RESEARCH ARTICLE

WILEY

Voxel-based finite elements with hourglass control in fast Fourier transform-based computational homogenization

Matti Schneider

Institute of Engineering Mechanics,
Karlsruhe Institute of Technology (KIT),
Karlsruhe, Germany

Correspondence
Matti Schneider, Institute of Engineering
Mechanics, Karlsruhe Institute of
Technology (KIT), Karlsruhe, Germany.
Email: matti.schneider@kit.edu

Abstract

The power of fast Fourier transform (FFT)-based methods in computational micromechanics critically depends on a seamless integration of discretization scheme and solution method. In contrast to solution methods, where options are available that are fast, robust and memory-efficient at the same time, choosing the underlying discretization scheme still requires the user to make compromises. Discretizations with trigonometric polynomials suffer from spurious oscillations in the solution fields and lead to ill-conditioned systems for complex porous materials, but come with rather accurate effective properties for finitely contrasted materials. The staggered grid discretization, a finite-volume scheme, is devoid of bulk artifacts in the solution fields and works robustly for porous materials, but does not handle anisotropic materials in a natural way. Fully integrated finite-element discretizations share the advantages of the staggered grid, but involve a higher memory footprint, require a higher computational effort due to the increased number of integration points and typically overestimate the effective properties. Most widely used is the rotated staggered grid discretization, which may also be viewed as an underintegrated trilinear finite element discretization, which does not impose restrictions on the constitutive law, has fewer artifacts than Fourier-type discretizations and leads to rather accurate effective properties. However, this discretization comes with two downsides. For a start, checkerboard artifacts are still present. Second, convergence problems occur for complex porous microstructures. The work at hand introduces FFT-based solution techniques for underintegrated trilinear finite elements with hourglass control. The latter approach permits to suppress local hourglass modes, which stabilizes the convergence behavior of the solvers for porous materials and removes the checkerboards from the local solution field. Moreover, the hourglass-control parameter can be adjusted to "soften" the material response compared to fully integrated elements, using only a single integration point for nonlinear analyses at the same time. To be effective, the introduced technology requires a displacement-based implementation. The article exposes an efficient

way for doing so, providing minimal interfaces to the most commonly used solution techniques and the appropriate convergence criterion.

1 | INTRODUCTION

1.1 | State of the art

Micromechanics is based on the paradigm that the mechanical response of a heterogeneous material emerges from the knowledge of both the microstructure and the mechanical behavior of the individual phases. The microstructure itself serves as a construction plan for the heterogeneous material under consideration. Modern digital image processing methods[1-5] enable a detailed view on these microstructures. However, they also show their inherent complexity. Moreover, these digital images typically come in the form of rather large three-dimensional voxel (volume pixel) images.

Using this data as input for computational homogenization comes with specific challenges. One strategy[6-8] uses each voxel as a finite element (FE), leading to rather large discrete systems to be solved. However, the regular structure of the grid permits a number of optimizations[9-11] to be exploited. Another strategy uses a coarsening of the mesh to estimate the interfaces present between the materials.[12-15] Then, either a conforming mesh may be generated[16,17] or interface-aware finite-element strategies[18-20] may be used. On the one hand, it is typically challenging to automatically generate a high-quality conforming mesh[21] due to the inherent complexity of the microstructures at hand. On the other hand, interface-aware finite elements may suffer from ill-conditioning.[22-24]

Within the class of voxel-based discretizations, the approach put forward by Moulinec and Suquet[25,26] turned out to be particularly efficient. This strategy is based on the fast Fourier transform (FFT) and serves as the foundation of FFT-based computational micromechanics. The original method[25,26] has a number of characteristics. For a start, it is based on the Lippmann–Schwinger equation,[27-29] a reformulation of the micromechanical problem in terms of an equivalent integral equation. Second, it is based on regular grids with periodic boundary conditions. Third, it uses the FFT to evaluate the action of the Green's operator for strains, sometimes called Eshelby–Green operator. Last but not least, it supported inelastic material laws from the very beginning.

In their original formulation, Moulinec and Suquet did not clearly distinguish between the discretization method and the solution technique. In subsequent years, a number of improved solution methods was introduced, including polarization methods,[30-32] Krylov subspace methods[33-35] for linear problems, fast gradient methods[36-38] and Newton[39-41] as well as Quasi-Newton methods[42-44] and a transition to quantum computers[45].

Parallel to the developments in solver technology, efforts were directed to both understanding and extending the discretization method used by Moulinec and Suquet.[25,26] Early attempts include regarding the discretization as a discretization of an integral equation.[34,35,46] However, the most straightforward interpretation was given by Vondřejc et al.[47] They showed that a discretization by trigonometric polynomials together with a simple quadrature by the trapezoidal rule of the average elastic energy gives rise to the discretization used by Moulinec and Suquet.[25,26] The basic solution scheme of Moulinec and Suquet emerges then naturally as a preconditioned Richardson iteration[48] or, equivalently, via a (projected) gradient descent.[41,49] Alternative discretization schemes were studied to replace this discretization with the hope to improve the quality of both the local solution fields and the accuracy of the computed effective properties. These alternatives include fully integrated trigonometric polynomials,[50,51] voxel-wise constant fields,[34,35] finite-difference,[52-54] finite-volume[55-57] and finite-element[58-60] discretizations. More recently, higher-order discretizations[61,62] were also studied.

To be effective, the solution strategy and the discretization scheme need to be both well-aligned and closely cooperated. Unfortunately, each of these discretization schemes come with specific disadvantages which makes them suitable for a subset of problems, but not all of them. In recent years, research effort was devoted to identify a subset of these pairs consisting of discretization scheme and solution method which work reasonably well on large classes of micromechanical problems, freeing the user of the daunting task of selecting the appropriate combination.

TABLE 1 Material parameters used in this article, with source

Material	Source	E in GPa	ν	Inelastic parameters	
Aluminum	[99]	70	0.3		
E-glass fibers	[100]	72	0.22		
Polymer matrix	[100]	3.0	0.35	$\sigma_Y = 20$ MPa	$k_1 = 1$ MPa
				$k_2 = 15$ MPa	$m = 150$
Quartz sand grains	[101,102]	66.9	0.25		
Quartz glass binder	[103]	71.7	0.17		

Discretizations based on trigonometric polynomials[47,50,51] suffer from ringing artifacts and lead to ill-conditioned discrete systems for complex porous materials. However, they are particularly well-suited to low-contrast materials like polycrystalline materials and permit computing two-sided bounds of the effective elastic moduli if full integration is used. For discretizations based on the Hashin–Shtrikman variational principle,[34,35,62] the computed effective properties depend on the reference material, requiring the latter to be chosen in a salient way. The finite-difference discretization on a staggered grid[55] comes with a stable convergence behavior for porous materials and features local solution fields devoid of artifacts. However, the staggered placement of the variables makes working with anisotropic material laws challenging and may further lead to inaccuracies at inter- and surfaces. The most widely used discretization nowadays is the discretization on a rotated staggered grid,[63,64] introduced into FFT-based computational micromechanics by Willot.[52] It shares the simplicity of implementation with the original Moulinec–Suquet discretization, but converges for a large class of porous materials. On the downside, the rotated staggered grid comes with checkerboard artifacts in the solution fields and fails to converge for some complex porous microstructures, see Schneider[65] or Grimm-Strele and Kabel[66(5.5)]. Subsequently, finite element discretizations on a regular grid were integrated into the FFT framework, essentially unaware of previous works.[67-69] This integration came in two flavors. Schneider et al.[58] provided a strain-based implementation where the strains at the integration points serve as the unknowns. Interestingly, the same study pointed out that the trilinear hexahedron element with one-point integration and the discretization on a staggered grid give rise to identical discretized systems. Such underintegrated brick elements are well-known to suffer from hourglassing effects.[70-72] This insight sheds light on the sources of the non-convergence of solvers when the rotated staggered grid discretization is used for complex microstructures with pores. Indeed, the local hourglass modes, associated to zero-energy deformations of the *element*, may be combined to global hourglass modes, giving rise non-trivial zero-energy deformations of the *structure*, which are not present on a regular grid of solid materials. Only the latter get filtered out by the FFT preconditioner. The approach of Leuschner–Fritzen[59] avoids the strain-based perspective, and is based on a pure displacement implementation. The strategy follows classical finite-element guidelines, assembling both the FE stiffness matrix *and* the preconditioner in Fourier space. As a consequence, for linear elastic materials, the computational effort that comes with multiple integration points is avoided, at the expense of a much higher memory footprint. Recently, Ladecký et al.[60] proposed to evaluate the action of the FE stiffness matrix on the fly.

1.2 | The problem with fully integrated voxel finite elements

Finite-element discretizations are the golden standard when it comes to computational solid mechanics. They are also ubiquitous in computational micromechanics, see, for example, the review article by Matouš et al.[73] In this light, it appears very paradoxical that finite elements did not enter the mainstream of FFT-based computational micromechanics, despite being introduced more than half a decade before. It is the purpose of this section to shed some light on the underlying reasons.

Let us consider different finite-element discretizations on a regular grid with periodic boundary conditions. As a benchmark, we take a rather simple single spherical inclusion at 12.9% volume fraction. We furnish the inclusion with the elastic parameters of E-glass, and consider a polymer matrix with isotropic elastic moduli, see Table 1 in Section 4.

