \quad \boldsymbol{v} \in \boldsymbol{H}. \tag{30}$$

Standard arguments, see Appendix A.1, permit to demonstrate that for any fixed constant $\alpha_0 > 0$, the original problem (24) is equivalent to the Lippmann–Schwinger equation

$$\boldsymbol{u} = -\boldsymbol{G}^0\boldsymbol{D}^*\big(\boldsymbol{S}(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u}) - \alpha_0(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u})\big), \tag{31}$$

where Green's operator $\boldsymbol{G}^0$ is the bounded linear operator

$$\boldsymbol{G}^0 \in L(\boldsymbol{H}', \boldsymbol{H}), \quad \boldsymbol{G}^0 = \frac{1}{\alpha_0}(\boldsymbol{D}^*\boldsymbol{D})^{-1}. \tag{32}$$

The fixed-point scheme associated to the displacement-based Lippmann–Schwinger equation (31) with an initial guess $\boldsymbol{u}^0 \in \boldsymbol{H}$ reads

$$\boldsymbol{u}^{k+1} = -\boldsymbol{G}^0\boldsymbol{D}^*\big(\boldsymbol{S}(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u}^k) - \alpha_0(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u}^k)\big), \quad k = 0, 1, \ldots, \tag{33}$$

where the constant α_0 needs to be chosen properly to ensure convergence. In a concrete implementation, the action of Green's operator is evaluated in Fourier space. The operators $\boldsymbol{D}$ and $\boldsymbol{D}^*$ may also be evaluated in Fourier space, but the formulation (33) precludes these evaluations to be made in an efficient way, essentially because the action of the operator $\boldsymbol{S}$—which acts in real space—interferes with such a procedure. To avoid these difficulties, traditional FFT-based methods operate on the space $\boldsymbol{V}$. In fact, introducing the total deformation gradient

$$\boldsymbol{F} = \overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u}, \tag{34}$$

it is readily apparent that the deformation gradient (34) associated to a solution to the Lippmann–Schwinger equation (31) (or, equivalently, to the equilibrium problem (24)) solves the Lippmann–Schwinger equation

$$\boldsymbol{F} = \overline{\boldsymbol{\varepsilon}} - \boldsymbol{\Gamma}^0(\boldsymbol{S}(\boldsymbol{F}) - \alpha_0\, \boldsymbol{F}), \tag{35}$$

with the Eshelby–Green operator

$$\boldsymbol{\Gamma}^0 = \boldsymbol{D}\boldsymbol{G}^0\boldsymbol{D}^* \equiv \frac{1}{\alpha_0}\boldsymbol{D}(\boldsymbol{D}^*\boldsymbol{D})^{-1}\boldsymbol{D}^*. \tag{36}$$

Conversely, any field $\boldsymbol{F} \in \boldsymbol{V}$ solving the equation (35) also gives rise to a solution to the displacement-based Lippmann–Schwinger equation (31) via the rule

$$\boldsymbol{u} = -\boldsymbol{G}^0\boldsymbol{D}(\boldsymbol{S}(\boldsymbol{F}) - \alpha_0\, \boldsymbol{F}). \tag{37}$$

For the convenience of the reader, the arguments are collected in Appendix A.2.

In view of Equation (36), it is readily apparent that the action of the operators $\boldsymbol{D}$, $\boldsymbol{G}^0$ and $\boldsymbol{D}^*$ may be evaluated in a single run for Equation (35), avoiding the computational overhead involved in evaluating the scheme (33). Based on the Lippmann-Schwinger equation (35) and for any initial guess $\boldsymbol{F}^0 \in \boldsymbol{V}$, the associated fixed point iterative scheme

$$\boldsymbol{F}^{k+1} = \overline{\boldsymbol{\varepsilon}} - \boldsymbol{\Gamma}^0\big(\boldsymbol{S}(\boldsymbol{F}^k) - \alpha_0\, \boldsymbol{F}^k\big), \quad k = 0, 1, \ldots, \tag{38}$$

is called basic scheme.[1,2] Thus, we have two completely equivalent fixed-point schemes at out disposal, a displacement-based version (33) and a method involving the deformation gradient (38). Due to their equivalence, the iterates coincide provided the initial condition and the parameter α_0 are chosen identically. Rather, the *implementations* differ only in their computational expense and memory footprint.

To discuss the choice of the step size α_0, we stick to the deformation-gradient version (38) and suppose that there are positive constants $\alpha_\pm$, s.t. the Lipschitz estimate

$$\|\boldsymbol{S}(\boldsymbol{F}_1) - \boldsymbol{S}(\boldsymbol{F}_2)\|_{\boldsymbol{V}} \leq \alpha_+ \, \|\boldsymbol{F}_1 - \boldsymbol{F}_2\|_{\boldsymbol{V}}, \quad \boldsymbol{F}_1, \boldsymbol{F}_2 \in \boldsymbol{V}, \tag{39}$$

holds and the strong monotonicity condition

$$\langle \boldsymbol{S}(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u}_1) - \boldsymbol{S}(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u}_2), \boldsymbol{D}\boldsymbol{u}_1 - \boldsymbol{D}\boldsymbol{u}_2 \rangle_{\boldsymbol{V}} \geq \alpha_- \, \|\boldsymbol{D}\boldsymbol{u}_1 - \boldsymbol{D}\boldsymbol{u}_2\|_{\boldsymbol{V}}^2 \tag{40}$$

is satisfied for all $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \boldsymbol{H}$ and $\overline{\boldsymbol{\varepsilon}} \in \boldsymbol{V}$ from above. Then, the equilibrium problem (24) has a unique solution $\boldsymbol{u}^*$. If we furthermore assume that the stress operator (28) arises as the differential of a Fréchet differentiable function $\psi : \boldsymbol{H} \to \mathbb{R}$, the basic scheme satisfies the convergence estimate

$$\|\boldsymbol{F}^{k+1} - \boldsymbol{F}^*\|_{\boldsymbol{V}} \leq \rho(\alpha_0) \, \|\boldsymbol{F}^k - \boldsymbol{F}^*\|_{\boldsymbol{V}}, \quad \rho(\alpha_0) = \frac{\max(|\alpha_+ - \alpha_0|, |\alpha_- - \alpha_0|)}{\alpha_0}, \tag{41}$$

for the solution

$$\boldsymbol{F}^* = \overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u}^*, \tag{42}$$

of the Lippmann–Schwinger equation (35). The bound ρ is minimized for the value

$$\alpha_0 = \frac{\alpha_- + \alpha_+}{2}, \quad \text{s.t.} \quad \rho(\alpha_0) = \frac{\alpha_+ - \alpha_-}{\alpha_+ + \alpha_-}. \tag{43}$$

This result follows from standard optimization arguments (Thm. 2.1.15 of Reference 89), exploiting the connection of the basic scheme to gradient descent,[34] see Schneider (sec. 2, Appx. A of Reference 62).

With this framework at hand, the entire zoo of Lippmann–Schwinger solvers becomes available, including fast and conjugate gradient methods,[35,36,38–40] Newton-type methods[34,37,41–43] and polarization schemes.[23,25,26,29,30] We refer to the review article[72] for more details.

3.2 | Solving the reference problem with discrete sine and cosine series

We would like to embed the discretization on a rotated staggered grid, discussed in Section 2.2, into the framework established in the previous Section 3.1. Therefore, we prescribe the following:

- The Hilbert spaces $\boldsymbol{H}$ and $\boldsymbol{V}$ are given by the finite-dimensional spaces $\boldsymbol{H}_{\boldsymbol{N}}^3$ and $\boldsymbol{V}_{\boldsymbol{N}}^{3\times3}$ defined in the Equations (13) and (14).
- The bounded linear operator $\boldsymbol{D}$ in Equation (25) is taken to be the finite difference gradient (21).
- The stress operator $\boldsymbol{S}$ of Equation (28) is just the point-wise evaluation of the stress function (2), that is,

$$\boldsymbol{S}(\boldsymbol{\varepsilon})(\boldsymbol{x}) = \mathcal{S}(\boldsymbol{x}, \boldsymbol{\varepsilon}(\boldsymbol{x})) \quad \text{for} \quad \boldsymbol{\varepsilon} \in \boldsymbol{V}_{\boldsymbol{N}}^{3\times3} \quad \text{and} \quad \boldsymbol{x} \in Y_{\boldsymbol{N}}^{\mathrm{qu}}. \tag{44}$$

- The abstract element $\overline{\boldsymbol{\varepsilon}} \in \boldsymbol{V}$ in Equation (29) is taken to be a strain $\overline{\boldsymbol{\varepsilon}} \in \mathrm{Sym}(3)$ (4), considered as a constant field on the quadrature grid (11).

For these choices, it becomes clear that the general equilibrium equation (24)—in weak form (30)—recovers the equilibrium equation (22) for the rotated staggered grid discretization. In particular, the Lippmann–Schwinger framework becomes available under the condition that Green's operator (32) can be computed readily. Thus, the purpose of this section is to introduce an efficient technique for computing Green's operator (and the Eshelby–Green operator (36)).

The starting point is the use of sine polynomials (sec. 3.1 of Reference 86), that is, functions of the form

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_{\mathbf{k} \in \mathbb{N}_{\boldsymbol{N}}^3} \hat{\boldsymbol{u}}(\mathbf{k})\ \mathtt{sss}(\mathbf{k}, \boldsymbol{x}), \quad \boldsymbol{x} \in Y, \tag{45}$$

for the frequencies

$$\mathbb{N}_{\boldsymbol{N}}^3 = \{1, 2, \dots, N_1 - 1\} \times \{1, 2, \dots, N_2 - 1\} \times \{1, 2, \dots, N_3 - 1\}, \tag{46}$$

and the shorthand notation

$$\mathtt{sss}(\mathbf{k}, \boldsymbol{x}) = \sin\left(\pi\ \frac{k_1 x_1}{L_1}\right) \sin\left(\pi\ \frac{k_2 x_2}{L_2}\right) \sin\left(\pi\ \frac{k_3 x_3}{L_3}\right) \quad \text{for} \quad \mathbf{k} = (k_1, k_2, k_3) \in \mathbb{N}^3, \quad \boldsymbol{x} \in Y. \tag{47}$$

We will write $\mathtt{S}_N(Y; \mathbb{R}^3)$ for the space of such sine polynomials (45). The mapping which evaluates a sine polynomial (45) on the nodal grid (10),

$$\mathrm{ev} : \mathtt{S}_N(Y; \mathbb{R}^3) \to \boldsymbol{H}_{\boldsymbol{N}}^3, \quad (\mathrm{ev}(\boldsymbol{u}))(\boldsymbol{x}) = \boldsymbol{u}(\boldsymbol{x}), \quad \boldsymbol{u} \in \mathtt{S}_N(Y; \mathbb{R}^3), \quad \boldsymbol{x} \in Y_{\boldsymbol{N}}^{\mathtt{nod}}, \tag{48}$$

is well-defined, because any sine polynomial (45) vanishes on the boundary of the cell Y, in particular on the boundary of the nodal grid (12). Written out explicitly,

$$\boldsymbol{u}\left(\frac{j_1 L_1}{N_1}, \frac{j_2 L_2}{N_2}, \frac{j_3 L_3}{N_3}\right) = \sum_{\mathbf{k} \in \mathbb{N}_{\boldsymbol{N}}^3} \hat{\boldsymbol{u}}(\mathbf{k})\ \sin\left(\pi\ \frac{k_1 j_1}{N_1}\right) \sin\left(\pi\ \frac{k_2 j_2}{N_2}\right) \sin\left(\pi\ \frac{k_3 j_3}{N_3}\right), \quad \mathbf{j} \in \mathbb{N}_{\boldsymbol{N}}^3, \tag{49}$$

the mapping (47) turns out to be nothing but the discrete sine transform (`DST-I`),[90] performed along each of the three coordinate directions. It is well-known[90] that the `DST-I` is invertible, and we obtain the explicit expression for the sine coefficients

$$\hat{\boldsymbol{u}}(\mathbf{k}) = \frac{1}{8 N_1 N_2 N_3} \sum_{\mathbf{j} \in \mathbb{N}_{\boldsymbol{N}}^3} \boldsymbol{u}\left(\frac{j_1 L_1}{N_1}, \frac{j_2 L_2}{N_2}, \frac{j_3 L_3}{N_3}\right)\ \sin\left(\pi\ \frac{k_1 j_1}{N_1}\right) \sin\left(\pi\ \frac{k_2 j_2}{N_2}\right) \sin\left(\pi\ \frac{k_3 j_3}{N_3}\right), \quad \mathbf{j} \in \mathbb{N}_{\boldsymbol{N}}^3. \tag{50}$$

Thus, the mapping (48) provides a one-to-one correspondence between sine polynomials (45) and discrete displacements (13) on the rotated staggered grid, that is, it manifests as a particular change of coordinates.