In Figure 1A, the *xx*-components of effective stresses under 5% uniaxial extension are shown for increasing resolution and different discretization methods. Both the Moulinec–Suquet discretization and the staggered grid give rise to rather similar predictions. The rotated staggered grid, referred to as HEX8R thereafter, predicts a slightly stiffer response. Yet, we
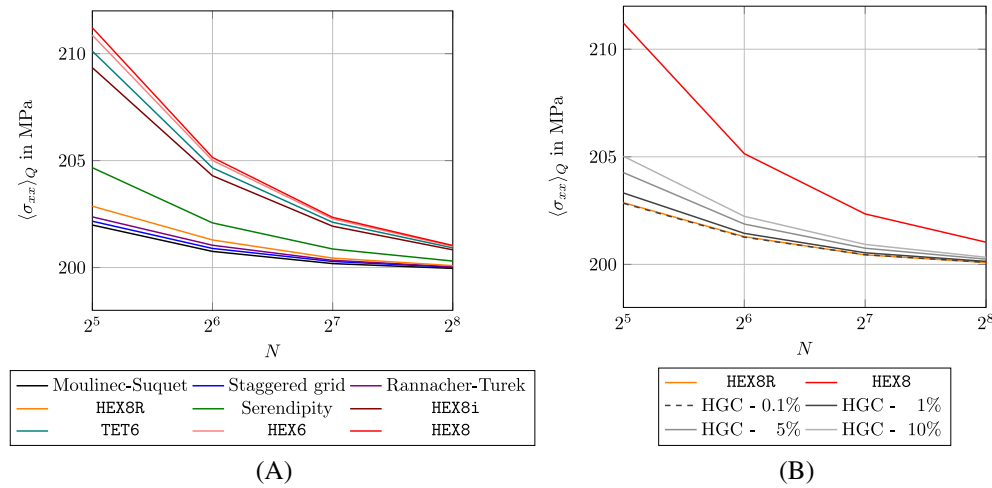
FIGURE 1 Comparison of discretization methods classically used in FFT-based computational micromechanics, different fully integrated and underintegrated voxel finite elements and trilinear HEX elements with hourglass control (HGC) for a single spherical glass inclusion in a polymer matrix and different resolutions N^3. (A) Classical discretizations; (B) different hourglass parameters
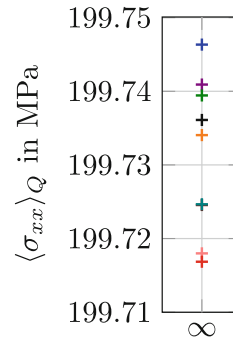

FIGURE 2 Extrapolated effective stresses

observe a convergence upon grid refinement to a common value. When considering the fully integrated hexahedron element (HEX8), we observe a strong overestimation of the effective stresses. Even for the finest resolution considered with 256^3 voxels, there is a strong discrepancy between HEX8 and the classical discretizations used in FFT-based methods. Neither using six integration points (HEX6), located on the faces of the element, nor relying upon a subdivision into six tetrahedra[74] (TET6) does improve the situation. Including incompatible modes[75] (HEX8i) leads to a marginal improvement. Even using serendipity elements,[76] a type of quadratic elements with eight integration points, lead to a much smaller error. However, computing the FFT preconditioner for these elements is significantly more expensive. Indeed, twenty nodes need to be considered for each voxel, leading to an inversion of a 12×12-matrix at each frequency in Fourier space. The only finite element which appears to be competitive to classical FFT-based discretizations is the Rannacher–Turek element,[77] also called the rotated Q_1-element. Interestingly, the Rannacher–Turek element was introduced to handle problems of fluid flow, that is, may handle incompressibility naturally. In this light, it shares some similarities with the finite-difference discretization on a staggered grid. However, the unknowns in the Rannacher–Turek element are located on the element faces. As a consequence, when constructing the preconditioner, a complex 9×9-matrix needs to be inverted instead of a 3×3-matrix for HEX8. Also, a displacement-based implementation of the Rannacher–Turek element requires to store nine degrees of freedom per voxel. In particular, turning the Rannacher–Turek element into an efficient alternative requires further work.

Figure 2 shows the stress values extrapolated from the values $N \in \{128, 256\}$

assuming a linear rate of convergence, using the coloring scheme of Figure 1A. We observe that these extrapolated values coincide for the first four significant digits. Thus, we may confirm the theorical considerations[35,46,65] that all of these considerations converge to the same value.

In addition to the lack of accuracy associated with these investigated finite-element discretizations, dealing with inelastic and nonlinear constitutive laws infers a further computational overhead due to the number of integration points per voxel. Indeed, for the considered discretizations, at least six integration points are necessary per element. Classical discretizations used in FFT-based micromechanics operate on a single integration point. If evaluating the constitutive law is expensive, the overhead induced by multiple integration points may be significant. Moreover, storing the FE tangent necessary for applying Newton's method also becomes rather costly. Indeed, one may either store the full FE tangent matrix or store the material tangents for all integration points. Both options appear unwise when memory footprint is a concern, a common scenario for industrial-scale microstructures.

1.3 | Contributions

For FFT-based computational micromechanics, the discretization scheme and the solution method need to be selected in a synergetic way. For the solution methods, it is possible to balance convergence speed and memory consumption as desired, see Schneider[78(3)] for a discussion. However, for the discretization schemes, the situation is less favorable. As discussed previously, the currently available candidates all come with some shortcomings. The purpose of this article is to propose an improvement of the currently most popular discretization used in FFT-based computational micromechanics: the rotated staggered grid.[52]

We interpret the latter as underintegrated trilinear finite elements, HEX8R, and attribute its shortcomings, the lack of convergence for complex porous materials and the artifacts in the solution fields, to a lack of hourglass control. Indeed, it is well-known in the FE community[70-72] that some underintegrated finite elements lead to element force-displacement matrices which do not have full rank. Deformation modes in the kernel of the element force-displacement matrix are called *hourglass modes*. The article at hand proposes to improve the HEX8R element by hourglass control.

In addition to curing the previously managed shortcomings of HEX8R, using hourglass control does also alleviate the problems of other finite-element discretizations described in Section 1.2. Indeed, by adjusting the hourglass-control parameter $\rho \in (0, 1]$, it is possible to eliminate the overly stiff response of fully integrated finite elements, see Figure 1B, for the single-inclusion example. Moreover, as hourglass-controlled hexahedron elements use only a single integration point, such a strategy also comes with benefits in terms of runtime.

Integrating hourglass control into existing strain-based FFT solvers for micromechanics is not trivial. Indeed, the strategy proposed in Schneider et al.[58] is intrinsically linked to multiple integration points, and accounting for extra hourglass control is far from straightforward (not to mention efficient). Therefore, it appears wise to work with a displacement-based implementation, as pioneered by Leuscher and Fritzen[59] for finite elements in the FFT community. This change to a displacement point-of-view makes it necessary to transfer some technology available for strain-based solvers, leading to additional (minor) contributions. Indeed, this article reports on a consistent use of mixed boundary conditions in a displacement-based setting (compare also Lucarini and Segurado[79,80]), uses only a single type of implementation to deal with both descendants of the basic scheme and conjugate-gradient (CG) methods, and establishes the consistent convergence criterion well-known in a strain setting[44,49,81] to the displacement setting.

This work is organized as follows. Section 2 discusses the finite-element discretization (specialized to HEX8-type elements) in a concise way and provides a streamlined presentation of hourglass control in general form. In Section 3, equivalents of the most powerful solvers[78(3)] for the strain setting are expressed in a form amenable to integration into a displacement-based code. In this way, rather fast convergence may be reached for inelastic problems without having to resort to Newton-type methods that require storing the tangent matrix. Last but not least, Section 4 provides a number of computational investigations that shed further light on the introduced ideas.

2 | DISCRETIZATION SCHEMES

2.1 | The homogenization problem

We are concerned with a cuboid cell

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

in three spatial dimensions. We assume that a heterogeneous stress operator

$$\boldsymbol{\sigma} : Y \times \mathrm{Sym}(3) \to \mathrm{Sym}(3), \tag{2}$$

is given, which maps, for every point $\boldsymbol{x} \in Y$, a strain tensor $\boldsymbol{\varepsilon}$ to the corresponding stress tensor $\boldsymbol{\sigma}(\boldsymbol{x}, \boldsymbol{\varepsilon})$. Typically, the stress operator (2) arises from an inelastic problem after a discretization in time and a static condensation of the internal variables when a single load step is considered.[82-84]

Additionally, we assume a mixed macroscopic loading[79,85,86] to be given in terms of a strain tensor $\overline{\boldsymbol{\varepsilon}} \in \mathrm{Sym}(3)$ and a stress tensor $\overline{\boldsymbol{\sigma}} \in \mathrm{Sym}(3)$, together with two orthogonal and complementary tensors $\mathbb{P}$ and $\mathbb{Q}$, s.t. the conditions

$$\mathbb{P} : \overline{\boldsymbol{\varepsilon}} = \overline{\boldsymbol{\varepsilon}} \quad \text{and} \quad \mathbb{Q} : \overline{\boldsymbol{\sigma}} = \overline{\boldsymbol{\sigma}}, \tag{3}$$

hold. Following Suquet,[87] we seek a pair $(\boldsymbol{E}, \boldsymbol{u}) \in \mathrm{Sym}(3) \times H^1_{\#}(Y; \mathbb{R}^3)$, consisting of a macroscopic strain $\boldsymbol{E}$ and a periodic displacement fluctuation field $\boldsymbol{u}$ with square-integrable derivatives, s.t. the following conditions

$$\mathbb{P} : \boldsymbol{E} = \overline{\boldsymbol{\varepsilon}}, \tag{4}$$

$$\mathbb{Q} : \langle \boldsymbol{\sigma}(\cdot, \boldsymbol{E} + \boldsymbol{\nabla}^s \boldsymbol{u}) \rangle_Y = \overline{\boldsymbol{\sigma}}, \tag{5}$$

$$\int_Y \boldsymbol{\nabla}^s \boldsymbol{w} : \boldsymbol{\sigma}(\boldsymbol{x}, \boldsymbol{E} + \boldsymbol{\nabla}^s \boldsymbol{u}) \, d\boldsymbol{x} = 0, \tag{6}$$

hold for all $\boldsymbol{w} \in H^1_{\#}(Y; \mathbb{R}^3)$. Here, we denote by

$$\langle \psi \rangle_Y = \frac{1}{\mathrm{vol}(Y)} \int_Y \psi(\boldsymbol{x}) \, d\boldsymbol{x}, \tag{7}$$

the mean value of a scalar field $\psi \in L^1(Y)$, which is extended to vector and tensor quantities component-wise, and $\boldsymbol{\nabla}^s$ refers to the symmetrized gradient operator.

The first Equation (4) ensures that the $\mathbb{P}$-part of the strain $\boldsymbol{E}$ equals the prescribed macroscopic strain $\overline{\boldsymbol{\varepsilon}}$. The second Equation (5) forces the $\mathbb{Q}$-part of the average stress to coincide with the prescribed macroscopic stress $\overline{\boldsymbol{\sigma}}$. The last Equation (6) encodes (quasi-)static equilibrium without microscopic body forces. Under mild conditions on the stress operator (2), existence and uniqueness of solutions of the Equations (4)–(6) can be proved in the class of displacement fields with vanishing mean, for example, if the operator $\boldsymbol{\sigma}$ satisfies suitable continuity and monotonicity properties.[46,88] Additionally, it is not uncommon in mechanics to study problems without unique solutions, for example, when softening material behavior is investigated. We will not rule out such a case for the paper at hand, as we are mainly interested in discretization techniques which admit an efficient preconditioning in a natural way.

Once the homogenization problem (4)–(6) has been solved, the average strain tensor $\boldsymbol{E}$ and the average stress tensor $\langle \boldsymbol{\sigma}(\boldsymbol{E} + \boldsymbol{\nabla}^s \boldsymbol{u}) \rangle_Y$ are the primary quantities of interest. Oftentimes the full strain tensor $\overline{\boldsymbol{\varepsilon}}$ is prescribed, that is, $\mathbb{Q} \equiv 0$ holds. Then, the average stress tensor $\langle \boldsymbol{\sigma}(\boldsymbol{E} + \boldsymbol{\nabla}^s \boldsymbol{u}) \rangle_Y$ is of sole interest. However, when a comparison to experimental results is desired, mixed boundary conditions are often more appropriate, for example, for uni- or bi-axial loading.[79,85]

It will turn out that the general formulation (4)–(6) is beneficial for FFT-based solvers even in the case $\mathbb{Q} \equiv 0$, see Section 3.3.

2.2 | Discretization by finite elements

Suppose the domain Y is decomposed into a Cartesian grid with $N \equiv N_1 \times N_2 \times N_3$ hexahedron finite elements,[89] that is, N_i hexahedra in the ith coordinate direction ($i = 1, 2, 3$). We suppose that a generic FE function $\boldsymbol{v} \in H^1_{\#}(Y; \mathbb{R}^3)$ can be written in the form

$$\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{N}(\boldsymbol{x}) \, \underline{v} \tag{8}$$

with the matrix-valued field $\boldsymbol{N} : Y \to \mathbb{R}^{3 \times 3N}$ that collects all shape functions and the nodal degrees of freedom $\underline{v} \in \mathbb{R}^{3N}$. In this manuscript, we will use trilinear shape functions for definiteness. However, the technique is applicable to other

(conforming and non-conforming) finite elements, as long as a Cartesian grid is considered. We will use Mandel's notation[90,91] to convert between stress/strain tensors and vectors in $\mathbb{R}^6$ implicitly. For instance, the element $\underline{\bar{\varepsilon}}$ corresponds to the macroscopic strain $\bar{\varepsilon}$ in Mandel's notation. A similar conversion applies to fourth-order tensors like $\mathbb{P}$, which are converted to matrices $\underline{\underline{P}}$. Then, the strain-displacement matrices $\underline{\underline{B}} : Y \to \mathbb{R}^{6 \times 3N}$ are implicitly defined via the identity

$$\underline{\underline{B}}(\boldsymbol{x}) = \underline{\boldsymbol{\nabla}^s \boldsymbol{N}(\boldsymbol{x})}. \tag{9}$$

With this notation at hand, we may insert the fields

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{N}(\boldsymbol{x}) \, \underline{u} \quad \text{and} \quad \boldsymbol{w}(\boldsymbol{x}) = \boldsymbol{N}(\boldsymbol{x}) \, \underline{w}, \tag{10}$$

into Equation (6) to yield

$$\int_Y \underline{w}^T \underline{\underline{B}}^T \underline{\sigma} \left(\boldsymbol{x}, \underline{E} + \underline{\underline{B}} \, \underline{u} \right) \, d\boldsymbol{x} = 0. \tag{11}$$

If we enforce this equation to hold for arbitrary $\underline{w} \in \mathbb{R}^{3N}$, we may equivalently seek $\underline{u} \in \mathbb{R}^{3N}$ to solve the (vector) equation

$$\int_Y \underline{\underline{B}}^T \underline{\sigma} \left(\boldsymbol{x}, \underline{E} + \underline{\underline{B}} \, \underline{u} \right) \, d\boldsymbol{x} = 0. \tag{12}$$

Thus, the (fully integrated) finite element approximation (8) seeks a pair $(\underline{E}, \underline{u}) \in \mathbb{R}^6 \times \mathbb{R}^{3N}$, s.t. the equations

$$\underline{\underline{P}} \, \underline{E} = \underline{\bar{\varepsilon}}, \tag{13}$$

$$\underline{\underline{Q}} \left\langle \underline{\sigma} \left(\cdot, \underline{E} + \underline{\underline{B}} \, \underline{u} \right) \right\rangle_Y = \underline{\bar{\sigma}}, \tag{14}$$

$$\int_Y \underline{\underline{B}}^T \underline{\sigma} \left(\boldsymbol{x}, \underline{E} + \underline{\underline{B}} \, \underline{u} \right) \, d\boldsymbol{x} = 0, \tag{15}$$

hold. It is more convenient to use a divide-and-conquer strategy to transform this global description into a local version that is easier to manage. For this purpose, we consider each finite element Y_e, $e = 1, \dots, N$, individually, and consider:

- The local strain-displacement matrix $\underline{\underline{B_e}} : Y \to \mathbb{R}^{6 \times 24}$, which vanishes outside of the element Y_e and associates the strain inside this element to the current nodal values.
- An extraction matrix $\underline{\underline{\Lambda_e}} \in \mathbb{R}^{24 \times 3N}$, a Boolean matrix, which associates the global nodal degrees of freedom with the local nodal degrees of freedom. More precisely, $(\Lambda_e)_{ij} = 1$ if the ith local degree of freedom corresponds to the jth global degree of freedom.
- An element-local stress operator $\underline{\sigma_e} : \mathbb{R}^6 \to \mathbb{R}^6$ which satisfies the consistency condition

$$\underline{\sigma}(\boldsymbol{x}, \underline{\varepsilon}) = \underline{\sigma_e}(\underline{\varepsilon}) \quad \text{for all} \quad \boldsymbol{x} \in Y_e, \quad \underline{\varepsilon} \in \mathbb{R}^6. \tag{16}$$

In particular, we suppress the dependence on the previous time step for notational clarity.

As a consequence of these considerations, the division identity

$$\underline{\underline{B}}(\boldsymbol{x}) = \sum_{e=1}^N \underline{\underline{B_e}}(\boldsymbol{x}) \underline{\underline{\Lambda_e}}, \tag{17}$$

holds which permits us to rewrite Equation (15) in the divide-and-conquer form

$$\sum_{e=1}^N \int_{Y_e} \underline{\underline{\Lambda_e}}^T \underline{\underline{B_e}}^T \underline{\sigma_e} \left(\underline{E} + \underline{\underline{B_e}} \, \underline{\underline{\Lambda_e}} \, \underline{u} \right) \, d\boldsymbol{x} = 0. \tag{18}$$

We split the latter equation into the more convenient form

$$\sum_{e=1}^{N} \underline{\underline{\Lambda_e}}^T \underline{g}_e \left(\underline{E}, \underline{\underline{\Lambda_e}} \, \underline{u} \right) = 0 \tag{19}$$

with the function $\underline{g}_e : \mathbb{R}^6 \times \mathbb{R}^{24} \to \mathbb{R}^{24}$ defined as

$$\underline{g}_e(\underline{E}, \underline{u}_e) = \int_{Y_e} \underline{\underline{B_e}}^T \underline{\sigma}_e \left(\underline{E} + \underline{\underline{B_e}} \, \underline{u}_e \right) \, d\mathbf{x}, \tag{20}$$

which computes the equivalent nodal forces on the element Y_e for prescribed nodal displacements $\underline{u}_e$. For a general nonlinear stress operator, evaluating the integral (20) is computationally expensive. Instead, approximated variants of the function $\underline{g}_e$ are used based on numerical integration with a number q of integration points inside each element Y_e. Therefore, we consider expressions of the form

$$\underline{g}_e^{\text{appr}}(\underline{E}, \underline{u}_e) = \sum_{i=1}^{q} w_i \, \underline{\underline{B_e}}(\mathbf{x}_e^i)^T \underline{\sigma}_e \left(\underline{E} + \underline{\underline{B_e}}(\mathbf{x}_e^i) \, \underline{u}_e \right) \tag{21}$$

with a number $w_1, \ldots, w_q$ of positive integration weights, which sum to unity, and appropriate integration points $\mathbf{x}_e^i \in Y_e$ ($i = 1, \ldots, q$). The classical trilinear hexahedron elements with eight integration points based on product Gaussian quadrature arise this way, and we denote the corresponding function by $\underline{g}_e^{\text{HEX8}}$. Similarly, when the integration points are placed on the centers of the six faces, we arrive at the function $\underline{g}_e^{\text{HEX6}}$, see Dunavant.[92] Using only a single integration point at the center of the cell produces trilinear finite elements with reduced integration, leading to the function $\underline{g}_e^{\text{HEX8R}}$.