We will use this correspondence to express the interesting operators, like the gradient operator $\boldsymbol{D}$, in terms of the frequencies $\{\hat{\boldsymbol{u}}(\mathbf{k})\}_{\mathbf{k} \in \mathbb{N}_{\boldsymbol{N}}^3}$. As a preliminary step, we reduce the complexity by considering the one-dimensional case first. We suppose that the interval $[0, L]$ contains the $N \in \mathbb{N}_{>0}$ grid points

$$x_j = \frac{j + \frac{1}{2}}{N}\ L, \quad j = 0, 1, \dots, N - 1, \tag{51}$$

with mesh spacing $h = L/N$ and where the points are centered on the grid. Suppose that the one-dimensional sine polynomial

$$u(x) = \sum_{k=1}^{N-1} \hat{u}_k \sin\left(\pi \frac{kx}{L}\right), \quad x \in [0, L], \tag{52}$$

is given with sine coefficients $\hat{u}_k$ ($k = 1, 2, \dots, N - 1$). Evaluating the central finite difference δ^{c} on the grid (51) yields

$$\delta^{\mathrm{c}} u(x_j) \equiv \frac{u\left(\frac{j+1}{N}\ L\right) - u\left(\frac{j}{N}\ L\right)}{h} = \frac{1}{h} \sum_{k=1}^{N-1} \hat{u}_k \left[\sin\left(\pi \frac{(j+1)k}{N}\right) - \sin\left(\pi \frac{jk}{N}\right)\right]. \tag{53}$$

By using a suitable trigonometric identity, we may write

$$\delta^{\mathrm{c}} u(x_j) = \sum_{k=1}^{N-1} \frac{2\sin\left(\frac{k\pi}{2N}\right)}{h}\, \hat{u}_k \cos\left(\pi \frac{\left(j+\frac{1}{2}\right)k}{N}\right). \tag{54}$$

The derivations on the trigonometry are collected in Appendix B. The details on the finite differences comprise Appendix B.1. Formula (54) states that we may express the central difference (53) in terms of a cosine transform (DCT-III) of suitably weighted sine coefficients (45), that is,

$$\delta^{\mathrm{c}} u(x_j) = \sum_{k=1}^{N-1} \widehat{\delta^{\mathrm{c}} u}_k \cos\left(\pi \frac{\left(j+\frac{1}{2}\right)k}{N}\right) \quad \text{with} \quad \widehat{\delta^{\mathrm{c}} u}_k = \frac{2\sin\left(\frac{k\pi}{2N}\right)}{h}\, \hat{u}_k \quad (k = 1, 2, \dots, N-1). \tag{55}$$

In addition to the central difference (53), we also consider the average A^{c} of two adjacent values of the sine polynomial (52)

$$A^{\mathrm{c}} u(x_j) \equiv \frac{u\left(\frac{j+1}{N} L\right) + u\left(\frac{j}{N} L\right)}{2} = \frac{1}{2} \sum_{k=1}^{N-1} \hat{u}_k \left[\sin\left(\pi \frac{(j+1)k}{N}\right) + \sin\left(\pi \frac{jk}{N}\right)\right]. \tag{56}$$

By a similar argument as for the difference case (54), we may write the average (56) in the following form

$$A^{\mathrm{c}} u(x_j) = \sum_{k=1}^{N-1} \cos\left(\frac{k\pi}{2N}\right)\, \hat{u}_k \sin\left(\pi \frac{\left(j+\frac{1}{2}\right)k}{N}\right), \tag{57}$$

see Appendix B.2. Thus, the average (56) may be expressed in terms of a suitable sine transform (DST-III) with properly weighted sine coefficients (52)

$$A^{\mathrm{c}} u(x_j) = \sum_{k=1}^{N-1} \widehat{A^{\mathrm{c}} u}_k \sin\left(\pi \frac{\left(j+\frac{1}{2}\right)k}{N}\right), \quad \text{where} \quad \widehat{A^{\mathrm{c}} u}_k = \cos\left(\frac{k\pi}{2N}\right)\, \hat{u}_k \quad (k = 1, 2, \dots, N-1). \tag{58}$$

After these one-dimensional preliminaries, we return our attention to the three-dimensional case. With an eye toward the finite-difference deformation gradient (21), we are interested in the finite-difference operators D_i $(i = 1, 2, 3)$, defined in Equations (15)–(17). In fact, we may express these operators by applying central differences and averages in suitable coordinate directions independently

$$D_1 = \delta_1^{\mathrm{c}} \otimes A_2^{\mathrm{c}} \otimes A_3^{\mathrm{c}}, \quad D_2 = A_1^{\mathrm{c}} \otimes \delta_2^{\mathrm{c}} \otimes A_3^{\mathrm{c}} \quad \text{and} \quad D_3 = A_1^{\mathrm{c}} \otimes A_2^{\mathrm{c}} \otimes \delta_3^{\mathrm{c}}. \tag{59}$$

Here δ_i $(i = 1, 2, 3)$ refers to the forward difference (53) in the ith coordinate direction, that is, using $N = N_i$ and $h = h_i$, whereas A_i^{c} $(i = 1, 2, 3)$ stands for averaging (56) in the ith coordinate direction for $N = N_i$. To make sense of the notation in Equation (59), we focus on the operator D_1. The finite-difference operator (59) may be considered as the composition of three operations. First, the finite-difference operator (53) is applied in x-direction. Subsequently, the result is averaged both in y- and in z-direction. The operators D_2 and D_3 may be evaluated with a similar logic, interchanging the direction of the applied finite differences appropriately. With these considerations at hand, we are led to the expressions

$$D_1 \boldsymbol{u}[\mathbf{j}] = \sum_{\mathbf{k} \in \mathbb{N}_N^3} \xi_1(\mathbf{k}) \hat{\boldsymbol{u}}(\mathbf{k})\, \cos\left(\pi \frac{\left(j_1+\frac{1}{2}\right)k_1}{N_1}\right) \sin\left(\pi \frac{\left(j_2+\frac{1}{2}\right)k_2}{N_2}\right) \sin\left(\pi \frac{\left(j_3+\frac{1}{2}\right)k_3}{N_2}\right), \tag{60}$$

$$D_2\boldsymbol{u}[\mathbf{j}] = \sum_{\mathbf{k}\in\mathbb{N}_{\boldsymbol{N}}^3} \xi_2(\mathbf{k})\hat{\boldsymbol{u}}(\mathbf{k}) \ \sin\left(\pi\frac{\left(j_1+\frac{1}{2}\right)k_1}{N_1}\right)\cos\left(\pi\frac{\left(j_2+\frac{1}{2}\right)k_2}{N_2}\right)\sin\left(\pi\frac{\left(j_3+\frac{1}{2}\right)k_3}{N_2}\right), \tag{61}$$

$$D_3\boldsymbol{u}[\mathbf{j}] = \sum_{\mathbf{k}\in\mathbb{N}_{\boldsymbol{N}}^3} \xi_3(\mathbf{k})\hat{\boldsymbol{u}}(\mathbf{k}) \ \sin\left(\pi\frac{\left(j_1+\frac{1}{2}\right)k_1}{N_1}\right)\sin\left(\pi\frac{\left(j_2+\frac{1}{2}\right)k_2}{N_2}\right)\cos\left(\pi\frac{\left(j_3+\frac{1}{2}\right)k_3}{N_2}\right), \tag{62}$$

for any index $\mathbf{j} \in \mathbb{N}_{\boldsymbol{N}}^3$ with the vector

$$\begin{aligned}
\xi_1(\mathbf{k}) &= \frac{2N_1}{L_1}\sin\left(\frac{k_1\pi}{2N_1}\right)\cos\left(\frac{k_2\pi}{2N_2}\right)\cos\left(\frac{k_3\pi}{2N_3}\right),\\
\xi_2(\mathbf{k}) &= \frac{2N_2}{L_2}\cos\left(\frac{k_1\pi}{2N_1}\right)\sin\left(\frac{k_2\pi}{2N_2}\right)\cos\left(\frac{k_3\pi}{2N_3}\right),\\
\xi_3(\mathbf{k}) &= \frac{2N_3}{L_2}\cos\left(\frac{k_1\pi}{2N_1}\right)\cos\left(\frac{k_2\pi}{2N_2}\right)\sin\left(\frac{k_3\pi}{2N_3}\right)
\end{aligned} \tag{63}$$

for $\mathbf{k} \in \mathbb{N}_{\boldsymbol{N}}^3$ and where we used the shorthand notation (18)

$$D_i\boldsymbol{u}[\mathbf{j}] \equiv D_i\boldsymbol{u}\left(\left(j_1+\frac{1}{2}\right)\frac{L_1}{N_1}, \left(j_2+\frac{1}{2}\right)\frac{L_2}{N_2}, \left(j_3+\frac{1}{2}\right)\frac{L_3}{N_3}\right), \quad i = 1,2,3, \tag{64}$$

on the quadrature grid (11).

The adjoint operator $\boldsymbol{D}^*$ defined in Equation (27) for the discretization at hand (20) takes fields on the quadrature grid (11) and produces fields on the nodal grid (10) via suitable central averages. In particular, the arguments established for the operator $\boldsymbol{D}$ apply to the operator $\boldsymbol{D}^*$ with minimal modifications. Omitting the fine details, we may thus express the action of the operator $\boldsymbol{D}^*\boldsymbol{D}$ on an arbitrary displacement-fluctuation field $\boldsymbol{u} \in \boldsymbol{H}_{\boldsymbol{N}}^3$ in the compact form

$$\boldsymbol{D}^*\boldsymbol{D}\boldsymbol{u}\left(\frac{j_1L_1}{N_1}, \frac{j_2L_2}{N_2}, \frac{j_3L_3}{N_3}\right) = \sum_{\mathbf{k}\in\mathbb{N}_{\boldsymbol{N}}^3} \|\boldsymbol{\xi}(\mathbf{k})\|^2\hat{\boldsymbol{u}}(\mathbf{k}) \ \sin\left(\pi\ \frac{k_1j_1}{N_1}\right)\sin\left(\pi\ \frac{k_2j_2}{N_2}\right)\sin\left(\pi\ \frac{k_3j_3}{N_3}\right), \tag{65}$$

for a fixed index $\mathbf{j} \in \mathbb{N}_{\boldsymbol{N}}^3$. Thus, the operator $\boldsymbol{D}^*\boldsymbol{D}$ acts by multiplying the sine coefficients of the displacement field $\boldsymbol{u}$ with the squared norm of the vector field (63) in sine-frequency space. As the vector $\boldsymbol{\xi}(\mathbf{k})$ does not vanish for $\mathbf{k} \in \mathbb{N}_{\boldsymbol{N}}^3$, we may thus express the action of Green's operator (32) in sine space

$$\boldsymbol{G}^0\boldsymbol{f}\left(\frac{j_1L_1}{N_1}, \frac{j_2L_2}{N_2}, \frac{j_3L_3}{N_3}\right) = \sum_{\mathbf{k}\in\mathbb{N}_{\boldsymbol{N}}^3} \frac{\hat{\boldsymbol{f}}(\mathbf{k})}{\alpha_0\ \|\boldsymbol{\xi}(\mathbf{k})\|^2} \ \sin\left(\pi\ \frac{k_1j_1}{N_1}\right)\sin\left(\pi\ \frac{k_2j_2}{N_2}\right)\sin\left(\pi\ \frac{k_3j_3}{N_3}\right), \tag{66}$$

for any body-force field

$$\boldsymbol{f}\left(\frac{j_1L_1}{N_1}, \frac{j_2L_2}{N_2}, \frac{j_3L_3}{N_3}\right) = \sum_{\mathbf{k}\in\mathbb{N}_{\boldsymbol{N}}^3} \hat{\boldsymbol{f}}(\mathbf{k}) \ \sin\left(\pi\ \frac{k_1j_1}{N_1}\right)\sin\left(\pi\ \frac{k_2j_2}{N_2}\right)\sin\left(\pi\ \frac{k_3j_3}{N_3}\right), \quad \mathbf{j} \in \mathbb{N}_{\boldsymbol{N}}^3, \tag{67}$$

and all indices $\mathbf{j} \in \mathbb{N}_{\boldsymbol{N}}^3$. We thus have all the tools at hand to establish Lippmann–Schwinger solvers—either in displacement form (31) making use of the explicit expression of Green's operator (66) and sine transforms (`DST-I` in all directions) or in deformation-gradient form (35), utilizing the formulas for the gradient (60)–(62) and Green's operator (66) to represent the Eshelby–Green operator (36) with suitable discrete sine and cosine transforms (`DST-II/DST-III` and `DCT-II/DCT-III` for various directions). For the latter strategy, the implementation details comprise the next Section 3.3.

3.3 | Implementation

This section comprises details on the implementation of the two solution schemes for the rotated staggered grid discretization with essential boundary conditions. We commence with the displacement-based basic scheme (33), which—by working on the displacement field—reduces the memory consumption but typically leads to less performing solvers for periodic boundary conditions.[87] The displacement-based strategy is summarized in Algorithm 1 for the case of Dirichlet boundary conditions. A computational overhead is inferred by the action of the operators $\boldsymbol{D}$ and $\boldsymbol{D}^*$ in lines (5) and (7) of Algorithm 1, respectively, which may be computed in real space by in-place stencil computations. Additionally, by working with the FFTW library,[91] we infer an additional overhead caused by the doubled memory use of the `DST-I` implementation. The computational overhead can be avoided by working with the deformation gradient in the first place.

Algorithm 1. The $\boldsymbol{u}$-based basic scheme with prescribed strain $\overline{\boldsymbol{\varepsilon}}$ and Dirichlet boundary conditions

1: $\boldsymbol{u} \leftarrow \boldsymbol{0}$
2: $\boldsymbol{g} \leftarrow \boldsymbol{0}$
3: $\texttt{res} \leftarrow \texttt{tol} + 1$
4: **while** $\texttt{res} > \texttt{tol}$ **do**
5: $\quad \boldsymbol{\sigma} \leftarrow \boldsymbol{S}(\cdot, \bar{\varepsilon} + \boldsymbol{D}\boldsymbol{u})$ ▷ Stencil-based gradient in real space (21)
6: $\quad \bar{\boldsymbol{\sigma}} \leftarrow \texttt{mean}(\boldsymbol{\sigma})$
7: $\quad \boldsymbol{f} \leftarrow \boldsymbol{D}^* \boldsymbol{\sigma}$ ▷ Stencil-based divergence in real space
8: $\quad \hat{\boldsymbol{f}} \leftarrow \texttt{RDFT}(\boldsymbol{f})$ ▷ Forward transform to Fourier space
9: $\quad \hat{\boldsymbol{g}}(\mathbf{k}) \leftarrow \begin{cases} \boldsymbol{0}, & k_1 k_2 k_3 = 0 \\ \frac{\hat{\boldsymbol{f}}(\mathbf{k})}{\|\xi\|^2}, & \text{otherwise} \end{cases}$ ▷ Apply the Green operator (66)
10: $\quad \texttt{res} \leftarrow \hat{\boldsymbol{f}} \cdot \hat{\boldsymbol{g}} / \|\bar{\boldsymbol{\sigma}}\|$
11: $\quad \boldsymbol{g} \leftarrow \texttt{RDFT}^{-1}(\hat{\boldsymbol{g}})$ ▷ Backward transform to real space
12: $\quad \boldsymbol{u} \leftarrow \boldsymbol{u} - \alpha^0 \boldsymbol{g}$
13: **end while**
14: **return** $\boldsymbol{D}\boldsymbol{u}, \bar{\boldsymbol{\sigma}}$

▷ $\texttt{RDFT}(\boldsymbol{g})$
1: $g_i \leftarrow \texttt{DST-I} \otimes \texttt{DST-I} \otimes \texttt{DST-I}(g_i) \quad i = 1, 2, 3$
2: **return** $\boldsymbol{g}$

▷ $\texttt{RDFT}^{-1}(\boldsymbol{g})$
1: $g_i \leftarrow \texttt{DST-I} \otimes \texttt{DST-I} \otimes \texttt{DST-I}(g_i) \quad i = 1, 2, 3$
2: $\boldsymbol{g} \leftarrow \boldsymbol{g}/(8N_1N_2N_3)$
3: **return** $\boldsymbol{g}$

This alternative approach amounts to the (deformation-gradient-based) basic scheme (38) where a stencil-based computation of the derivatives is circumvented by evaluating the action of these very operators in Fourier space at the expense of an increased memory footprint. To compute the derivatives in Fourier space with Dirichlet boundary conditions by the use of sine and cosine series, some extra work is required: Generally, in a deformation-gradient-based implementation, the displacement and the body-force fields—both living on the nodal grid (10)—are never evaluated in real space (49). Rather, they only enter via their sine coefficients $\hat{\boldsymbol{u}}(\mathbf{k})$ and $\hat{\boldsymbol{f}}(\mathbf{k})$, respectively. In fact, the primary quantity of interest is the deformation gradient field (21) in view of the representation (59). To understand it better, it is instructive to consider the one-dimensional case first.

Let us return to the representation formula (55) for the central difference operator (53)

$$\delta^c u(x_j) = \sum_{k=1}^{N-1} \widehat{\delta^c u}_k \cos\left(\pi \frac{\left(j + \frac{1}{2}\right)k}{N}\right) \quad \text{with} \quad \widehat{\delta^c u}_k = \frac{2\sin\left(\frac{k\pi}{2}\right)}{h} \hat{u}_k \quad (k = 1, 2, \dots, N-1). \tag{68}$$

For a sequence $(y_0, y_1, \dots, y_{N-1}) \in \mathbb{R}^N$, the DCT-III is computed via

$$Y_j = y_0 + 2\sum_{k=1}^{N-1} y_k \cos\left(\pi \frac{\left(j+\frac{1}{2}\right)k}{N}\right). \tag{69}$$

In particular, by setting

$$y_0 = 0, \quad y_1 = \frac{1}{2}\widehat{\delta^{\mathrm{c}}u}_1, \quad \dots, \quad y_{N-1} = \frac{1}{2}\widehat{\delta^{\mathrm{c}}u}_{N-1}, \tag{70}$$

we may compute the action of the central difference operator (53)

$$\delta^{\mathrm{c}}u(x_j) \equiv Y_j, \tag{71}$$

via the DCT-III (69). Taking a look at the action of the average operator (58)

$$A^{\mathrm{c}}u(x_j) = \sum_{k=1}^{N-1} \widehat{A^{\mathrm{c}}u}_k \sin\left(\pi \frac{\left(j+\frac{1}{2}\right)k}{N}\right) \quad \text{with} \quad \widehat{A^{\mathrm{c}}u}_k = \cos\left(\frac{k\pi}{2}\right)\hat{u}_k \quad (k = 1, 2, \dots, N-1), \tag{72}$$

we may profit from the DST-III which is defined as

$$Y_j = (-1)^j y_{N-1} + 2\sum_{k=0}^{N-2} y_k \sin\left(\pi \frac{\left(j+\frac{1}{2}\right)(k+1)}{N}\right), \tag{73}$$

for a given input signal $(y_0, y_1, \dots, y_{N-1}) \in \mathbb{R}^N$. Thus, we may set

$$y_0 = \frac{1}{2}\widehat{A^{\mathrm{c}}u}_1, \quad y_1 = \frac{1}{2}\widehat{A^{\mathrm{c}}u}_2, \quad \dots \quad y_{N-2} = \frac{1}{2}\widehat{A^{\mathrm{c}}u}_{N-1}, \quad y_{N-1} = 0, \tag{74}$$

and compute the values (72) by the DST-III (73) with

$$A^{\mathrm{c}}u(x_j) \equiv Y_j. \tag{75}$$

Similar to the difference case (70), one of the coefficients of the input signal needs to be set to zero. However, a *shift* of the coefficients is necessary, as well. To account for this shift, we introduce the shift operator S_-, which acts via

$$S^-(y_0, y_1, \dots, y_{N-1}) = (y_1, \dots, y_{N-1}, y_0). \tag{76}$$

With this operator at hand, we may thus rewrite Equation (74) in the form

$$A^{\mathrm{c}}u = \frac{1}{2}\mathtt{DCT\text{-}III} \circ S_-\left(0, \widehat{A^{\mathrm{c}}u}_1, \widehat{A^{\mathrm{c}}u}_2, \dots, \widehat{A^{\mathrm{c}}u}_{N-1}\right), \tag{77}$$

where the concatenation of operations is denoted by the symbol $\circ$.

With these operations at hand, it is immediately clear how to produce an implementation of the deformation-gradient-based basic scheme, summarized in Algorithm 2. The actions of the discrete divergence, Green's operator (66) and the gradient (60)–(62) are condensed in line 8 of Algorithm 2. We account for the fact that the zero sine frequencies vanish per definition (45) by setting these frequencies to zero. To understand the computation of the gradient (60)–(62), line 1 in the $\mathtt{RDFT}^{-1}(\boldsymbol{F})$ subroutine reveals the action of the differences (70) and (71) and the averages (77), applied to each direction individually. As the DST-II and the DCT-II are inverses of the DST-III and the DCT-III,

Algorithm 2. The $\boldsymbol{F}$-based basic scheme with prescribed strain $\bar{\boldsymbol{\varepsilon}}$ and Dirichlet boundary conditions

$\boldsymbol{F} \leftarrow \bar{\boldsymbol{\varepsilon}}$
$\mathtt{res} \leftarrow \mathtt{tol} + 1$
while $\mathtt{res} > \mathtt{tol}$ **do**
  $\boldsymbol{F}_{\mathrm{old}} \leftarrow \boldsymbol{F}$
  $\boldsymbol{\tau} \leftarrow \boldsymbol{S}(\cdot, \boldsymbol{F}) - \alpha_0 \boldsymbol{F}$ ▷ Compute material law and polarization
  $\bar{\boldsymbol{\sigma}} \leftarrow \mathtt{mean}(\boldsymbol{\tau}) + \alpha_0 \bar{\boldsymbol{\varepsilon}}$ ▷ Calculate mean stress (23)
  $\hat{\boldsymbol{\tau}} \leftarrow \mathtt{RDFT}(\boldsymbol{\tau})$ ▷ Forward transform to Fourier space
  $\hat{\boldsymbol{F}}(\mathbf{k}) = \begin{cases} \mathbf{0}, & k_1 k_2 k_3 = 0 \\ -\frac{\xi \otimes \xi \hat{\tau}(\mathbf{k})}{\alpha^0 \|\xi\|^2}, & \text{otherwise} \end{cases}$ ▷ Apply the Eshelby–Green operator (66)
  $\boldsymbol{F} \leftarrow \mathtt{RDFT}^{-1}(\hat{\boldsymbol{F}})$ ▷ Backward transform to real space
  $\boldsymbol{F} \leftarrow \bar{\boldsymbol{\varepsilon}} + \boldsymbol{F}$ ▷ Adjust average strain
  $\mathtt{res} \leftarrow \|\boldsymbol{F} - \boldsymbol{F}_{\mathrm{old}}\| / \|\bar{\boldsymbol{\sigma}}\|$ ▷ Use standard ℓ^2-norm
end while
return $\boldsymbol{F}, \bar{\boldsymbol{\sigma}}$

▷ $\mathtt{RDFT}(\boldsymbol{F})$

$F_{1i} \leftarrow \mathtt{DCT\text{-}II} \otimes (S_+ \circ \mathtt{DST\text{-}II}) \otimes (S_+ \circ \mathtt{DST\text{-}II})(F_{1i}), \quad i = 1,2,3$
$F_{2i} \leftarrow (S_+ \circ \mathtt{DST\text{-}II}) \otimes \mathtt{DCT\text{-}II} \otimes (S_+ \circ \mathtt{DST\text{-}II})(F_{2i}), \quad i = 1,2,3$
$F_{3i} \leftarrow (S_+ \circ \mathtt{DST\text{-}II}) \otimes (S_+ \circ \mathtt{DST\text{-}II}) \otimes \mathtt{DCT\text{-}II}(F_{3i}), \quad i = 1,2,3$
return $\boldsymbol{F}$

▷ $\mathtt{RDFT}^{-1}(\boldsymbol{F})$

$F_{1i} \leftarrow \mathtt{DCT\text{-}III} \otimes (\mathtt{DST\text{-}III} \circ S_-) \otimes (\mathtt{DST\text{-}III} \circ S_-)(F_{1i}), \quad i = 1,2,3$
$F_{2i} \leftarrow (\mathtt{DST\text{-}III} \circ S_-) \otimes \mathtt{DCT\text{-}III} \otimes (\mathtt{DST\text{-}III} \circ S_-)(F_{2i}), \quad i = 1,2,3$
$F_{3i} \leftarrow (\mathtt{DST\text{-}III} \circ S_-) \otimes (\mathtt{DST\text{-}III} \circ S_-) \otimes \mathtt{DCT\text{-}III}(F_{3i}), \quad i = 1,2,3$
$\boldsymbol{F} \leftarrow \boldsymbol{F} / (8 N_1 N_2 N_3)$
return $\boldsymbol{F}$

respectively, up to a factor of $2N$, and the forward shift operator S^+ acting via

$$S^+(y_0, y_1, \ldots, y_{N-1}) = (y_{N-1}, y_0, y_1, \ldots, y_{N-2}), \tag{78}$$

negates the action of the backward shift (76), the forward operation $\mathtt{RDFT}(\boldsymbol{F})$ becomes clear as well.

We close with the following remarks.

1. More powerful solvers may be developed improving the basic scheme (38). We refer to the review article[72] for further details.
2. As discussed in Risthaus–Schneider,[86] working with a finite-strain preconditioner for a small-strain problem increases the required iterations roughly by a factor of two compared to the case of a small-strain preconditioner. Unfortunately, it is not clear whether a small-strain preconditioner may be represented efficiently by using sine/cosine series. The underlying problem appears to be the mixing of sine and cosine terms in the shear components.

4 | COMPUTATIONAL INVESTIGATIONS

4.1 | Setup

We examine the numerical performance of applying essential boundary conditions in combination with a finite difference discretization. We implemented the rotated staggered grid discretization with Dirichlet boundary conditions as an extension to our in-house FFT solvers[72,87] for the displacement-based solver ($\boldsymbol{u}$-Q1R) and for the deformation-gradient-based solver (Q1R). Building on the prior changes to the deformation-gradient-based solver,[86] further updates to employ the

rotated staggered grid discretization involved an adaptation of the preconditioner and making the `DCT-II/DST-II` and `DCT-III/DST-III` routines of the FFTW library[91] available to the solver routines. Similarly, the displacement-based solver utilizes the `DST-I` routines of the same library. As the boundary conditions are handled via the preconditioners, most of the available solvers may be used directly in our homogenization framework. The conjugate gradient (CG) solver turned out to perform best with Dirichlet boundary conditions.[86] For this reason, for all following evaluations, we rely on the CG solver. Additionally, we validate our results against underintegrated-Q1 finite-element results obtained by the *FEniCSx*[92] library and the PETsc solver. Runtimes were recorded on a 2×48-core AMD EPYC CPU with 1024GB of RAM.

4.2 | Computational performance

As shown in the authors' previous publication,[86] it is possible to extend FFT-based homogenization schemes to include Dirichlet boundary conditions without a critical loss in computational performance. In the following, we will investigate the performance of the Q1R discretization with Dirichlet (or essential) boundary conditions (Q1R-D, Equation 66) and compare it to its periodic counterpart (Q1R-P, Willot[53]), as well as to the Moulinec–Suquet discretization with periodic boundary conditions (MS-P,[1]) and with Dirichlet boundary conditions (MS-D, Moulinec-Suquet[86]). We assess the performance of these discretizations in a deformation-gradient-based setting, summarized in Algorithm 2. For comparison, we additionally evaluate the discretization in a displacement-based setting as laid out in Algorithm 1. We denote this combination by the shorthand $\boldsymbol{u}$-Q1R-D for Dirichlet boundary conditions and by $\boldsymbol{u}$-Q1R-P for periodic boundary conditions.