In case of linear elastic material behavior, a few simplifications apply. So let us suppose that in the element Y_e, the stress operator can be written in the form

$$\underline{\sigma}_e(\underline{\varepsilon}) = \underline{\underline{C_e}} \, \underline{\varepsilon}, \quad \underline{\varepsilon} \in \mathbb{R}^6, \tag{22}$$

with a symmetric positive definite stiffness matrix $\underline{\underline{C_e}} \in \mathbb{R}^{6 \times 6}$. Then, the expression (21) becomes

$$\underline{g}_e^{\text{appr}}(\underline{E}, \underline{u}_e) = \sum_{i=1}^{q} w_i \, \underline{\underline{B_e}}(\mathbf{x}_e^i)^T \underline{\underline{C_e}} \left(\underline{E} + \underline{\underline{B_e}}(\mathbf{x}_e^i) \, \underline{u}_e \right), \tag{23}$$

which we may also write in the more convenient form

$$\underline{g}_e^{\text{appr}}(\underline{E}, \underline{u}_e) = \underline{f}_e + \underline{\underline{K_e}}^{\text{appr}} \, \underline{u}_e \tag{24}$$

with the abbreviations $\underline{f}_e \in \mathbb{R}^{24}$ and $\underline{\underline{K_e}}^{\text{appr}} \in \mathbb{R}^{24 \times 24}$ defined by

$$\underline{f}_e = \sum_{i=1}^{q} w_i \, \underline{\underline{B_e}}(\mathbf{x}_e^i)^T \underline{\underline{C_e}} \, \underline{E} \quad \text{and} \quad \underline{\underline{K_e}}^{\text{appr}} = \sum_{i=1}^{q} w_i \, \underline{\underline{B_e}}(\mathbf{x}_e^i)^T \underline{\underline{C_e}} \, \underline{\underline{B_e}}(\mathbf{x}_e^i). \tag{25}$$

Please note that for fixed stiffness $\underline{\underline{C_e}}$, the expressions for the nodal forces $\underline{f}_e$ induced by the macroscopic strain coincide for the three considered discretizations. However, the element-stiffness matrices $\underline{\underline{K_e}}^{\text{appr}}$ differ, leading to distinct matrices $\underline{\underline{K_e}}^{\text{HEX8}}$, $\underline{\underline{K_e}}^{\text{HEX6}}$, and $\underline{\underline{K_e}}^{\text{HEX8R}}$, in general. The apparent advantage of the representation (24) is that, if only a reasonable number of different stiffness matrices C_e occur, the pairs $\left(\underline{f}_e, \underline{\underline{K_e}}^{\text{appr}} \right)$ can be precomputed and cached. In particular, the computational effort for evaluating the expression (24) is *independent* of the number of integration points.[59] This situation

contrasts with the general (physically) nonlinear case (20), as the nonlinear function $\underline{\sigma}_e$ needs to be evaluated at each of the q integration points, which may be rather costly.

After dealing with Equation (15), we study the condition (14) for the mean stress more closely. Arguing as for the equilibrium condition, we deduce the identity

$$\left\langle \underline{\sigma}(\cdot, \underline{E} + \underline{\underline{B}}\, \underline{u}) \right\rangle_Y = \frac{1}{N} \sum_{e=1}^{N} \frac{1}{\text{vol}(Y_e)} \int_{Y_e} \underline{\sigma}_e \left(\underline{E} + \underline{\underline{B}}_e\, \underline{u}_e \right) \, d\boldsymbol{x}. \tag{26}$$

When evaluating the latter integral, similar problems arise as for the expression (20). Therefore, we use the approximation

$$\left\langle \underline{\sigma}(\cdot, \underline{E} + \underline{\underline{B}}\, \underline{u}) \right\rangle_Y \approx \frac{1}{N} \sum_{e=1}^{N} \sum_{i=1}^{q} w_i\, \underline{\sigma}_e \left(\underline{E} + \underline{\underline{B}}_e(\boldsymbol{x}_e^i)\, \underline{u}_e \right), \tag{27}$$

consistent to the rule (21). Once the stresses

$$\underline{\sigma}_e^i = \underline{\sigma}_e(\underline{E} + \underline{\underline{B}}_e(\boldsymbol{x}_e^i)\, \underline{u}_e), \quad i = 1, \dots, q, \tag{28}$$

have been evaluated for each integration point during the computation of the expression (21), computing

$$\sum_{i=1}^{q} w_i\, \underline{\sigma}_e(\underline{E} + \underline{\underline{B}}_e(\boldsymbol{x}_e^i)\, \underline{u}_e) = \sum_{i=1}^{q} w_i\, \underline{\sigma}_e^i, \tag{29}$$

comes with minimal additional computational cost. For linear elastic constitutive behavior (22), it is more convenient to write

$$\sum_{i=1}^{q} w_i\, \underline{\sigma}_e(\underline{E} + \underline{\underline{B}}_e(\boldsymbol{x}_e^i)\, \underline{u}_e) = \underline{\underline{C}}_e\, \langle \underline{\varepsilon} \rangle_e \tag{30}$$

with the element-averaged strain

$$\langle \underline{\varepsilon} \rangle_e = \overline{\underline{\varepsilon}} + \underline{\underline{A}}_e\, \underline{u}_e \quad \text{and} \quad \underline{\underline{A}}_e = \sum_{i=1}^{q} w_i\, \underline{\underline{B}}_e(\boldsymbol{x}_e^i). \tag{31}$$

Please note that the matrix $\underline{\underline{A}}_e$ may also be used to express the nodal force vector (25)

$$\underline{f}_e = \underline{\underline{A}}_e^T \underline{\underline{C}}_e\, \underline{E}. \tag{32}$$

We close this section with a few remarks.

1. For a prescribed approximation scheme, we thus seek solutions to the equations

$$\underline{\underline{P}}\, \underline{E} = \overline{\underline{\varepsilon}}, \tag{33}$$

$$\underline{\underline{Q}}\, \underline{S}(\underline{E}, \underline{u}) = \overline{\underline{\sigma}}, \tag{34}$$

$$\underline{r}(\underline{E}, \underline{u}) = 0, \tag{35}$$

with the vector residual

$$\underline{r}(\underline{E}, \underline{u}) = \sum_{e=1}^{N} \underline{\underline{\Lambda}}_e^T \underline{g}_e^{\text{appr}} \left(\underline{E}, \underline{\underline{\Lambda}}_e\, \underline{u} \right), \tag{36}$$

and the average stress associated to a displacement fluctuation field

$$\underline{S}(\underline{E}, \underline{u}) = \frac{1}{N} \sum_{e=1}^{N} \sum_{i=1}^{q} w_i \; \underline{\underline{\sigma}}_e(\underline{E} + \underline{\underline{B}}_e(\boldsymbol{x}_e^i) \; \underline{\underline{\Lambda}}_e \; \underline{u}). \tag{37}$$

2. The conceptual Boolean matrices $\underline{\underline{\Lambda}}_e$, which are never assembled, describe how to evaluate the force vector (35) in a matrix-free way, as discussed by Willot et al.[93(7)] and Ladecký et al.[60(5.1)] in more detail. During the processing of each voxel, it is necessary to evaluate the element-averaged stress (35). It is convenient to compute the element-averaged strain (31), as well, as it is a quantity of interest for the postprocessing, for instance when writing out the solution fields.

    Let us remark that such an element-based strategy is more computationally demanding than strain-based implementations[40,52] if only a single integration point is considered. Indeed, for linear elastic problems, in strain-based implementations, a 6×6 matrix needs to be applied per voxel to transform the strain into a stress. In contrast, for the displacement-based implementation at hand, a 24×24-matrix needs to be applied to convert the eight displacements at the nodes to the eight forces at the eight nodes. As working with displacements instead of strains comes with certain other benefits, for instance fewer Fourier transforms, we will come back to the computational expense later, in suitable computational experiments.

3. It is convenient to set up the local strain-displacement matrix $\underline{\underline{B}}_e : Y \to \mathbb{R}^{6 \times 24}$ as follows. For the element under consideration, suppose the *scalar* gradient matrix $\underline{\underline{B}}_e^{\text{scalar}} : Y \to \mathbb{R}^{3 \times 8}$ is known. Then, the strain-displacement matrix admits the form

$$\underline{\underline{B}}_e = \underline{\underline{S}} \; \underline{\underline{B}}_e^{\text{scalar}} \otimes \underline{\underline{\text{Id}}}_{3 \times 3}, \tag{38}$$

    where $\otimes$ denotes the Kronecker product of matrices and the matrix $\underline{\underline{S}} \in \mathbb{R}^{6 \times 9}$ extracts from a full deformation gradient the symmetrized (strain) part in Voigt-Mandel notation. In this way, the level of difficulty is reduced to the scalar case.

4. For linear elastic material behavior, it is convenient to precompute and cache a few quantities that are used repeatedly. These encompass the element stiffness matrix $\underline{\underline{K}}_e^{\text{appr}}$ and the element-force vector induced by the macroscopic strain $\underline{f}_e$, see Equation (25), together with the matrix $\underline{\underline{A}}_e$ from Equation (31).

5. For the sake of exposition, the discussion was restricted to hexahedron finite elements with eight nodes. However, pretty much any other kind of finite element can be handled in a similar way, including the construction of an FFT-based preconditioner, as long as the regular-grid structure is retained. This is well-known in the literature,[58,61] and underlined by the examples considered in Section 1.2.

2.3 | Hourglass control

Let us fix an element Y_e and suppose that this element is governed by a linear elastic behavior (22) with stiffness $\underline{\underline{C}}_e$. When using one-point integration, the element stiffness-matrix $\underline{\underline{K}}_e^{\text{HEX8R}}$ does not have full rank. Rather, so-called hourglass modes[70-72] form the kernel of this matrix. The name *hourglass modes* derives from the characteristic deformation patterns that arise in two-dimensional analyses of finite-element discretizations with quadrilateral elements and one-point quadrature without hourglass stabilization.

In contrast, the fully integrated stiffness matrix $\underline{\underline{K}}_e^{\text{HEX8}}$ has full rank. However, its predicted response may be too stiff, and full integration comes with a high computational cost in the (physically) nonlinear case. For this purpose, we study hourglass stabilization of underintegrated elements[70-72] and present it in a general form. Before we start let us note that we may write

$$\underline{\underline{K}}_e^{\text{HEX8}} = \left\langle \underline{\underline{B}}_e^T \underline{\underline{C}}_e \underline{\underline{B}}_e \right\rangle_{Y_e}, \tag{39}$$

as the integration points for full integration permit to evaluate the integral exactly in the linear elastic case. Here, the operation $\langle \cdot \rangle_{Y_e}$ averages a quantity over the finite element Y_e. Similarly, we may express the stiffness matrix of the

underintegrated element

$$\underline{\underline{K}}_e^{\text{HEX8R}} = \left\langle \underline{\underline{\overline{B}}}_e^T \underline{\underline{C}}_e \underline{\underline{\overline{B}}}_e \right\rangle_{Y_e} \quad \text{with} \quad \underline{\underline{\overline{B}}}_e = \left\langle \underline{\underline{B}}_e \right\rangle_{Y_e}. \tag{40}$$

As the integrand is homogeneous in the element Y_e, one may also leave out the averaging and write

$$\underline{\underline{K}}_e^{\text{HEX8R}} = \underline{\underline{\overline{B}}}_e^T \underline{\underline{C}}_e \underline{\underline{\overline{B}}}_e. \tag{41}$$

To understand the source of hourglassing, it is instructive to inspect the quantity

$$\underline{\underline{K}}_e^{\text{stab}} = \left\langle \left(\underline{\underline{B}}_e - \underline{\underline{\overline{B}}}_e \right)^T \underline{\underline{C}}_e \left(\underline{\underline{B}}_e - \underline{\underline{\overline{B}}}_e \right) \right\rangle_{Y_e}, \tag{42}$$

in detail. Expanding the square yields

$$\begin{aligned}
\underline{\underline{K}}_e^{\text{stab}} &= \left\langle \left(\underline{\underline{B}}_e - \underline{\underline{\overline{B}}}_e \right)^T \underline{\underline{C}}_e \left(\underline{\underline{B}}_e - \underline{\underline{\overline{B}}}_e \right) \right\rangle_{Y_e} \\
&= \left\langle \underline{\underline{B}}_e^T \underline{\underline{C}}_e \left(\underline{\underline{B}}_e - \underline{\underline{\overline{B}}}_e \right) \right\rangle_{Y_e} - \left\langle \underline{\underline{\overline{B}}}_e^T \underline{\underline{C}}_e \left(\underline{\underline{B}}_e - \underline{\underline{\overline{B}}}_e \right) \right\rangle_{Y_e}.
\end{aligned} \tag{43}$$

For the second term, we observe

$$\left\langle \underline{\underline{\overline{B}}}_e^T \underline{\underline{C}}_e \left(\underline{\underline{B}}_e - \underline{\underline{\overline{B}}}_e \right) \right\rangle_{Y_e} = \underline{\underline{\overline{B}}}_e^T \underline{\underline{C}}_e \left\langle \underline{\underline{B}}_e - \underline{\underline{\overline{B}}}_e \right\rangle_{Y_e} = \underline{\underline{\overline{B}}}_e^T \underline{\underline{C}}_e \left(\left\langle \underline{\underline{B}}_e \right\rangle_{Y_e} - \underline{\underline{\overline{B}}}_e \right) \equiv 0, \tag{44}$$

as both $\underline{\underline{\overline{B}}}_e$ and $\underline{\underline{C}}_e$ are homogeneous and the identity (40) holds. Therefore, the Equation (43) becomes

$$\begin{aligned}
\underline{\underline{K}}_e^{\text{stab}} &= \left\langle \underline{\underline{B}}_e^T \underline{\underline{C}}_e \left(\underline{\underline{B}}_e - \underline{\underline{\overline{B}}}_e \right) \right\rangle_{Y_e} \\
&= \left\langle \underline{\underline{B}}_e^T \underline{\underline{C}}_e \underline{\underline{B}}_e \right\rangle_{Y_e} - \left\langle \underline{\underline{B}}_e^T \underline{\underline{C}}_e \underline{\underline{\overline{B}}}_e \right\rangle_{Y_e} \\
&= \left\langle \underline{\underline{B}}_e^T \underline{\underline{C}}_e \underline{\underline{B}}_e \right\rangle_{Y_e} - \left\langle \underline{\underline{B}}_e^T \right\rangle_{Y_e} \underline{\underline{C}}_e \underline{\underline{\overline{B}}}_e \\
&= \left\langle \underline{\underline{B}}_e^T \underline{\underline{C}}_e \underline{\underline{B}}_e \right\rangle_{Y_e} - \underline{\underline{\overline{B}}}_e^T \underline{\underline{C}}_e \underline{\underline{\overline{B}}}_e \\
&= \underline{\underline{K}}_e^{\text{HEX8}} - \underline{\underline{K}}_e^{\text{HEX8R}},
\end{aligned} \tag{45}$$

where we used the identity (40) again. The latter equation may also be recast in the form

$$\underline{\underline{K}}_e^{\text{HEX8}} = \underline{\underline{K}}_e^{\text{HEX8R}} + \underline{\underline{K}}_e^{\text{stab}}, \tag{46}$$

which conveniently expresses the difference between the element stiffness-matrices corresponding to full and reduced integration. In particular, the (symmetric and positive semidefinite) stabilization term $\underline{\underline{K}}_e^{\text{stab}}$ is actually responsible for suppressing the hourglass modes in the fully integrated element.

Inspired by the exact expression (46), one may consider the hourglass-stabilized element stiffness-matrix

$$\underline{\underline{K}}_e^{\text{HGC},\rho} = \underline{\underline{K}}_e^{\text{HEX8R}} + \rho \, \underline{\underline{K}}_e^{\text{stab}}, \tag{47}$$

involving an hourglass-stabilization parameter $\rho \in (0, 1]$. For any $\rho > 0$, the element stiffness-matrix (47) has full rank. The limit case $\rho \to 0$ recovers the underintegrated case (40), whereas full integration (39) may be reproduced, as well, by setting $\rho = 1$.

For nonlinear constitutive behavior, hourglass control is defined by the expression

$$\underline{g}_e^{\text{HGC},\rho}(\underline{E}, \underline{u}_e) = \underline{g}_e^{\text{HEX8R}}(\underline{E}, \underline{u}_e) + \rho \, \underline{\underline{K}}_e^{\text{stab}} \, \underline{u}_e, \tag{48}$$

where the stabilization term involves a non-degenerate elastic stiffness. For the work at hand, we use the initial elastic stiffness $\underline{\underline{C}}_e$. Different possibilities will be discussed below. The nodal forces computed with hourglass stabilization (48) are comprised of the nodal forces obtained from reduced integration, stabilized by a term which removes the hourglass modes. Clearly, reduced integration is recovered for $\rho \to 0$. However, in contrast to the linear elastic case, for $\rho = 1$, the expression (48) is different from the corresponding form for full integration.

Hourglass control (48) is convenient for nonlinear finite-element analyses as it uses only a single nonlinear evaluation of the constitutive law for each element, but is protected from spurious hourglassing as if more integration points were used. For nonlinear constitutive laws that are rather expensive to evaluate, using only a single nonlinear evaluation leads to a significant speed-up compared to full integration. Moreover, for a suitably chosen parameter ρ, the hourglass-controlled element (48) avoids the overly stiff response associated to a higher integration point count.

A few remarks complete this section.

1. Hourglass control with positive hourglass-stabilization parameter ρ avoids an ill-conditioning of the finite-element discretization for porous materials provided the microstructure satisfies a mechanical stability condition, see Schneider.[65] In particular, FFT-based solution methods may be applied in a reliable manner. Due to finite computer arithmetic, the hourglass-stabilization parameter ρ must not be too small. We will come back to this issue in Section 4.
2. In the classical literature on hourglassing, stabilizations of the form

$$\underline{\underline{K}}_e^{\text{stab,classical}} = \lambda_e \left\langle \left(\underline{\underline{B}}_e - \overline{\underline{\underline{B}}}_e \right)^T \left(\underline{\underline{B}}_e - \overline{\underline{\underline{B}}}_e \right) \right\rangle_{Y_e}, \tag{49}$$

were used with a prefactor λ_e which lies within the spectrum of the stiffness matrix $\underline{\underline{C}}_e$. To avoid possible problems of this classical form with strongly anisotropic materials, we retain the more general form (47).
3. Using the (algorithmic) tangent stiffness

$$\underline{\underline{C}}_e^{\text{alg}} = \frac{\partial \sigma_e}{\partial \underline{\varepsilon}} \left(\underline{E} + \overline{\underline{\underline{B}}}_e \, \underline{u}_e \right), \tag{50}$$

in the hourglass stabilization term instead of the elastic stiffness leads to an error

$$\underline{g}_e^{\text{HEX8}}(\underline{E}, \underline{u}_e) - \underline{g}_e^{\text{HEX8R}}(\underline{E}, \underline{u}_e) = O(h^2), \tag{51}$$

that is quadratic in the edge length h of the considered element Y_e. However, for (tri-)linear elements the L^2-error in the strains converges at most as $O(h)$, and it suffices to work with any symmetric and positive definite stiffness tensor $\underline{\underline{C}}_e^{\text{choice}}$ to arrive at the estimate

$$\underline{g}_e^{\text{HEX8}}(\underline{E}, \underline{u}_e) - \underline{g}_e^{\text{HEX8R}}(\underline{E}, \underline{u}_e) = O(h). \tag{52}$$

Still, it should be kept in mind that the combination $\rho \, \underline{\underline{C}}_e$ should be chosen to be close to the tangent stiffness for the cases of interest. In particular, choosing the term $\rho \, \underline{\underline{C}}_e$ too large may induce an undesired stiffening of the material response.
4. To construct the precomputable quantities convenient for linear elastic materials, the routines developed for full integration may actually be re-used for hourglass control. For this purposes, it is convenient to work with the modified strain-displacement matrix

$$\underline{\underline{B}}_e^{\rho} = \overline{\underline{\underline{B}}}_e + \sqrt{\rho} \left(\underline{\underline{B}}_e - \overline{\underline{\underline{B}}}_e \right), \tag{53}$$

instead of the original strain-displacement matrix $\underline{\underline{B}}_e$.

5. For nonlinear and inelastic material behavior (48), it may be difficult to find a value ρ for the hourglass-control parameter that is suitably small yet sufficiently stabilizing for the entire deformation. Therefore, it appears tempting to permit the parameter ρ to depend on the current deformation. Possibilities[70-72] include a dependence on the stress or on the material tangent. However, such a modification leads to further complications for an *implicit* treatment of the governing mechanical equations. Thus, we leave this topic for future studies.