We consider a simple heterogeneous but periodic microstructure, that is, a single spherical inclusion centered in the volume element of interest with a radius of $5/16L$, fixed relative to the volume element size. In our experiments, we use the isotropic elastic material constants of E-glass for the spherical inclusion, that is, a Young's modulus of 72 GPa and a Poisson's ratio of 0.2. Accordingly, we set the Young's modulus of the polymer matrix material to 3.0 GPa with a Poisson's ratio of 0.35. We apply a uniaxial mean strain of 5%. The residual versus the number of iterations in the (linear) CG solver for the single spherical inclusion on a 256^3 grid is shown in Figure 2A for the considered schemes. We observe that for this finite material contrast, both schemes with periodic boundary conditions (Q1R-P and MS-P) reach a residual below the tolerance of 10^{-5} in fewer iterations than for Dirichlet boundary conditions. This is due to the

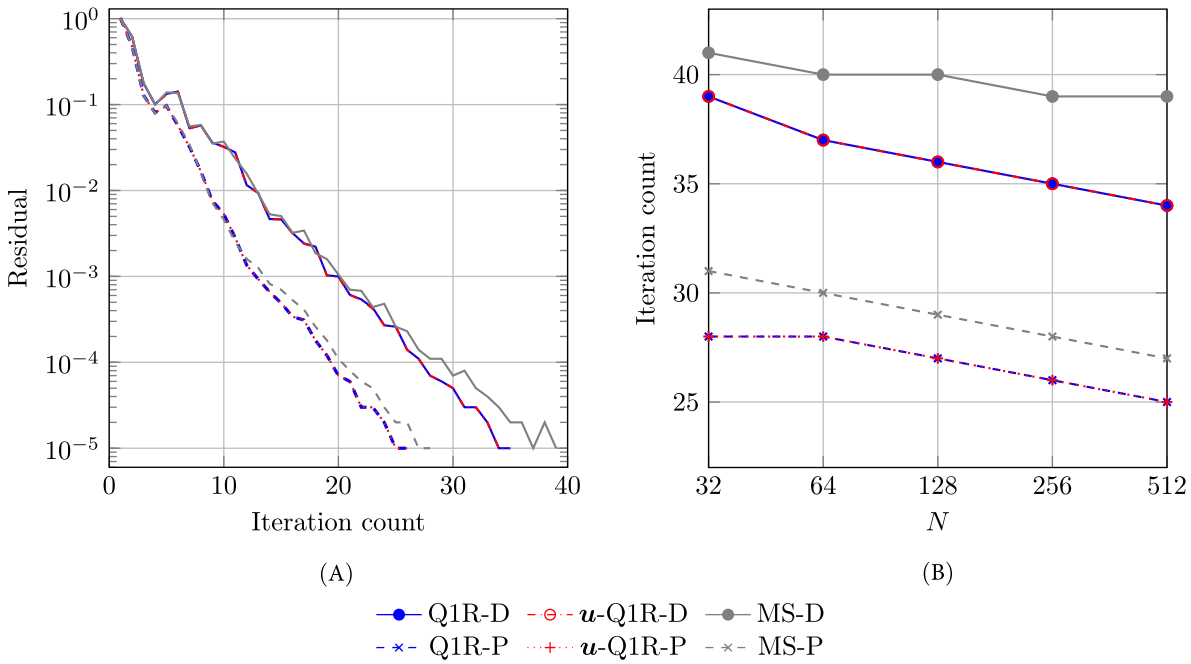


FIGURE 2 Residuals and iteration count for the single spherical inclusion using the conjugate gradient solver and a tolerance of 10^{-5}. (A) Residual versus iteration count on a 256^3 grid. (B) Iteration counts for various resolutions.

large-strain preconditioner we introduced to apply Dirichlet boundary conditions. To be more precise, the number of iterations for linear CG scales as the square root of the condition number of the linear system. We observe that for periodic boundary conditions, both discretization types require about ten fewer iterations than their counterparts with Dirichlet boundary conditions. Unfortunately, since the large-strain preconditioner doubles the condition number of the system, an increase in the number of iterations is to be expected regardless of the discretization used. Nevertheless, the Q1R-D discretization allows a noticeable reduction in the number of iterations compared to the MS-D discretization (by about 10%). This advantage becomes even more apparent in Figure 2B, where the iteration counts for different resolutions of the single spherical inclusion are plotted. In particular, the Q1R-D discretization reduces the number of iterations by up to 10% compared to the MS-D discretization for the same resolution. This effect mirrors the reduction in the number of iterations that we observe when comparing the Q1R-P and MS-P discretizations. At higher resolutions, all discretization types show a reduction in the number of iterations, which is caused by the decrease in the number of interface voxels.[93]

In addition, the MS-D discretization shows the same number of iterations for all resolutions except for the lowest-resolution ($N = 32$). For both the periodic ($\boldsymbol{u}$-Q1R-P) and the Dirichlet ($\boldsymbol{u}$-Q1R-D) case, the displacement-based implementations show a convergence behavior that is indistinguishable from their respective deformation-gradient-based (Q1R) counterparts as they compute the same iterates but in a different manner. Another important aspect of the computational performance is the actual runtime of the solver. As the large-strain preconditioner for Dirichlet boundary conditions involves nine instead of six entries for the gradient field, we expect an increase in solver runtime for Dirichlet boundary conditions. To evaluate this effect, in Figure 3A the total solver runtime is plotted for various resolutions of the single spherical inclusion. In general, we notice that for all discretizations the runtimes are of the order of the voxel count $\mathcal{O}(N^3)$. The increase in runtimes for Dirichlet boundary conditions is rooted in the increase of iterations and the increase in operations per voxel. The Dirichlet boundary conditions scale to larger volume elements the same way as periodic boundary conditions do. The runtime for the Q1R discretization is rather close to the MS discretization for both periodic and Dirichlet boundary conditions. The displacement-based $\boldsymbol{u}$-Q1R schemes show a different behavior. Even though the periodic $\boldsymbol{u}$-Q1R-P scheme is considerably slower than the Q1R-P scheme for all sizes, the displacement-based scheme with Dirichlet boundary conditions ($\boldsymbol{u}$-Q1R-D) does not show a similar increase in runtime over the Q1R-D scheme. In fact, the $\boldsymbol{u}$-Q1R-D scheme is slightly faster than the Q1R-D scheme for all investigated domain sizes larger than a low size $N = 64$.

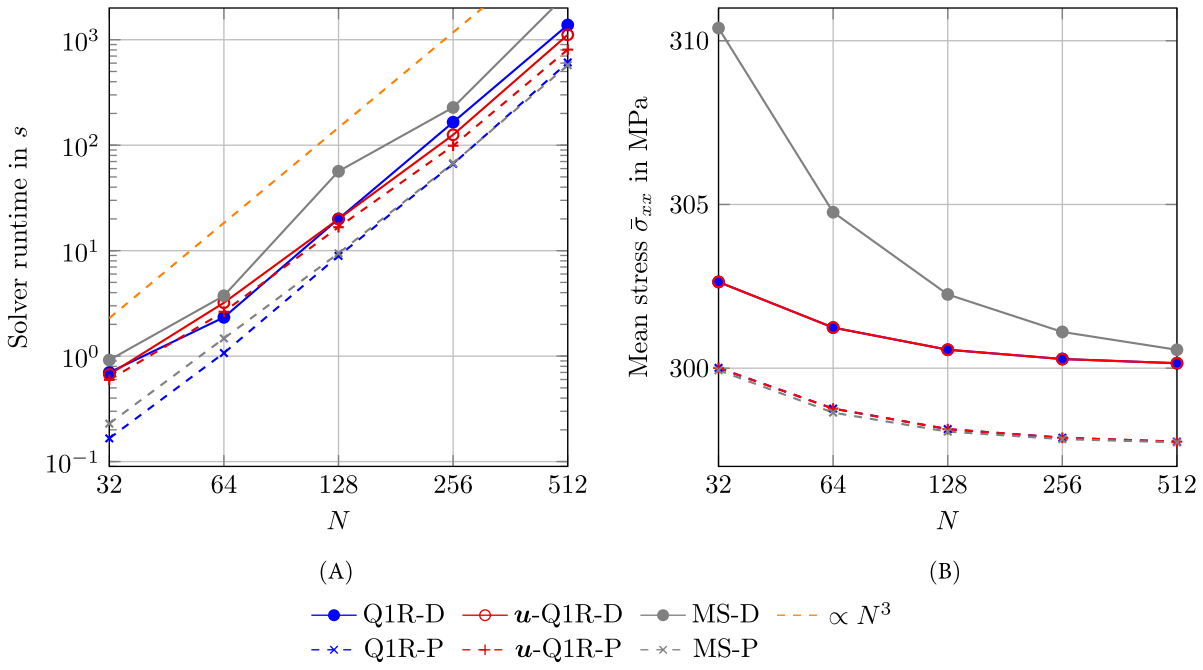


FIGURE 3 Solver runtime and apparent mean stress for the single spherical inclusion and various resolutions N^3 with the conjugate gradient solver. (A) Solver runtimes. (B) Mean stress $\bar{\sigma}_{xx}$ for various resolutions N^3.

As we saw in Figure 2B, this difference is not rooted in the convergence properties of the scheme (as both schemes have exactly the same number of iterations), but in the runtime of a single iteration. Here, the $\boldsymbol{u}$-Q1R-D scheme avoids some of the added overhead of the slower DCT and DST transforms by computing only three instead of nine transforms per iteration. Therefore, the runtime of the $\boldsymbol{u}$-Q1R-D scheme is rather close to the runtime of the $\boldsymbol{u}$-Q1R-P scheme. Nevertheless, like for the $\boldsymbol{u}$-Q1R-P scheme, we need to compute the stencil-based gradient for the $\boldsymbol{u}$-Q1R-D scheme. For this reason, both schemes are slower than the Q1R-P scheme. All discretizations retain a linear scaling of the runtime in relation to the number of voxels.

To conclude the discussion on this first example, we will take a look at the apparent mean stress $\overline{\sigma}_{xx}$ for a mean strain of 5% in x-direction as shown in Figure 3B. As asserted by Hill,[94] we generally observe a higher apparent stress for Dirichlet boundary conditions than for periodic boundary conditions. As in the periodic case, the apparent mean stresses converge for higher resolutions for the Q1R-D discretization. Here, the MS-D discretization with Dirichlet boundary conditions presents an outlier. The MS-D discretization implicitly sets the outermost voxel of the domain to zero, virtually increasing the volume fraction of the inclusion in the domain. Therefore, especially for small resolutions, this discretization scheme returns higher apparent stresses than the Q1R-D discretization. The effective stresses computed for both the $\boldsymbol{u}$-Q1R-D and $\boldsymbol{u}$-Q1R-P implementations coincide with their deformation-gradient-based counterparts Q1R-D and Q1R-P up to numerical precision, respectively. We computed validation results for hexahedral elements with reduced integration and Dirichlet boundary conditions (Q1R-D) using the FEniCSx FE-library. The apparent mean stresses are listed in Table 1. For the low resolutions $N = 32$ and $N = 64$, the validation computations provide extremely close results to our Q1R-D and $\boldsymbol{u}$-Q1R-D solution schemes. Alas, the direct FE validation solver does not converge for larger resolutions including $N = 128$ and higher. This divergent behavior is caused by spurious hourglassing modes.[95,96] Consequently, we also computed validation results with hourglass control (HGC) enabled, which led to convergence but provided slightly higher apparent stresses for the FE results. These studies demonstrate that there is an excellent agreement of both our solution schemes with Dirichlet boundary conditions with the results of established solvers. Furthermore, our solvers allow for noticably finer resolutions compared to the FE approach, as shown in Figure 3B.

4.3 | Dirichlet boundary conditions on periodic microstructures

After discussing the computational performance of the Q1R-D discretization, we take another look at the influence of the selected boundary conditions on the apparent properties and the local stress fields. For computational homogenization, the choice of boundary conditions is strongly linked to the periodicity of the microstructure. For instance, employing statistically similar but periodized microstructures provides faster convergence of the computed apparent properties toward the actual effective properties.[6-8] To this end, we use a slightly smaller single spherical inclusion with a radius of $L/4$ inside a cube $[0, L]^3$ and shift it in direction of the x-axis by a varying extent x_{sh} and apply a uniaxial strain in direction of the x-axis. The resulting periodic microstructures are shown in Figure 4. In Figure 5, the local stress fields σ_{xx} are shown for an inclusion centered on a domain surface, that is, the one shown in Figure 4A. In this case we observe a significant increase of the local stresses inside the spherical inclusion for all solution fields with Dirichlet boundary conditions. Comparing the Q1R-D discretization (Figure 5A) with the MS-D discretization (Figure 5C), we notice a strong decrease of the ringing artifacts introduced by the trigonometric approximation implied by the discrete Fourier transforms.[53] This phenomenon mirrors the reduction in ringing artifacts observed for the conventional, periodic Q1R-P discretization (Figure 5B). The local stress fields for the spherical inclusion shifted by $x_{\text{sh}} = 3L/16$—so that it only partially intersects

TABLE 1 Apparent mean stress $\overline{\sigma}_{xx}$ in MPa for the single spherical inclusion and various resolutions N^3 for different methods of computation.

	Q1R-D	$\boldsymbol{u}$-Q1R-D	Q1R-D (FE)	
N	(ours)	(ours)	no HGC	1% HGC
32	302.63321	302.63321	302.63321	302.89896
64	301.24000	301.24000	301.24003	301.31681
128	300.56354	300.56354	—	300.60452

Abbreviations: HGC, hourglass control; Q1R-D, Q1R discretization with Dirichlet (or essential) boundary.

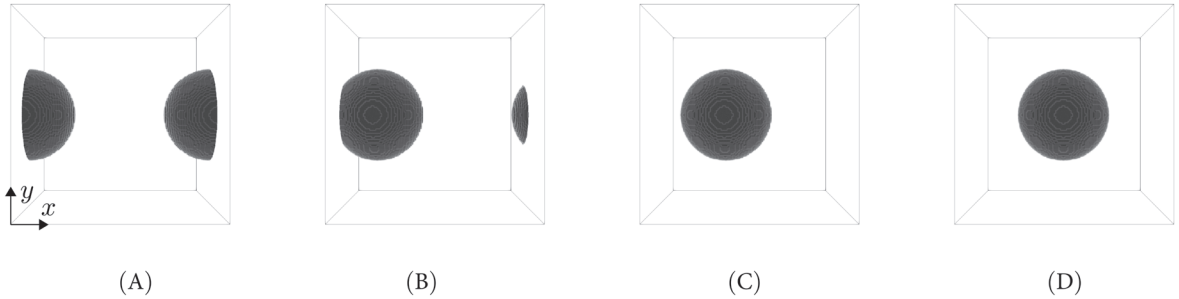


FIGURE 4 Spherical inclusion in a cubic cell $[0, L]^3$ with shift x_{sh} in x-direction. (A) $x_{\text{sh}} = 0$. (B) $x_{\text{sh}} = 3L/16$. (C) $x_{\text{sh}} = 3L/8$. (D) $x_{\text{sh}} = L/2$.