3 | SOLUTION METHODS

3.1 | Equations to solve and the preconditioner

For fixed discretization, we are interested in solving the Equations (33)–(35). Then, for any parameter $s > 0$, to be specified later, we may rewrite the Equations (33)–(35) in the compact form

$$
\begin{aligned}
0 &= \underline{\underline{Q}}\ \underline{S}(\underline{E}, \underline{u}) - \overline{\underline{\sigma}} + \frac{1}{s} \left(\underline{\underline{P}}\ \underline{E} - \overline{\underline{\varepsilon}} \right), \\
0 &= \underline{r}(\underline{E}, \underline{u}),
\end{aligned}
\tag{54}
$$

where we used that the projectors $\underline{\underline{P}}$ and $\underline{\underline{Q}}$ are *complementary* projectors. As the second part of Equation (54) discretizes a partial differential equation, a preconditioning strategy may be used to improve the conditioning of the nonlinear system. In FFT-based computational micromechanics, such a preconditioning is achieved in terms of a suitable Green's operator $\underline{\underline{G}}$ associated to the discretization at hand.[25,26,93] More precisely, the Green's operator is defined via

$$\underline{u} = \underline{\underline{G}}\ \underline{f} \quad \text{precisely if} \quad \sum_{e=1}^{N} \underline{\underline{\Lambda}}_e^T \underline{\underline{K}}_e^0 \underline{\underline{\Lambda}}_e\ \underline{u} = \underline{f}, \tag{55}$$

for any right-hand side $\underline{f}$ with vanishing mean and where $\underline{\underline{K}}_e^0$ is the linear elastic element stiffness matrix with the "unit stiffness" $\underline{\underline{C}}_e \equiv \underline{\underline{\mathrm{Id}}}_{6 \times 6}$ in Voigt-Mandel notation, see Equation (25). Please note that the preconditioner $\underline{\underline{G}}$ is based on *the same* strain-displacement matrix as used for the FEM problem (54). With this preconditioner at hand, we may recast the equations to be solved (54) in a convenient fixed-point form

$$
\begin{bmatrix} \underline{E} \\ \underline{u} \end{bmatrix} = \begin{bmatrix} \underline{E} \\ \underline{u} \end{bmatrix} - s \begin{bmatrix} \underline{\underline{Q}}\ \underline{S}(\underline{E}, \underline{u}) - \overline{\underline{\sigma}} + 1/s \left(\underline{\underline{P}}\ \underline{E} - \overline{\underline{\varepsilon}} \right) \\ \underline{\underline{G}}\ \underline{r}(\underline{E}, \underline{u}) \end{bmatrix},
\tag{56}
$$

a discretized version of the Lippmann–Schwinger equation.[27-29] To compute the preconditioner in practice, we recall well-established strategies from the literature.[58-60] As the preconditioner (55) involves a homogeneous reference "stiffness" (and periodic boundary conditions), it may conveniently be computed via discrete Fourier transforms. Indeed, translation-invariant operators on a regular periodic lattice correspond to block-Fourier multipliers in Fourier space.[94] Thus, the operator on the right-hand side of Equation (55) is represented, for each Fourier frequency ξ in Fourier space, by a 3×3-matrix $\hat{\underline{\underline{K}}}(\xi)$ with complex coefficients. Due to the symmetry of the matrix $\underline{\underline{K}}_e^0$, the matrix $\hat{\underline{\underline{K}}}(\xi)$ is Hermitian. Thus, the action of the Green's operator $\underline{\underline{G}}$ may be computed by inverting the matrix $\hat{\underline{\underline{K}}}(\xi)$ at each frequency where it is non-singular. For fully integrated and hourglass-stabilized finite elements, only the frequency zero leads to such a singularity. For underintegrated elements, the hourglass modes need to be excluded as well[52] Equation (41).

As noticed by Fritzen-Leuschner[59] Equations (47) and (48), the 3×3 Hermitian matrix $\hat{\underline{\underline{K}}}(\xi)$ admits the representation

$$\underline{\underline{\hat{K}}}(\xi) = \underline{\underline{Z}}(\xi)^H \, \underline{\underline{K}}_e^0 \, \underline{\underline{Z}}(\xi) \tag{57}$$

with matrices $\underline{\underline{Z}}(\xi) \in \mathbb{R}^{24 \times 3}$ that encode the relative positions of the nodes within the element. Here, H stands for Hermitian conjugation, that is, complex conjugation and transposition. As the matrices $\underline{\underline{Z}}$ do not depend on whether we use full, reduced or hourglass-stabilized integration, the formula (57) turns out to be extremely convenient for the implementation.

The action of the Green's operator, $\underline{f} \mapsto \underline{\underline{G}}\underline{f}$, may be computed in-place in terms of three forward and three backward FFTs, together with solving a single 3×3-linear system of equations with complex coefficients for each Fourier frequency. In particular, no extra storage is required for this strategy.[52,58] As noticed by Leuschner–Fritzen,[59(3.2)] the actual bottleneck in terms of computing speed is the assembly of the matrix $\underline{\underline{\hat{K}}}(\xi)$ in Equation (57). To speed up computations, it is therefore advantageous to compute and store the (pseudo-)inverses $\underline{\underline{\hat{K}}}(\xi)^\dagger$ in a preprocessing step. As these inverses are Hermitian, it suffices to store nine scalar fields per frequency, three for the (real) diagonal and six for the real and imaginary part of the upper triangular part of the matrix. Caching the preconditioner was also used by Ladecký et al..[60]

3.2 | Gradient-descent methods and the residual

For a sequence s^k of positive step sizes and an initial configuration $(\underline{E}^0, \underline{u}^0)$, the fixed-point form (56) of the equations to be solved gives rise to the computational scheme

$$\begin{bmatrix} \underline{E}^{k+1} \\ \underline{u}^{k+1} \end{bmatrix} = \begin{bmatrix} \underline{E}^k \\ \underline{u}^k \end{bmatrix} - s^k \begin{bmatrix} \underline{\underline{Q}} \, \underline{S}(\underline{E}^k, \underline{u}^k) - \overline{\underline{\sigma}} + 1/s^k \left(\underline{\underline{P}} \, \underline{E}^k - \overline{\underline{\varepsilon}} \right) \\ \underline{\underline{G}} \, \underline{r}(\underline{E}^k, \underline{u}^k) \end{bmatrix}. \tag{58}$$

If, for some index k, the constraint

$$\underline{\underline{P}} \, \underline{E}^k = \overline{\underline{\varepsilon}}, \tag{59}$$

is satisfied, all subsequent iterates will satisfy this constraint, as well. In particular, if the initial strain $\underline{E}^0$ satisfies the constraint (59), for instance by setting $\underline{E}^0 = \overline{\underline{\varepsilon}}$, the constraint (59) will be satisfied for all subsequent iterations. If the constraint (59) is not satisfied, the iterative scheme (58) will ensure that the constraint (59) holds after a single iteration, and, subsequently, will hold for all succeeding iterates. In particular, it is reasonable and convenient to study the simplified iterative scheme

$$\begin{bmatrix} \underline{E}^{k+1} \\ \underline{u}^{k+1} \end{bmatrix} = \begin{bmatrix} \underline{E}^k \\ \underline{u}^k \end{bmatrix} - s^k \begin{bmatrix} \underline{\underline{Q}} \, \underline{S}(\underline{E}^k, \underline{u}^k) - \overline{\underline{\sigma}} \\ \underline{\underline{G}} \, \underline{r}(\underline{E}^k, \underline{u}^k) \end{bmatrix}. \tag{60}$$

For constant step size $s \equiv s^k$, this iterative scheme corresponds to the basic scheme introduced by Moulinec and Suquet[25,26] in a displacement-based implementation[59,80,93] with mixed boundary conditions.[79,85]

To close such an approach algorithmically, two topics need to be addressed, namely an appropriate termination criterion and a suggestion on how to select the step size s. Let us start with a suitable convergence criterion. Consistent with developments for strain-based implementations of FFT-based computational micromechanics,[49,65,78] we use the criterion

$$\mathrm{res}_k \overset{!}{\le} \mathrm{tol} \, \|\underline{S}(\underline{E}^k, \underline{u}^k)\| \tag{61}$$

with

$$\mathrm{res}_k = \sqrt{\|\underline{\underline{Q}} \, \underline{S}(\underline{E}^k, \underline{u}^k) - \overline{\underline{\sigma}}\|^2 + \underline{r}(\underline{E}^k, \underline{u}^k)^T \underline{\underline{G}} \, \underline{r}(\underline{E}^k, \underline{u}^k)}, \tag{62}$$

for a desired tolerance `tol` > 0. In particular, the strain- and the displacement-based implementations use identical convergence criteria and give rise to identical residuals, up to effects of finite numerical precision, as long as identical discretization schemes are used.

To select the step size s, let us record that it should have the dimension of a compliance. It has been analyzed[30,31,36] that, in case of symmetric and uniformly positive definite algorithmic tangents, the fastest convergence rate is reached by choosing $1/s$ to be the average of the smallest and the largest eigenvalues of the algorithmic material tangent, evaluated over all microscopic points in the cell and all applied strain tensors. For linear elastic materials, only the distinct elasticity tensors need to be analyzed in terms of their eigenvalues. In the (physically) nonlinear case, more effort is required.

There is a minimal modification

ALGORITHM 1. Displacement-based Barzilai-Borwein basic scheme (`maxit, tol`)

1: Determine initial guess $(\underline{E}, \underline{u})$ and initial step size s

2: $\underline{\Sigma} \leftarrow \bar{\sigma}$ ▷ Ensures $q = 0$ for $k = 0$

3: $\underline{f} \leftarrow 0$ ▷ Ensures $q = 0$ for $k = 0$

4: $\mathtt{res} \leftarrow \mathtt{tol}\, \|\underline{\Sigma}\| + 1$ ▷ Dummy value to enter while loop

5: $k \leftarrow 0$

6: **while** $k < \mathtt{maxit}$ **and** $\mathtt{res} > \mathtt{tol}\, \|\underline{\Sigma}\|$ **do**

7:     $k \leftarrow k + 1$

8:     $\mathtt{res_{old}} \leftarrow \mathtt{res}$

9:     $\begin{bmatrix} \underline{\Sigma} \\ \underline{f} \\ \underline{q} \end{bmatrix} \leftarrow \begin{bmatrix} \underline{S}(\underline{E}, \underline{u}) \\ \underline{r}(\underline{E}, \underline{u}) \\ \underline{f}^T \underline{r}(\underline{E}, \underline{u}) + (\underline{Q}\, \underline{\Sigma} - \bar{\sigma})^T (\underline{Q}\, \underline{S}(\underline{E}, \underline{u}) - \bar{\sigma}) \end{bmatrix}$ ▷ Real space computation

10:     $\begin{bmatrix} \underline{f} \\ \mathtt{res} \end{bmatrix} \leftarrow \begin{bmatrix} \underline{G} \underline{f} \\ \underline{f}^T \underline{G} \underline{f} \end{bmatrix}$ ▷ Fourier space computation

11:     $\mathtt{res} \leftarrow \sqrt{\mathtt{res} + \|\underline{Q}\, \underline{\Sigma} - \bar{\sigma}\|^2}$

12:     $s \leftarrow \mathtt{res_{old}^2}\, s / \left(\mathtt{res_{old}^2} - q \right)$ ▷ Use different update to get basic scheme

13:     $\underline{E} \leftarrow \bar{\varepsilon} + \underline{Q}\, \underline{E} - s \left(\underline{Q}\, \underline{\Sigma} - \bar{\sigma} \right)$

14:     $\underline{u} \leftarrow \underline{u} - s\underline{f}$

15: **end while**

---

of the basic scheme which was shown to speed up the computations significantly and, at the same time, dispenses with the need of costly eigendecompositions of algorithmic tangents. For the Barzilai-Borwein method,[44] the step size is inductively updated by the formula[95]

$$s^k = \frac{\mathtt{res}_{k-1}^2}{\mathtt{res}_{k-1}^2 - \left(\underline{Q}\, \underline{S}(\underline{E}^{k-1}, \underline{u}^{k-1}) - \bar{\sigma} \right)^T \left(\underline{Q}\, \underline{S}(\underline{E}^k, \underline{u}^k) - \bar{\sigma} \right) - \underline{r}(\underline{E}^k, \underline{u}^k)^T \underline{G}\, \underline{r}(\underline{E}^{k-1}, \underline{u}^{k-1})}\, s^{k-1}. \tag{63}$$

For the initial step size s^0, the basic step size determined for the initial elastic behavior is recommended. The Barzilai-Borwein update formula (63) only requires to compute an additional L^2-inner product between two vector fields compared to an implementation of the basic scheme. The residual (62) is computed anyway, and the inner product for the error in the macroscopic stress can be computed at negligible computational cost.

The basic scheme (58) is conveniently implemented on *two* displacement fields. Although it would be possible to implement the scheme on a single displacement field, the convergence criterion (61) appears to require two such fields. Not counting internal variables and the last converged time step, the equivalent of five displacement fields is necessary to store when working with the basic scheme and the precomputation strategy discussed in Section 3.1. If underintegrated elements are used, the action preconditioner can be efficiently computed on the fly and no precomputations will be

necessary. Thus, in this case, only two displacement fields suffice. These considerations also apply to the Barzilai-Borwein scheme.[44] Thus, no additional memory is required compared to the basic scheme.

For the readers' convenience, an algorithmic summary of the (Barzilai-Borwein) basic scheme is given in Algorithm 1. The bracket notation indicates where some quantities are re-used for each voxel or Fourier frequency to reduce both computational expense and memory footprint. The residual may conveniently be computed in Fourier space. When using real-space transforms, some care is advised. Indeed, Parseval's theorem does hold, but the zero and the Nyqvist frequencies are counted with a factor unity, whereas the other frequencies need to be furnished with a prefactor two.

3.3 | Conjugate-gradient methods

ALGORITHM 2. Displacement-based Fletcher–Reeves nonlinear CG (`maxit`, `tol`)

1: Determine initial guess $(\underline{E}, \underline{u})$ and initial step size s

2: $\underline{\Sigma} \leftarrow \bar{\sigma}$

3: $[\underline{f}, \underline{d}, \underline{\Sigma}_d, k] \leftarrow [0, 0, 0, 0]$

4: $\mathtt{res} \leftarrow \mathtt{tol}\,\|\underline{\Sigma}\| + 1$ ▷ Dummy value to enter while loop

5: **while** $k < \mathtt{maxit}$ **and** $\mathtt{res} > \mathtt{tol}\,\|\underline{\Sigma}\|$ **do**

6:      $k \leftarrow k + 1$

7:      $\mathtt{res}_{\mathrm{old}} \leftarrow \mathtt{res}$

8:      $\begin{bmatrix} \underline{\Sigma} \\ \underline{f} \end{bmatrix} \leftarrow \begin{bmatrix} S(\underline{E}, \underline{u}) \\ r(\underline{E}, \underline{u}) \end{bmatrix}$ ▷ Real space computation

9:      $\begin{bmatrix} \underline{f} \\ \mathtt{res} \end{bmatrix} \leftarrow \begin{bmatrix} \underline{\underline{G}}\underline{f} \\ \underline{f}^T \underline{\underline{G}} \underline{f} \end{bmatrix}$ ▷ Fourier space computation

10:      $\mathtt{res} \leftarrow \sqrt{\mathtt{res} + \|\underline{\underline{Q}}\,\underline{\Sigma} - \bar{\sigma}\|^2}$

11:      $\beta \leftarrow \mathtt{res}^2 / \mathtt{res}_{\mathrm{old}}^2$ ▷ One may safeguard to $\beta \in [0, 1]$

12:      $\underline{d} \leftarrow -\underline{f} + \beta\,\underline{d}$

13:      $\underline{\Sigma}_d \leftarrow -\left(\underline{\underline{P}}\,\underline{E} - \bar{\varepsilon}\right)/s - \underline{\underline{Q}}\,\underline{\Sigma} - \bar{\sigma} + \beta\,\underline{\Sigma}_d$

14:      $\underline{u} \leftarrow \underline{u} + s\,\underline{d}$

15:      $\underline{E} \leftarrow \underline{E} + s\,\underline{\Sigma}_d$

16: **end while**

For the linear conjugate gradient method, it is convenient to rewrite the problem (54) in the following form

$$\underline{\underline{Q}}\,S(\underline{E}, \underline{u}) + \frac{1}{s}\,\underline{\underline{P}}\,\underline{E} = \bar{\sigma} + \frac{1}{s}\,\bar{\varepsilon},$$
$$r(\underline{E}, \underline{u}) = 0, \tag{64}$$

which should be regarded as a block-linear system

$$\underline{\underline{A}} \begin{bmatrix} \underline{E} \\ \underline{u} \end{bmatrix} = \begin{bmatrix} \bar{\sigma} + \bar{\varepsilon}/s \\ 0 \end{bmatrix}, \tag{65}$$

in terms of a suitable block-diagonal matrix $\underline{\underline{A}}$. With the block preconditioner

$$\begin{bmatrix} \underline{\underline{\mathrm{Id}}}_{6\times6} & \\ & \underline{\underline{G}} \end{bmatrix}, \tag{66}$$

a standard implementation of the preconditioned (linear) conjugate gradient method is advised.[96(11.5.2)] Then, the classical CG residual coincides with the residual (62), and the convergence criterion (61) may be used to remain consistent with the previous section.

Please note that in the form (64), the macroscopic loading conditions appear on the right-hand side in a clean form, and no additional terms involving the stiffness (or an eigenstrain) do appear on the right hand side. In particular, the force residual to be reached is zero. This formulation (and the resulting CG) is actually closer in spirit to the original work of Zeman et al.[33] than subsequent works.[41,59,60]

The formulation (64) is also advantageous for the case of pure strain loading, that is, $\underline{\underline{P}} = \mathrm{Id}_{\underline{\underline{6 \times 6}}}$ and $\overline{\sigma} = 0$, as it does not necessitate any splitting of the term $\underline{r}$. In particular, the presented CG method can be handled within a single implementation of the basic step, increasing the conciseness and maintainability of the developed code. The preconditioned linear conjugate gradient method is applicable provided the constitutive behavior is linear thermoelastic, and the implementation is standard.

For nonlinear constitutive behavior, nonlinear extensions of the conjugate gradient method[97,98] were shown to be effective in the context of FFT-based computational micromechanics.[38] The classical Fletcher–Reeves CG[97] turned out to be both simple and effective. The algorithm, see Algorithm 2, may be implemented with little extra effort on top of a running basic scheme. Let us remark that we use the *elastic* step size recommended for the basic scheme for the CG method. Moreover, Algorithm 2 operates on three displacement fields.

4 | COMPUTATIONAL INVESTIGATIONS

4.1 | Setup

The algorithms reported in this article were realized in Python (with Cython extensions) and parallelized using OpenMP. The subsequent sections also contain a comparison to an existing *strain-based* computational micromechanics code. Unless mentioned otherwise, the examples were solved up to a tolerance of $\mathtt{tol} = 10^{-5}$, measured in terms of the convergence criterion (61). The runtimes were recorded on a desktop computer with six 3.70 GHz cores and 32 GB RAM. Table 1 summarizes the material parameters used in this article.

4.2 | Porous materials

We consider microstructures involving a gaseous phase, that is, porous materials, as those represent challenges for some FFT-based solvers. More to the point, the source for convergence issues is not to be found in the solvers themselves. Rather, the underlying discretization scheme is responsible for possible non-convergence.[55,65,105]

For a start, we consider a specific cellular microstructure that comes with remarkable mechanical properties—the octet truss[104] whose micromechanical behavior has been studied by Lucarini et al.[106] with FFT-based methods.

The investigated microstructure is shown in Figure 3 and has a volume fraction of 10%. Thus, 90% of the geometry consists of the porous phase. Before taking a closer look at the convergence behavior, we study the local solution fields. More precisely, we consider a uniaxial extension to 5% in x-direction for linear elastic materials, where we furnish the solid space with the material properties of aluminum, see Table 1. In Figure 3C, the xx-component of the local stress field on the surface of the solid material is shown for the (non-consistent) staggered grid discretization.[55] We observe that the field lacks symmetry w.r.t. a reflection at the yz-plane going through the center of the cube. This is unexpected, as both the geometry and the loading scenario are actually invariant under such a reflection. Hence, the same symmetry is expected from the solution field. This lack of symmetry is actually caused by using the non-consistent version of the staggered grid, where the shear components of the stress field—which are actually evaluated on the voxel edges—are shifted to the voxel centers. Alternative versions of the staggered grid, as reported in Schneider et al.[55] and Grimm-Strele and Kabel,[66] may cure this problem, but come with an increased computational cost. This half-voxel shift at inter- and surfaces actually constitutes a limitation of the staggered grid discretization.