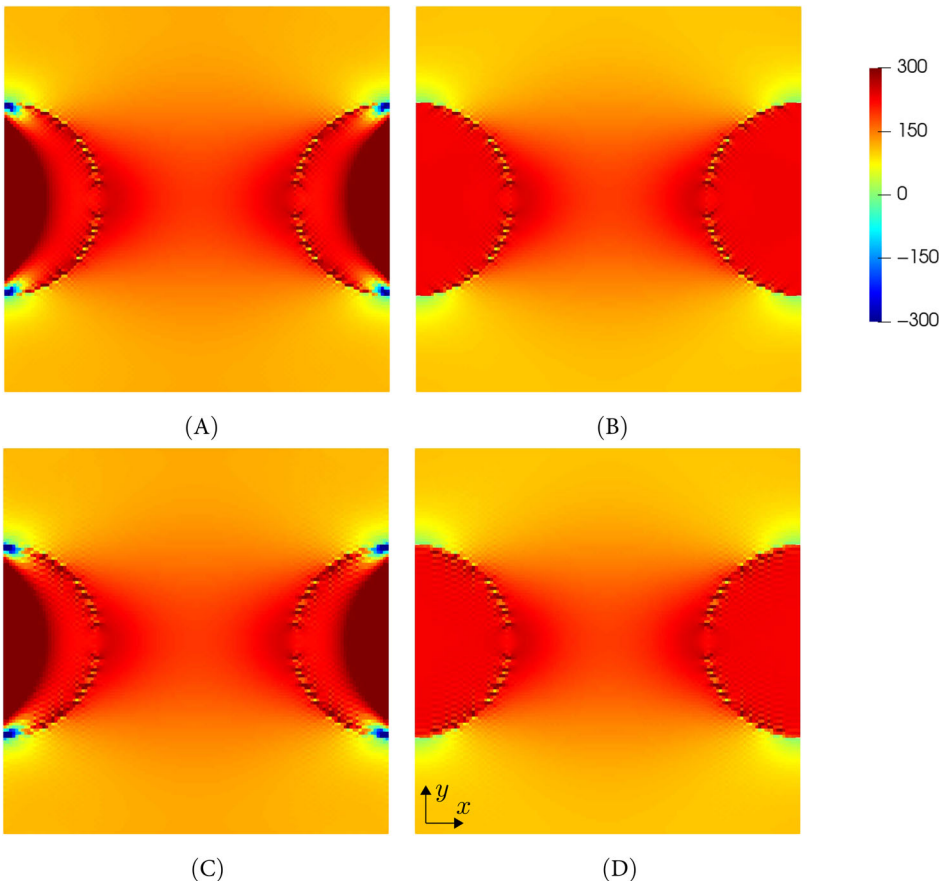


FIGURE 5 Local stress field σ_{xx} in MPa on a slice of a 128^3 solution field of the shifted spherical inclusion for a shift $x_{\text{sh}} = 0$. (A) Q1R discretization with Dirichlet (or essential) boundary (Q1R-D). (B) Q1R-periodic boundary (Q1R-P). (C) Moulinec–Suquet Dirichlet boundary (MS-D). (D) Moulinec–Suquet periodic boundary (MS-P).

the domain boundaries—are shown in Figure 6. The stress fields for periodic boundary conditions (Figure 6B,D) are not affected by the shift of the inclusion. On the other hand, applying Dirichlet boundary conditions leads to significantly different solution fields. For example, Figure 6A shows an increased local stress inside the spherical inclusion as well as areas of low stress above and below the inclusion. This higher stress gradient at the inclusion-matrix interface leads to ringing artifacts. For this example, the benefits of using the Q1R-D discretization become clear, as significantly fewer ringing artifacts are visible in Figure 6A.

Figure 7 shows the apparent stresses $\overline{\sigma}_{xx}$ for different shift magnitudes of the spherical inclusion with respect to the volume element. The periodic boundary conditions show no variation in the apparent stress. In contrast, using Dirichlet

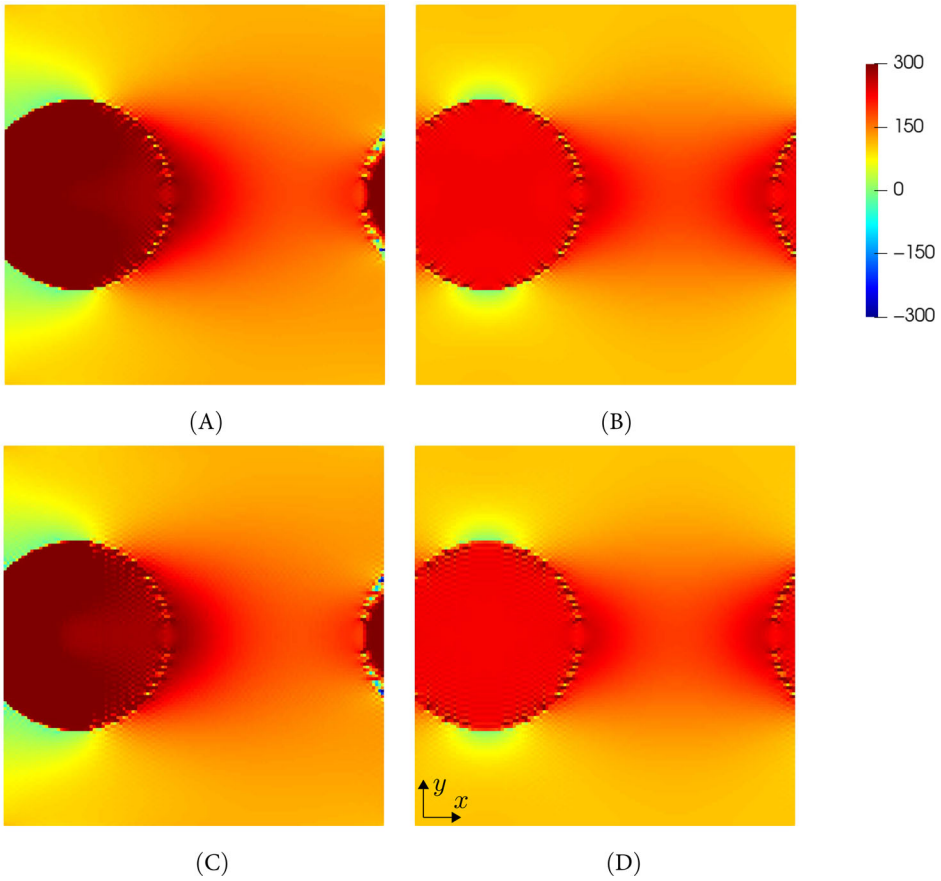


FIGURE 6 Local stress field σ_{xx} in MPa on a slice of a 128^3 solution field of the shifted spherical inclusion for a shift $x_{\text{sh}} = 3L/16$ (A) (A) Q1R discretization with Dirichlet (or essential) boundary (Q1R-D). (B) Q1R-periodic boundary (Q1R-P). (C) Moulinec–Suquet Dirichlet boundary (MS-D). (D) Moulinec–Suquet periodic boundary (MS-P).

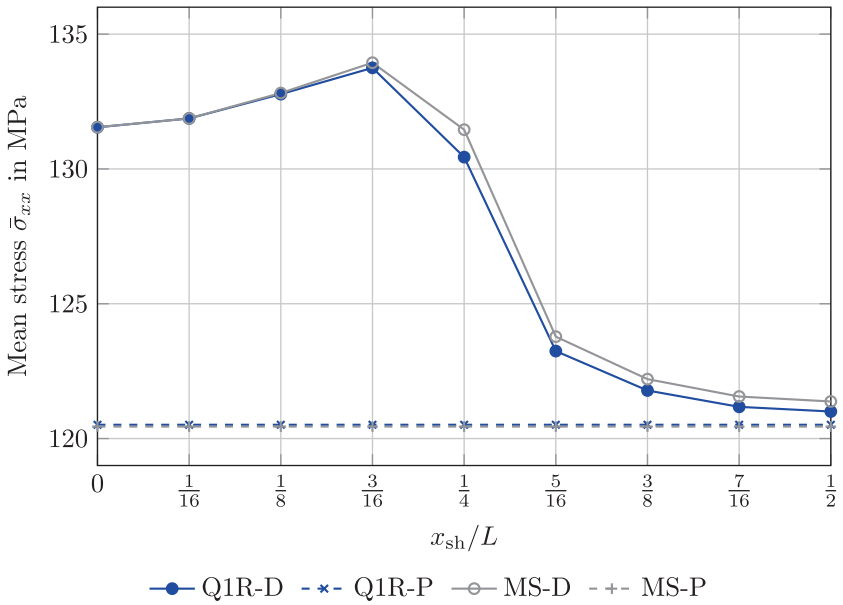


FIGURE 7 Mean stress $\bar{\sigma}_{xx}$ for various shift distances x_{sh} of a single spherical inclusion with a resolution of 128^3.

boundary conditions leads to influence of the shifting distance. All three considered discretization schemes with Dirichlet boundary conditions show a similar behavior: In cases where the sphere intersects the domain boundary, the apparent stress is significantly increased compared to the periodic boundary conditions. Shifting the spherical inclusion by $x_{\mathrm{sh}} = 3L/16$ leads to the highest apparent stresses. In this case, the inclusion is only partially constrained by the boundary condition. Increasing the shift beyond $x_{\mathrm{sh}} = 3L/16$ leads to a strong decrease of the apparent stresses. For larger shifts, the difference between the MS-D and Q1R-D discretizations becomes apparent. The MS-D discretization yields a higher apparent stress. When the inclusion is shifted to the center of the volume element ($x_{\mathrm{sh}}/L = 1/2$), the difference in apparent stress becomes less noticeable. In this case, the apparent stress of the Q1R-D discretization is closer to the stress of the Q1R-P discretization than to the stress of the MS-D discretization.

4.4 | Application to open cell foams

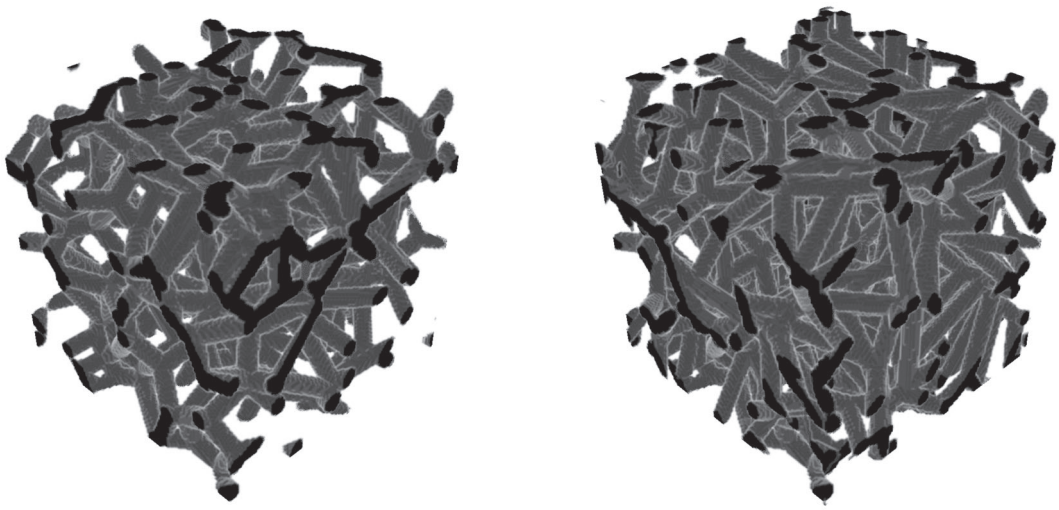
As an interesting application of Dirichlet boundary conditions to heterogeneous microstructures of industrial relevance we consider open cell foams, which are used in various applications due to their advantageous mechanical properties despite their light weight.[97,98]

Such foams represent a class of materials with a random microstructure, whose analysis by digital images leads to nonperiodic microstructures, in general. In particular, computational homogenization strategies using periodic boundary conditions, as typically assumed by FFT-based computational homogenization techniques, run into problems. In fact, for a high porosity, it might happen that the microstructure is no longer mechanically stable, and computational approaches presuming periodicity produce nonsense.

A possible work-around is based on generating (statistically equivalent) periodic microstructure models. However, such a strategy has to account for the additional error inferred by the microstructure-generation process and may not exploit the wealth of information present in real-world microstructure data.

Alternatively, one may directly apply Dirichlet boundary conditions to handle nonperiodic geometries. As an additional challenge, the presence of pores leads to an infinite material contrast, which is known to cause problems for solution methods using the Moulinec–Suquet discretization.[50,62]

To assess the influence of periodicity, we consider two generated volume elements of the open cell foams, a periodic and a nonperiodic one, consisting of a network of trusses with a porosity of 80%, that is, an aluminum volume fraction of 20%, see Figure 8. Both the periodic (Figure 8A) and the nonperiodic (Figure 8B) microstructure are generated from random Laguerre tessellations[99–101] where the trusses representing the foam are obtained by thickening the edges of the Laguerre cells. We computed the effective properties for a uniaxial strain loading in direction of the x-axis for an aluminum



(A) (B)

FIGURE 8 Open cell foam microstructures with a porosity of 80%. (A) Periodic open cell foam. (B) Nonperiodic open cell foam.

open cell foam with a Young's modulus of 72 GPa and a Poisson's ratio of 0.22. The voids in the open cell foam have zero stiffness, leading to an infinite material contrast. For both microstructures shown in Figure 8, we computed the local stress fields for both the Q1R-P and Q1R-D discretizations.

Figure 9 comprises three-dimensional views of the local stress fields σ_{xx} in the foam material for different boundary conditions. Figure 9A shows the local stress field for Dirichlet boundary conditions on the periodic microstructure. The Dirichlet boundary conditions lead to increased stresses at the domain boundary compared to periodic boundary conditions on the same microstructure in Figure 9C. Nevertheless, on interior trusses of the foam, the local stresses are on a similar level. In contrast, for the nonperiodic microstructure, we observe a strong difference between Dirichlet (Figure 9B) and periodic (Figure 9D) boundary conditions. Table 2 shows the apparent mean stresses $\overline{\sigma}_{xx}$ for the load case at hand.

Notably, the mean stress for periodic boundary conditions on the nonperiodic microstructure which is the lowest. This outlier is caused by the missing connection between trusses across the domain boundary for periodic boundary conditions. With Dirichlet boundary conditions and for the same microstructure, we do not observe this strong reduction in mean stress. Of course, there are still differences between the different models used. However, these differences are expected to diminish when considering larger cells.

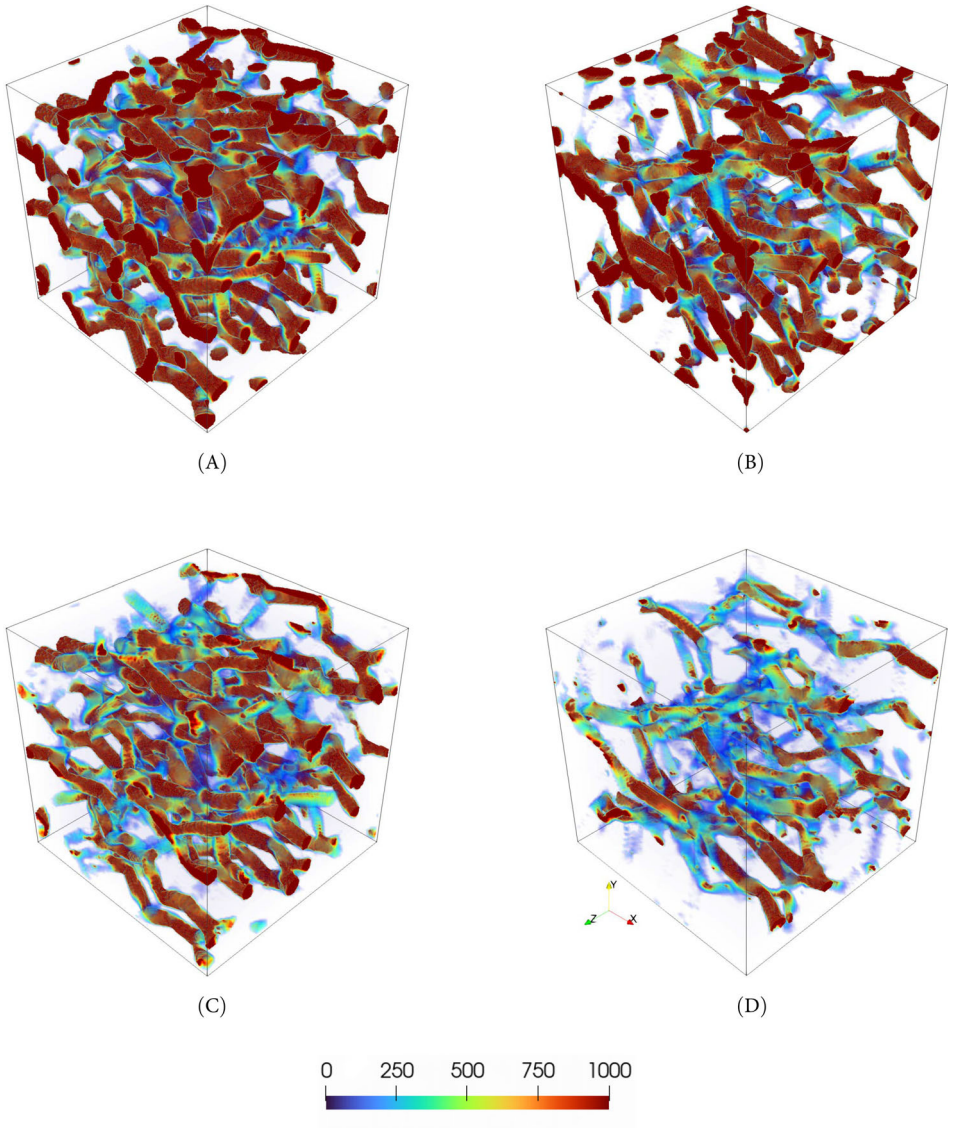


FIGURE 9 Local stress fields σ_{xx} in MPa on the on aluminum open cell foam with a porosity of 80% for 5% uniaxial extension. Only voxels with positive stress values are shown. (A) Dirichlet boundary conditions on the periodic microstructure. (B) Dirichlet boundary conditions on the nonperiodic microstructure. (C) Periodic boundary conditions on the periodic microstructure. (D) Periodic boundary conditions on the nonperiodic microstructure.

TABLE 2 Mean stress $\overline{\sigma}_{xx}$ in MPa for the open cell foam for a mean strain of 5%.

	Q1R-P	Q1R-D
Periodic microstructure (Figure 8A)	117.6	171.4
Nonperiodic microstructure (Figure 8B)	43.6	133.6

Abbreviations: Q1R-D, Q1R discretization with Dirichlet (or essential) boundary; Q1R-P, Q1R discretization with periodic (or essential) boundary.

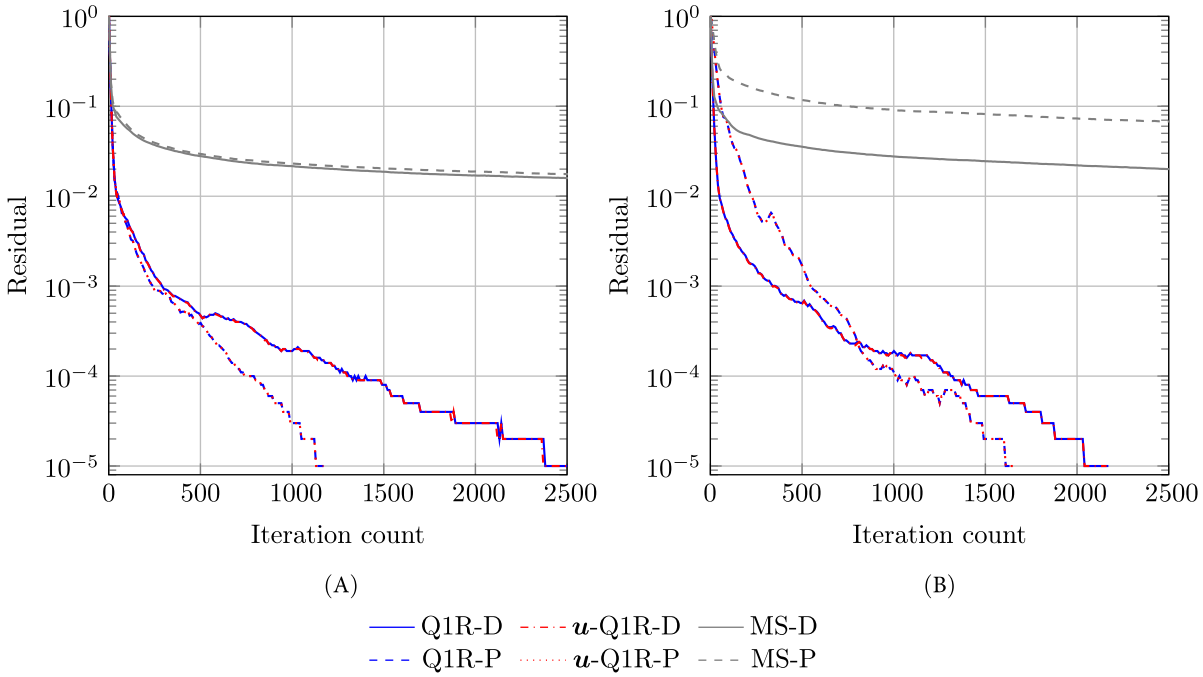


FIGURE 10 Residuals versus iteration count for the foam microstructures and the conjugate gradient solver. (A) Periodic microstructure. (B) Nonperiodic microstructure.

Finally, we take a look at the convergence behavior of the numerical solvers for the different microstructures. Generally speaking, FFT-based homogenization methods using ansatz functions with global support, like the Moulinec-Suquet discretization, run into convergence problems for materials with infinite material contrast.[72] For the problem at hand, Figure 10 shows the residual versus iteration count for various discretization schemes applied to both the periodic open foam microstructure (Figure 10A) and the nonperiodic microstructure (Figure 10B). For both the microstructure types, the Q1R-P reaches a residual below the tolerance in a similar amount of iterations. Using the Q1R-D discretization scheme and therefore Dirichlet boundary conditions, convergence is reached but necessitates more iterations, which is again rooted in the inferior conditioning associated to the preconditioner for Dirichlet boundary conditions. For the nonperiodic boundary conditions, the Q1R-D discretizations shows faster convergence at firstbut is overtaken by the Q1R-P discretization at around 1000 iterations.

The Moulinec–Suquet-based discretizations (MS-P and MS-D) reach convergence in neither case. Interestingly, for the periodic microstructure both discretization types show a similar decrease of the residual, although a residual below the tolerance is not reached in under 2500 iterations. In contrast, with the nonperiodic microstructure, the MS-P discretization stagnates at an even higher residual than the MS-D discretization with Dirichlet boundary conditions. This phenomenon is caused by the missing connection between trusses across the boundary with periodic boundary conditions.

5 | CONCLUSIONS

This work was devoted to developing Lippmann–Schwinger solvers for micromechanical problems at small strains with essential boundary conditions and the discretization on a rotated staggered grid. In a nutshell, we used formulations on

the displacement as well as on the deformation gradient and employed dedicated discrete sine and cosine transforms to tap into the power of FFT-based computational micromechanics available for periodic boundary conditions. In retrospect, the deformation-gradient-based implementation of the developed scheme is actually simpler than for the Moulinec–Suquet discretization considered in Risthaus-Schneider[86] because no weighted quadrature needs to be used. However, for the deformation-gradient-based version, the use of the DCTs and DSTs is a bit more involved for the rotated staggered grid than for the Moulinec–Suquet discretization, essentially for two reasons. For a start, different DCTs/DSTs need to be used for the inverse transform than for the forward transform. Moreover, shift operations (76) and (78) are necessary to apply Green's operator properly. Nevertheless, once these hurdles are overcome, the established scheme may be integrated into existing FFT-based computational micromechanics codes with little effort, taking advantage of the implemented solvers and macroscopic loadings[102,103] with ease. Additionally, we showed that the displacement-based $\boldsymbol{u}$-Q1R-D implementation provides benefits regarding the evaluation time, while providing the exact same convergence properties and apparent properties. Despite this, the deformation-gradient-based Q1R-D scheme provides similar performance while maintaining compatibility with our large solver framework.

Let us briefly comment on possible extensions and directions of further research. Other authors have been trying to impose essential boundary conditions for their own reasons, for example, to mimic an experimental setup,[82,83] for physical reasons[22,80] or as a prerequisite for a multilevel scheme.[84] These applications may be pursued with the developed technology.

On a related note, it might be of interest to combine essential boundary conditions for the displacement-fluctuation field in one direction with periodic or stress boundary conditions in other directions, extending the work of Gélébart[78] to solid mechanics. Working out a flexible framework for doing so might be of immediate interest. Also, opening up these more general boundary conditions for the displacement fluctuation to other popular discretizations in computational micromechanics, like the staggered grid,[50] could prove useful.

ACKNOWLEDGMENTS

MS acknowledges support from the European Research Council within the Horizon Europe program—project 101040238. Funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—418247895—is gratefully acknowledged by LR. We thank the reviewers for the constructive suggestions, which helped to significantly improve the manuscript.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Matti Schneider https://orcid.org/0000-0001-7017-3618

REFERENCES

1. Moulinec H, Suquet P. A fast numerical method for computing the linear and nonlinear mechanical properties of composites. *Comptes Rend Acad Sci Sér II*. 1994;318(11):1417-1423.
2. Moulinec H, Suquet P. A numerical method for computing the overall response of nonlinear composites with complex microstructure. *Comput Methods Appl Mech Eng*. 1998;157:69-94.
3. Zeller R, Dederichs PH. Elastic constants of polycrystals. *Phys Status Solidi*. 1973;55(2):831-842.
4. Kröner E. Bounds for effective elastic moduli of disordered materials. *J Mech Phys Solid*. 1977;25(2):137-155.
5. Mura T. *Micromechanics of Defects in Solids*. Martinus Nijhoff; 1987.
6. Terada K, Hori M, Kyoya T, Kikuchi N. Simulation of the multi-scale convergence in computational homogenization approaches. *Int J Solids Struct*. 2000;37(16):2285-2311.
7. van der Sluis O, Schreurs PJG, Brekelmans WAM, Meijer HEH. Overall behavior of heterogeneous elastoviscoplastic materials: effect of microstructural modelling. *Mech Mater*. 2000;32:449-462.
8. Schneider M, Josien M, Otto F. Representative volume elements for matrix-inclusion composites - a computational study on the effects of an improper treatment of particles intersecting the boundary and the benefits of periodizing the ensemble. *J Mech Phys Solid*. 2022;158:104652.
9. Lahellec N, Michel JC, Moulinec H, Suquet P. Analysis of inhomogeneous materials at large strains using fast fourier transforms. In: Miehe C, ed. *IUTAM Symposium on Computational Mechanics of Solid Materials at Large Strains. Solid Mechanics and Its Applications*. Vol 108. Springer; 2003:247-258.
10. Göküzüm FS, Keip M-A. An algorithmically consistent macroscopic tangent operator for FFT-based computational homogenization. *Int J Numer Methods Eng*. 2018;113:581-600.

11. Ma R, Truster TJ. FFT-based homogenization of hypoelastic plasticity at finite strains. *Comput Methods Appl Mech Eng*. 2019;349:499-521.
12. Herrmann KP, Müller WH, Neumann S. Linear and elastic-plastic fracture mechanics revisited by use of Fourier transforms: theory and application. *Comput Mater Sci*. 1999;6:186-196.
13. Li J, Meng S, Tian X, Song F, Jiang C. An non-local fracture model for composite laminates and numerical simulations by using the FFT method. *Compos Pt B*. 2012;43:961-971.
14. Li J, Tian X-X, Abdelmoula R. A damage model for crack prediction in brittle and quasi-brittle materials solved by the FFT method. *Int J Fract*. 2012;173:135-146.
15. Brenner R. Numerical computation of the response of piezoelectric composites using Fourier transform. *Phys Rev B*. 2009;79(18):1-15.
16. Brenner R, Bravo-Castillero J. Response of multiferroic composites inferred from a fast-Fourier-transform-based numerical scheme. *Smart Mater Struct*. 2010;19(11):115004.
17. Sixto-Camacho LM, Bravo-Castilleo J, Brenner R, et al. Asymptotic homogenization of periodic thermo-magneto-electro-elastic heterogeneous media. *Comput Math Appl*. 2013;66(10):2056-2074.
18. Monchiet V, Bonnet G, Lauriat G. A FFT-based method to compute the permeability induced by a stokes slip flow through a porous medium. *Comp Rend Méc*. 2009;337(4):192-197.
19. Bignonnet F, Dormieux L. FFT-based bounds on the permeability of complex microstructures. *Int J Numer Anal Method Geomech*. 2014;38:1707-1723.
20. Abdallah B, Willot F, Jeulin D. Stokes flow through a Boolean model of spheres: representative volume element. *Transport Porous Media*. 2015;109:711-726.
21. Segurado J, Lebensohn RA. An FFT-based approach for Bloch wave analysis: application to polycrystals. *Comput Mech*. 2021;68:981-1001.
22. Sancho R, de Pedraza VR, Lafourcade P, Lebensohn RA, Segurado J. An implicit FFT-based method for wave propagation in elastic heterogeneous media. *Comput Methods Appl Mech Eng*. 2023;404:115772.
23. Eyre DJ, Milton GW. A fast numerical scheme for computing the response of composites using grid refinement. *Eur Phys J Appl Phys*. 1999;6(1):41-47.
24. Michel JC, Moulinec H, Suquet P. A computational method based on augmented Lagrangians and fast Fourier transforms for composites with high contrast. *Comput Model Eng Sci*. 2000;1(2):79-88.
25. Michel JC, Moulinec H, Suquet P. A computational scheme for linear and non-linear composites with arbitrary phase contrast. *Int J Numer Methods Eng*. 2001;52:139-160.
26. Monchiet V, Bonnet G. A polarization-based FFT iterative scheme for computing the effective properties of elastic composites with arbitrary contrast. *Int J Numer Methods Eng*. 2012;89:1419-1436.
27. Monchiet V, Bonnet G. Numerical homogenization of nonlinear composites with a polarization-based FFT iterative scheme. *Comput Mater Sci*. 2013;79:276-283.
28. Moulinec H, Silva F. Comparison of three accelerated FFT-based schemes for computing the mechanical response of composite materials. *Int J Numer Methods Eng*. 2014;97:960-985.
29. Schneider M, Wicht D, Böhlke T. On polarization-based schemes for the FFT-based computational homogenization of inelastic materials. *Comput Mech*. 2019;64(4):1073-1095.
30. Schneider M. On non-stationary polarization methods in FFT-based computational micromechanics. *Int J Numer Methods Eng*. 2021;122(22):6800-6821.
31. Zhou H, Bhattacharya K. Accelerated computational micromechanics and its application to polydomain liquid crystal elastomers. *J Mech Phys Solids*. 2022;153:104470.
32. Willot F. The effective conductivity of strongly nonlinear media: the dilute limit. *Int J Solid Struct*. 2020;184:287-295.
33. Ernesti F, Schneider M. A fast Fourier transform based method for computing the effective crack energy of a heterogeneous material on a combinatorially consistent grid. *Int J Numer Methods Eng*. 2021;122(21):6283-6307.
34. Kabel M, Böhlke T, Schneider M. Efficient fixed point and Newton-Krylov solvers for FFT-based homogenization of elasticity at large deformations. *Comput Mech*. 2014;54(6):1497-1514.
35. Schneider M. An FFT-based fast gradient method for elastic and inelastic unit cell homogenization problems. *Comput Methods Appl Mech Eng*. 2017;315:846-866.
36. Ernesti F, Schneider M, Böhlke T. Fast implicit solvers for phase field fracture problems on heterogeneous microstructures. *Comput Methods Appl Mech Eng*. 2020;363:112793.
37. Schneider M. On the Barzilai-Borwein basic scheme in FFT-based computational homogenization. *Int J Numer Methods Eng*. 2019;118(8):482-494.
38. Brisard S, Dormieux L. FFT-based methods for the mechanics of composites: a general variational framework. *Comput Mater Sci*. 2010;49(3):663-671.
39. Zeman J, Vondřejc J, Novák J, Marek I. Accelerating a FFT-based solver for numerical homogenization of periodic media by conjugate gradients. *J Comput Phys*. 2010;229(21):8065-8071.
40. Schneider M. A dynamical view of nonlinear conjugate gradient methods with applications to FFT-based computational micromechanics. *Comput Mech*. 2020;66:239-257.
41. Gélébart L, Mondon-Cancel R. Non-linear extension of FFT-based methods accelerated by conjugate gradients to evaluate the mechanical behavior of composite materials. *Comput Mater Sci*. 2013;77:430-439.
42. Wicht D, Schneider M, Böhlke T. On quasi-Newton methods in FFT-based micromechanics. *Int J Numer Methods Eng*. 2020;121(8):1665-1694.

43. Chen Y, Gélébart L, Chateau C, Bornert M, Sauder C, King A. Analysis of the damage initiation in a SiC/SiC composite tube from a direct comparison between large-scale numerical simulation and synchrotron X-ray micro-computed tomography. *Int J Solid Struct*. 2019;161:111-126.
44. Wicht D, Schneider M, Böhlke T. Anderson-accelerated polarization schemes for FFT-based computational homogenization. *Int J Numer Methods Eng*. 2021;122(9):2287-2311.
45. Grimm-Strele H, Kabel M. Runtime optimization of a memory efficient CG solver for FFT-based homogenization: implementation details and scaling results for linear elasticity. *Comput Mech*. 2019;64(5):1339-1345.
46. Roters F, Diehl M, Shanthraj P, et al. 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. *Comput Mater Sci*. 2019;158:420-478.
47. Marano A, Gélébart L, Forest S. Intragranular localization induced by softening crystal plasticity: analysis of slip and kink bands localization modes from high resolution FFT-simulations results. *Acta Mater*. 2019;175:262-275.
48. Lebensohn RA, Kanjarla AK, Eisenlohr P. An elasto-viscoplastic formulation based on fast Fourier transforms for the prediction of micromechanical fields in polycrystalline materials. *Int J Plastic*. 2012;32–33:59-69.
49. Vondřejc J, Zeman J, Marek I. An FFT-based Galerkin method for homogenization of periodic media. *Comput Math Appl*. 2014;68(3):156-173.
50. Schneider M, Ospald F, Kabel M. Computational homogenization of elasticity on a staggered grid. *Int J Numer Methods Eng*. 2016;105(9):693-720.
51. Müller WH. Mathematical vs. experimental stress analysis of inhomogeneities in solids. *J Phys*. 1996;6:C1.139-C1.148.
52. Müller WH. Fourier transforms and their application to the formation of textures and changes of morphology in solids. In: Bahei-El-Din YA, Dvorak GJ, eds. *IUTAM Symposium on Transformation Problems in Composite and Active Materials*. Vol 60. Kluwer; 1998:61-72.
53. Willot F. Fourier-based schemes for computing the mechanical response of composites with accurate local fields. *Compt Rend Méc*. 2015;343:232-245.
54. Brisard S, Dormieux L. Combining Galerkin approximation techniques with the principle of Hashin and Shtrikman to derive a new FFT-based numerical method for the homogenization of composites. *Comput Methods Appl Mech Eng*. 2012;217–220:197-212.
55. Schneider M, Merkert D, Kabel M. FFT-based homogenization for microstructures discretized by linear hexahedral elements. *Int J Numer Methods Eng*. 2017;109:1461-1489.
56. Leuschner M, Fritzen F. Fourier-accelerated nodal solvers (FANS) for homogenization problems. *Comput Mech*. 2018;62:359-392.
57. Dorn C, Schneider M. Lippmann-Schwinger solvers for the explicit jump discretization for thermal computational homogenization problems. *Int J Numer Methods Eng*. 2019;118(11):631-653.
58. Vondřejc J. Improved guaranteed computable bounds on homogenized properties of periodic media by Fourier-Galerkin method with exact integration. *Int J Numer Methods Eng*. 2014;107:1106-1135.
59. Vondřejc J, Zeman J, Marek I. Guaranteed upper–lower bounds on homogenized properties by FFT-based Galerkin method. *Comput Methods Appl Mech Eng*. 2015;297:258-291.
60. Monchiet V. Combining FFT methods and standard variational principles to compute bounds and estimates for the properties of elastic composites. *Comput Methods Appl Mech Eng*. 2015;283:454-473.
61. Ladecký M, Leute RJ, Falsafilig A, et al. An optimal preconditioned FFT-accelerated filig;nite element solver for homogenization. *Appl Math Comput*. 2023;446:127835.
62. Schneider M. Lippmann-Schwinger solvers for the computational homogenization of materials with pores. *Int J Numer Methods Eng*. 2020;121(22):5017-5041.
63. Schneider M. Convergence of FFT-based homogenization for strongly heterogeneous media. *Math Method Appl Sci*. 2015;38(13):2761-2778.
64. Ye C, Chung ET. Convergence of trigonometric and finite-difference discretization schemes for FFT-based computational micromechanics. *BIT Numer Math*. 2023;63:11.
65. Saenger EH, Gold N, Shapiro SA. Modeling the propagation of elastic waves using a modified finite-difference grid. *Wave Motion*. 2000;31:77-92.
66. Saenger EH, Bohlen T. Finite-difference modeling of viscoelastic and anisotropic wave propagation using the rotated staggered grid. *Geophysics*. 2004;69:583-591.
67. Schneider M. Superaccurate effective elastic moduli via postprocessing in computational homogenization. *Int J Numer Methods Eng*. 2022;123(17):4119-4135.
68. Schneider M, Wicht D. Superconvergence of the effective Cauchy stress in computational homogenization of inelastic materials. *Int J Numer Methods Eng*. 2023;124(4):959-978.
69. Schneider M. On the effectiveness of the Moulinec-Suquet discretization for composite materials. *Int J Numer Methods Eng*. 2023;124(14):3191-3218.
70. Lebensohn RA, Rollett AD. Spectral methods for full-field micromechanical modelling of polycrystalline material. *Comput Mater Sci*. 2020;173:109336.
71. Segurado J, Lebensohn RA. Chapter one - computational homogenization of polycrystals. *Adv Appl Mech*. 2018;51:1-114.
72. Schneider M. A review of non-linear FFT-based computational homogenization methods. *Acta Mech*. 2021;232:2051-2100.
73. Gierden C, Kochmann J, Waimann J, Svendsen B, Reese S. A review of FEx02011;FFTx02011;based Twox02011;scale methods for computational Modeling of microstructure evolution and macroscopic material behavior. *Arch Comput Method Eng*. 2022;29:4115-4135.
74. Press WH, Teukolsky SA, Vetterling WT, Flannery BP. *Numerical Recipes in C - the Art of Scientific Computing*. Cambridge University Press; 2007.

75. Monchiet V, Bonnet G. FFT based iterative schemes for composite conductors with uniform boundary conditions. *Eur J Mech A Solid*. 2024;103:105146.
76. Risthaus L, Schneider M. Imposing different boundary conditions for thermal computational homogenization problems with FFT- and tensor-train-based Green's operator methods. *Int J Numer Methods Eng*. 2024;125(7):e7423.
77. Morin L, Paux J. A fast numerical method for the conductivity of heterogeneous media with dirichlet boundary conditions based on discrete sine–cosine transforms. *Comput Methods Appl Mech Eng*. 2024;421:116772.
78. Gélébart L. FFT-based simulations of heterogeneous conducting materials with combined non-uniform Neumann, periodic and Dirichlet boundary conditions. *Eur J Mech A Solid*. 2024;105:105248.
79. Wiegmann A. Fast Poisson, fast Helmholtz and fast linear elastostatic solvers on rectangular parallelepipeds. *Tech Rep Lawrence Berkeley Natl Lab*. 1999;LBNL-43565:1-21.
80. Grimm-Strele H, Kabel M. FFT-based homogenization with mixed uniform boundary conditions. *Int J Numer Methods Eng*. 2021;122:7241-7265.
81. Bödeker F, Herr P, Moshfegh R, Biel A, Marzi S. A novel FFT-based homogenization scheme for cohesive zones. *Proc Struct Integr*. 2022;42:490-497.
82. Bödeker F, Herr P, Biel A, Moshfegh R, Marzi S. An FFT-based homogenization scheme for cohesive zones with an application to adhesives and the core material of thin metal sandwich plates. *Theor Appl Fract Mech*. 2022;129:104186.
83. Gélébart L. A modified FFT-based solver for the mechanical simulation of heterogeneous materials with Dirichlet boundary conditions. *Compt Rend Méc*. 2020;348(8–9):693-704.
84. Nkoumbou Kaptchouang NB, Gélébart L. Multiscale coupling of FFT-based simulations with the LDC approach. *Comput Methods Appl Mech Eng*. 2022;394:114921.
85. Wathen A. Preconditioning. *Acta Numer*. 2015;24:329-376.
86. Risthaus L, Schneider M. Imposing Dirichlet boundary conditions directly for FFT-based computational micromechanics. *Comput Mech*. 2024; online. doi:10.1007/s00466-024-02469-1
87. Schneider M. Voxel-based finite elements with hourglass control in FFT-based computational homogenization. *Int J Numer Methods Eng*. 2022;123(24):6286-6313.
88. Park C, Sheen D. P1-nonconforming quadrilateral finite element methods for second-order elliptic problems. *SIAM J Numer Anal*. 2003;41(2):624-640.
89. Nesterov Y. *Introductory Lectures on Convex Optimization: A Basic Course. Mathematics and its applications*. Kluwer Academic Publishers; 2004.
90. Britanak V, Yip PC, Rao KR. *Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations*. Elsevier; 2010.
91. Frigo M, Johnson SG. The design and implementation of FFTW3. *Proc IEEE*. 2005;93:216-231.
92. Baratta IA, Dean JP, Dokken JS, et al. DOLFINx: the next generation FEniCS problem solving environment. *Zenodo*. 2023. doi:10.5281/zenodo.10447666
93. Lendvai J, Schneider M. Assumed strain methods in micromechanics, laminate composite voxels and level sets. *Int J Numer Methods Eng*. 2023;125:e7459.
94. Hill R. Elastic properties of reinforced solids: some theoretical principles. *J Mech Phys Solids*. 1963;11(5):357-372.
95. Flanagan DP, Belytschko T. A uniform strain hexahedron and quadrilateral with orthogonal hourglass control. *Int J Numer Methods Eng*. 1981;17:679-706.
96. Belytschko T, Ong JS-J, Liu WK, Kennedy JM. Hourglass control in linear and nonlinear problems. *Comput Methods Appl Mech Eng*. 1984;43:251-276.
97. Němeček J, Králík V, Vondřejc J. A two-scale micromechanical model for aluminium foam based on results from nanoindentation. *Comput Struct*. 2013;128:136-145.
98. Wan T, Liu Y, Zhou C, Chen X, Li Y. Fabrication, properties, and applications of open-cell aluminum foams: a review. *J Mater Sci Technol*. 2021;62:11-24.
99. Lautensack C, Zuyev S. Random Laguerre tessellations. *Adv Appl Probab*. 2008;40(3):630-650.
100. Redenbach C, Shklyar I, Andrä H. Laguerre tessellations for elastic stiffness simulations of closed foams with strongly varying cell sizes. *Int J Eng Sci*. 2012;50(1):70-78.
101. Kuhn J, Schneider M, Sonnweber-Ribic P, Böhlke T. Fast methods for computing centroidal Laguerre tessellations for prescribed volume fractions with applications to microstructure generation of polycrystalline materials. *Comput Methods Appl Mech Eng*. 2020;369:113175.
102. Kabel M, Fliegener S, Schneider M. Mixed boundary conditions for FFT-based homogenization at finite strains. *Comput Mech*. 2016;57(2):193-210.
103. Lucarini S, Segurado J. An algorithm for stress and mixed control in Galerkin-based FFT homogenization. *Int J Numer Methods Eng*. 2019;119:797-805.

How to cite this article: Risthaus L, Schneider M. FFT-based computational micromechanics with Dirichlet boundary conditions on the rotated staggered grid. *Int J Numer Methods Eng*. 2024;e7569. doi: 10.1002/nme.7569

APPENDIX A. DETAILS ON THE LIPPMANN–SCHWINGER EQUATION(S)

A.1 Derivation

The goal of this appendix is to show that the equilibrium equation (24)

$$\boldsymbol{D}^*\boldsymbol{S}(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u}) = \boldsymbol{0}, \tag{A1}$$

and the Lippmann–Schwinger equation (31)

$$\boldsymbol{u} = -\boldsymbol{G}^0\boldsymbol{D}^*\big(\boldsymbol{S}(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u}) - \alpha_0(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u})\big), \tag{A2}$$

are equivalent. We rewrite the Equation (A1) in the form

$$\alpha_0\ \boldsymbol{D}^*(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u}) = -\boldsymbol{D}^*\big(\boldsymbol{S}(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u}) - \alpha_0\ (\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u})\big), \tag{A3}$$

where we used that $\boldsymbol{D}^*$ is a linear operator. Due to the assumption of vanishing divergence (29), we deduce the equation

$$\alpha_0\ \boldsymbol{D}^*\boldsymbol{D}\boldsymbol{u} = -\boldsymbol{D}^*\big(\boldsymbol{S}(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u}) - \alpha_0\ (\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u})\big). \tag{A4}$$

Inverting the operator $\boldsymbol{D}^*\boldsymbol{D}$, made possible by Riesz' representation theorem and the assumed nondegeneracy of the inner product (26), we arrive at the equation

$$\boldsymbol{u} = -\boldsymbol{G}^0\boldsymbol{D}^*\big(\boldsymbol{S}(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u}) - \alpha_0(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u})\big). \tag{A5}$$

As every single operation we performed was an equivalence transformation, the desired equivalence of the Equations (A1) and (A2) is complete.

A.2 Equivalence of formulations

This section comprises the details on the equivalence of the Lippmann–Schwinger equation (31)

$$\boldsymbol{u} = -\boldsymbol{G}^0\boldsymbol{D}^*\big(\boldsymbol{S}(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u}) - \alpha_0(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u})\big), \tag{A6}$$

and the Lippmann–Schwinger equation (35)

$$\boldsymbol{F} = \overline{\boldsymbol{\varepsilon}} - \boldsymbol{\Gamma}^0(\boldsymbol{S}(\boldsymbol{F}) - \alpha_0\ \boldsymbol{F}). \tag{A7}$$

We start from Equation (A6), apply the operator $\boldsymbol{D}$ and add the average strain $\overline{\boldsymbol{\varepsilon}}$

$$\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u} = \overline{\boldsymbol{\varepsilon}} - \boldsymbol{D}\boldsymbol{G}^0\boldsymbol{D}^*\big(\boldsymbol{S}(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u}) - \alpha_0(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u})\big). \tag{A8}$$

Introducing the abbreviation (34) for the deformation gradient $\boldsymbol{F}$, we showed that a solution to the displacement-based equation (A6) gives rise to a solution of the deformation-gradient-based equation (A7).

Let us turn to the converse, that is, we suppose that a solution $\boldsymbol{F} \in \boldsymbol{V}$ of the Equation (A7) is given. We define the displacement fluctuation

$$\boldsymbol{u} = -\boldsymbol{G}^0\boldsymbol{D}(\boldsymbol{S}(\boldsymbol{F}) - \alpha_0\boldsymbol{F}). \tag{A9}$$

Then, Equation (A7) may be transformed as follows

$$\boldsymbol{F} = \overline{\boldsymbol{\varepsilon}} - \boldsymbol{\Gamma}^0(\boldsymbol{S}(\boldsymbol{F}) - \alpha_0\boldsymbol{F}) = \overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\big[-\boldsymbol{G}^0\boldsymbol{D}^*(\boldsymbol{S}(\boldsymbol{F}) - \alpha_0\boldsymbol{F})\big] = \overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u}. \tag{A10}$$

Thus, every solution $\boldsymbol{F}$ is actually kinematically compatible, that is, satisfies the Equation (34). To check the validity of the equilibrium equation (24) (which is by Appendix A.1 equivalent to the Lippmann–Schwinger equation (A6)), we compute

$$\boldsymbol{\Gamma}^0\boldsymbol{F} = \frac{1}{\alpha_0}\boldsymbol{D}(\boldsymbol{D}^*\boldsymbol{D})^{-1}\boldsymbol{D}^*(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u}) = \frac{1}{\alpha_0}\boldsymbol{D}(\boldsymbol{D}^*\boldsymbol{D})^{-1}\boldsymbol{D}^*\boldsymbol{D}\boldsymbol{u} = \frac{1}{\alpha_0}\boldsymbol{D}\boldsymbol{u} = \frac{1}{\alpha_0}(\boldsymbol{F} - \overline{\boldsymbol{\varepsilon}}), \tag{A11}$$

where we used the divergence constraint (29). Inserting this insight into Equation (A7) yields

$$\begin{aligned} \boldsymbol{F} &= \overline{\boldsymbol{\varepsilon}} - \boldsymbol{\Gamma}^0(\boldsymbol{S}(\boldsymbol{F}) - \alpha_0\boldsymbol{F}) \\ &= \overline{\boldsymbol{\varepsilon}} - \boldsymbol{\Gamma}^0\boldsymbol{S}(\boldsymbol{F}) + \alpha_0\,\boldsymbol{\Gamma}^0\boldsymbol{F} \\ &= \overline{\boldsymbol{\varepsilon}} - \boldsymbol{\Gamma}^0\boldsymbol{S}(\boldsymbol{F}) + \boldsymbol{F} - \overline{\boldsymbol{\varepsilon}} \\ \boldsymbol{0} &= -\boldsymbol{\Gamma}^0\boldsymbol{S}(\boldsymbol{F}). \end{aligned} \tag{A12}$$

Decoding the obtained equation

$$\boldsymbol{0} = \frac{1}{\alpha_0}\boldsymbol{D}(\boldsymbol{D}^*\boldsymbol{D})^{-1}\boldsymbol{D}^*\boldsymbol{S}(\boldsymbol{F}), \tag{A13}$$

and applying the operator $\boldsymbol{D}$ shows that the condition

$$\boldsymbol{D}^*\boldsymbol{S}(\overline{\boldsymbol{\varepsilon}} + \boldsymbol{D}\boldsymbol{u}) = \boldsymbol{0}, \tag{A14}$$

is satisfied, that is, the equilibrium equation (24) holds.

APPENDIX B. ELEMENTARY TRIGONOMETRY

B.1 Forward differences in terms of sine polynomials

We wish to show the formula (54)

$$\delta^{\mathrm{c}}u(x_j) = \sum_{k=1}^{N-1} \frac{2\sin\left(\frac{k\pi}{2N}\right)}{h}\,\hat{u}_k \cos\left(\pi\frac{\left(j+\frac{1}{2}\right)k}{N}\right), \tag{B1}$$

for the central difference (53)

$$\delta^{\mathrm{c}}u(x_j) = \frac{u\left(\frac{j+1}{N}\,L\right) - u\left(\frac{j}{N}\,L\right)}{h}, \tag{B2}$$

in terms of the sine-polynomial expression (52)

$$u(x) = \sum_{k=1}^{N-1} \hat{u}_k \sin\left(k\pi\frac{x}{L}\right), \quad x \in [0, L], \tag{B3}$$

with sine coefficients $\hat{u}_k$ $(k = 1, 2, \ldots, N-1)$.

In a first step, we insert the expression (B3) into the definition (B2) of the central difference and obtain the formula

$$\delta^{\mathrm{c}}u(x_j) = \frac{1}{h}\sum_{k=1}^{N-1} \hat{u}_k \left[\sin\left(\pi\frac{(j+1)k}{N}\right) - \sin\left(\pi\frac{jk}{N}\right)\right]. \tag{B4}$$

Let us take a look at the angle addition and subtraction theorems

$$\sin(\varphi_1 + \varphi_2) = \cos\varphi_1 \sin\varphi_2 + \sin\varphi_1 \cos\varphi_2, \tag{B5}$$

$$\sin(\varphi_1 - \varphi_2) = -\cos\varphi_1 \sin\varphi_2 + \sin\varphi_1 \cos\varphi_2, \tag{B6}$$

for the sine function and angles $\varphi_1, \varphi_2 \in \mathbb{R}$. Subtracting these formulas leads to the expression

$$\sin(\varphi_1 + \varphi_2) - \sin(\varphi_1 - \varphi_2) = 2\cos\varphi_1 \sin\varphi_2 \quad \text{for} \quad \varphi_1, \varphi_2 \in \mathbb{R}. \tag{B7}$$

With an eye toward Equation (B4), we set

$$\varphi_1 = \pi \frac{\left(j + \frac{1}{2}\right)k}{N} \quad \text{and} \quad \varphi_2 = \frac{k\pi}{2N}, \quad k = 1, 2, \ldots, N-1, \tag{B8}$$

to obtain the identity

$$\sin\left(\pi \frac{(j+1)k}{N}\right) - \sin\left(\pi \frac{jk}{N}\right) = 2\cos\left(\pi \frac{\left(j + \frac{1}{2}\right)k}{N}\right) \sin\left(\frac{k\pi}{2N}\right), \quad k = 1, 2, \ldots, N-1. \tag{B9}$$

Inserting these findings into the formula (B4), we obtain the identity

$$\delta^{\mathrm{c}} u(x_j) = \frac{2}{h} \sum_{k=1}^{N-1} \hat{u}_k \cos\left(\pi \frac{\left(j + \frac{1}{2}\right)k}{N}\right) \sin\left(\frac{k\pi}{2N}\right), \tag{B10}$$

that is, the desired expression (B1).

B.2 Averages in terms of sine polynomials

We want to establish the formula (57)

$$A^{\mathrm{c}} u(x_j) = \sum_{k=1}^{N-1} \cos\left(\frac{k\pi}{2N}\right) \hat{u}_k \sin\left(\pi \frac{\left(j + \frac{1}{2}\right)k}{N}\right), \tag{B11}$$

for the average (56)

$$A^{\mathrm{c}} u(x_j) = \frac{u\left(\frac{j+1}{N} L\right) + u\left(\frac{j}{N} L\right)}{2}, \tag{B12}$$

exploiting the representation (52)

$$u(x) = \sum_{k=1}^{N-1} \hat{u}_k \sin\left(k\pi \frac{x}{L}\right), \quad x \in [0, L], \tag{B13}$$

as a sine polynomial with sine coefficients $\hat{u}_k$ $(k = 1, 2, \ldots, N-1)$.

Inserting the expression (B13) into the definition (B12) leads to the identity

$$A^{\mathrm{c}} u(x_j) = \frac{1}{2} \sum_{k=1}^{N-1} \hat{u}_k \left[\sin\left(\pi \frac{(j+1)k}{N}\right) + \sin\left(\pi \frac{jk}{N}\right)\right]. \tag{B14}$$

Adding the angle addition and subtraction theorems (B5) for the sine function leads to the equation

$$\sin(\varphi_1 + \varphi_2) + \sin(\varphi_1 - \varphi_2) = 2\sin\varphi_1 \cos\varphi_2. \tag{B15}$$

Keeping the Equation (B14) in mind, we set

$$\varphi_1 = \pi \frac{\left(j + \frac{1}{2}\right)k}{N} \quad \text{and} \quad \varphi_2 = \frac{k\pi}{2N}, \quad k = 1, 2, \ldots, N-1, \tag{B16}$$

to obtain the formula

$$\sin\left(\frac{(j+1)k}{N}\right) + \sin\left(\frac{jk}{N}\right) = 2\sin\left(\pi \frac{\left(j + \frac{1}{2}\right)k}{N}\right)\cos\left(\frac{k\pi}{2N}\right), \quad k = 1, 2, \ldots, N-1. \tag{B17}$$

Inserting this insight into Equation (B14), we are led to the formula

$$A^{\mathrm{c}}u(x_j) = \sum_{k=1}^{N-1} \hat{u}_k \sin\left(\pi \frac{\left(j + \frac{1}{2}\right)k}{N}\right)\cos\left(\frac{k\pi}{2N}\right), \tag{B18}$$

that was to be shown (B11).

DOI: 10.1002/nme.7569
URN: urn:nbn:de:hbz:465-20240718-151744-2

Early View. Online Version of Record before inclusion in an issue. First published: 18 July 2024