The HEX8R-discretization[52] does not suffer from this lack of symmetry. Indeed, Figure 3A shows the perfectly symmetric solution fields. This salient property is inherited by the hourglass-stabilized finite element discretization presented
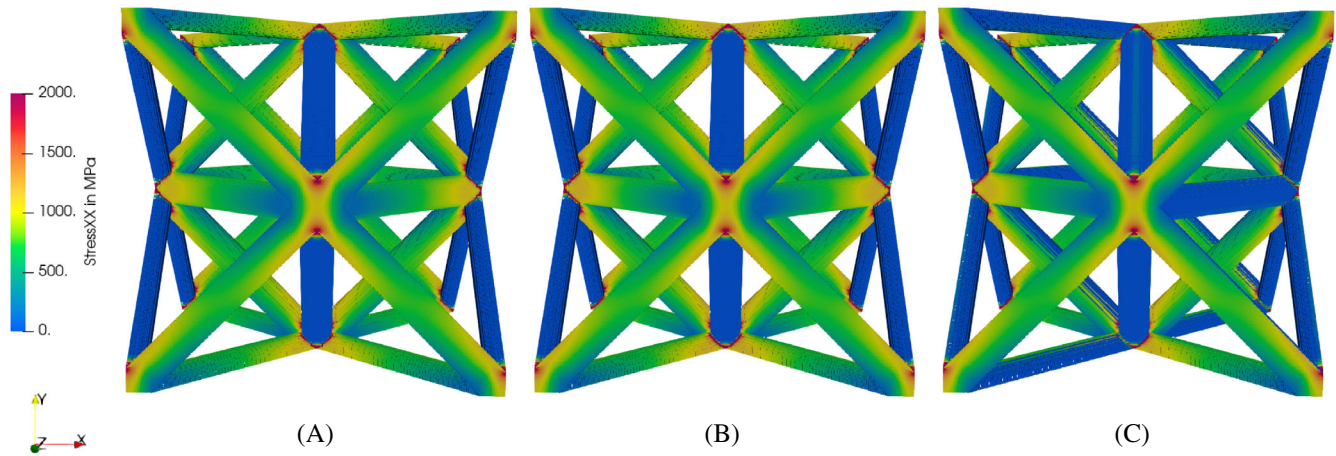
FIGURE 3 *xx*-component of the local stress field σ_{xx}, computed on an aluminum octet truss structure[104] under uniaxial extension to 5% and different discretization schemes. (A) Reduced integration;[52] (B) 1% hourglass control; (C) staggered grid[55]
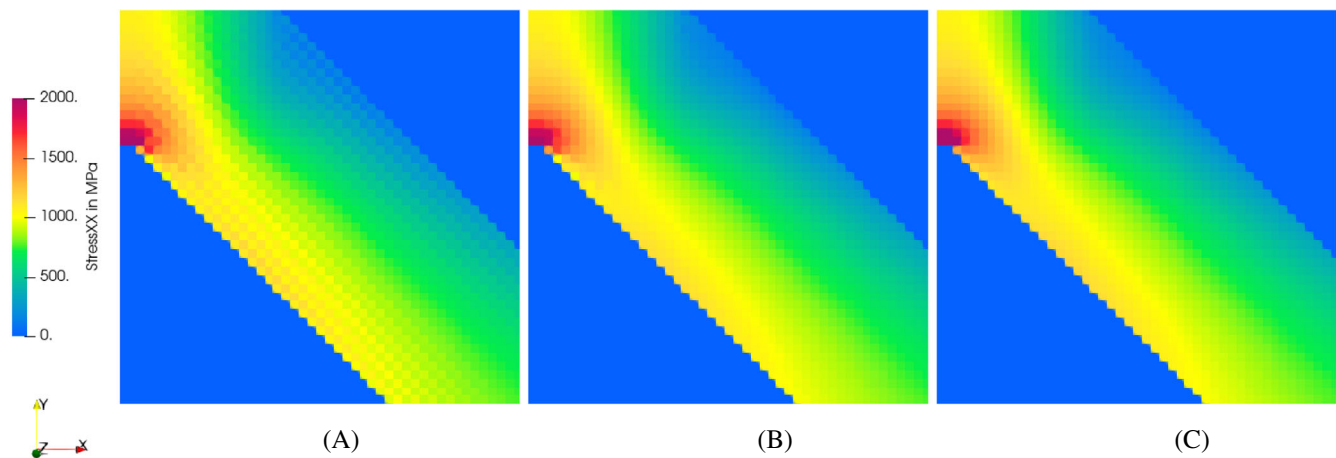

FIGURE 4 Magnified local solution fields in a central section through one of the arms of the octet truss shown in Figure 3. (A) Reduced integration;[52] (B) 1% hourglass control; (C) staggered grid[55]

in the work at hand. As a representative for different stabilization parameters ρ, Figure 3B shows the solution field with 1% hourglass stabilization.

However, the underintegrated hexahedron element also comes with certain limitations. To gain more insight, we study the local solution fields more closely in a two-dimensional section. In Figure 4A, which shows one of the corners of the octet truss in a transverse intersection, we observe that the trilinear finite element without hourglass stabilization shows its characteristic checkerboard pattern that was previously observed by several authors.[52,59,60] In contrast, see Figure 4C, the staggered grid discretization does not show such checkerboard patterns. Rather, the solution field appears extremely smooth. Taking a look at the solution fields with 1% hourglass stabilization, see Figure 4B, we observe that the solution fields are again smooth.

Thus, in terms of local solution quality, the hourglass-stabilized `HEX8` element unites the advantages of both the hexahedron element with reduced integration and the staggered grid discretization.

Complementing aesthetic characteristic, we study the convergence properties associated to various discretizations for the octet-truss examples. More precisely, we consider the linear conjugate-gradient method for the load case of uniaxial extension in *x*-direction, solved up to a residual of 10^{-5}.

Figure 5 compares the classical Moulinec–Suquet discretization based on trigonometric polynomials, the staggered grid, and a number of finite-element discretizations. We observe that the Moulinec–Suquet discretization fails to converge.
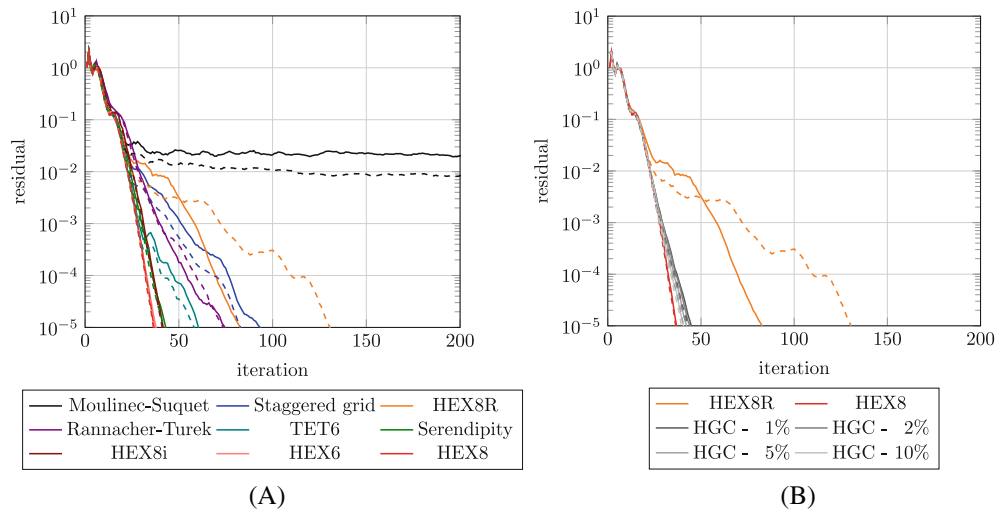
FIGURE 5 Residual versus iteration for the octet truss shown in Figure 3 and different discretization methods for the resolutions 128^3 (solid line) and 256^3 (dashed line). (A) Comparison of classical FFT-affine discretizations and various finite elements; (B) in-depth study on the influence of the hourglass stabilization parameter ρ

Rather, a residual below 10^{-2} is not reached for either resolution. All other considered finite-different and finite-element discretizations do converge. The hexahedron elements with six and eight integration points converge fastest, and show little influence of the resolution. The serendipity element and the incompatible-modes element follow closely. Both the Rannacher–Turek element and the hexahedron element with six sub-tetrahedra require both more iterations, and are trailed by the staggered grid discretization. In contrast to the other discretizations, the HEX8R shows a distinct dependence on the resolution. For the grid with eight times as many voxels, the iteration count increases by about 50%, indicating a possible instability for porous materials. We will come back to this observation later in this section.

In Figure 5B, the effect of various levels of hourglass control is studied on the performance of the conjugate gradient method. We observe that a higher degree of hourglass stabilization does also decrease the iteration count required to reach the desired tolerance. Yet, the difference between the different hourglass parameters is rather small. Indeed, even only 1% of hourglass stabilization leads to an iteration count well below 50, which is almost half of the number of iterations required for the underintegrated version for 128^3 voxels and about a third of the for 256^3 voxels. Thus, for the octet-truss example, hourglass control is rather effective in stabilizing the iteration count.

To get a broader perspective, we study a second material. The microstructure at hand, see Figure 6A, contains about 60% quartz sand grains, held together by less than 1% of anorganic binder. The microstructure was generated by the method described in Schneider et al.[107] by a mechanical-contraction type algorithm. This example is rather challenging for FFT-based solvers as it involves three materials: the sand grains, the binder *and* about 40% of pore space, all arranged in an intricate and complex pattern. We endow the microstructure with the linear elastic material parameters listed in Table 1 and subject the material to 5% uniaxial extension in *x*-direction. The residual versus iteration is shown in Figure 6B for the linear conjugate gradient method. We observe that the HEX8R discretization fails to reduce the residual below 3×10^{-4}. Thus, we recover a phenomenon that is well-known in the community. Whether or not the FFT-based solvers converge for the HEX8R discretization depends both on the geometry and the physical model under consideration. Indeed, for the octet truss that we studied earlier, there were no problems with convergence per se. Yet, the iteration count turned out to depend on the resolution. For the sand-grain structure at hand, matters are different. The reason is to be found in the hourglass modes. Indeed, in the absence of porous phases, that is, for finite material contrast, although the HEX8R discretization features *local* hourglass modes, all *global* hourglass modes can be suppressed by the FFT-based preconditioner. When pores are present, the local hourglass modes may induce global hourglass modes that are not accounted for by the preconditioner. Indeed, the possible hourglass modes depend on the given topology of the solid space. In particular, the numerical conditioning of the ensuing linear systems depends on the complexity of the solid/pore space. Moreover, the conditioning *does also* depend on the physical model under consideration. Indeed, it is well-known that for thermal conductivity, there is only one local hourglass mode, whereas there are four hourglass modes in three-dimensional
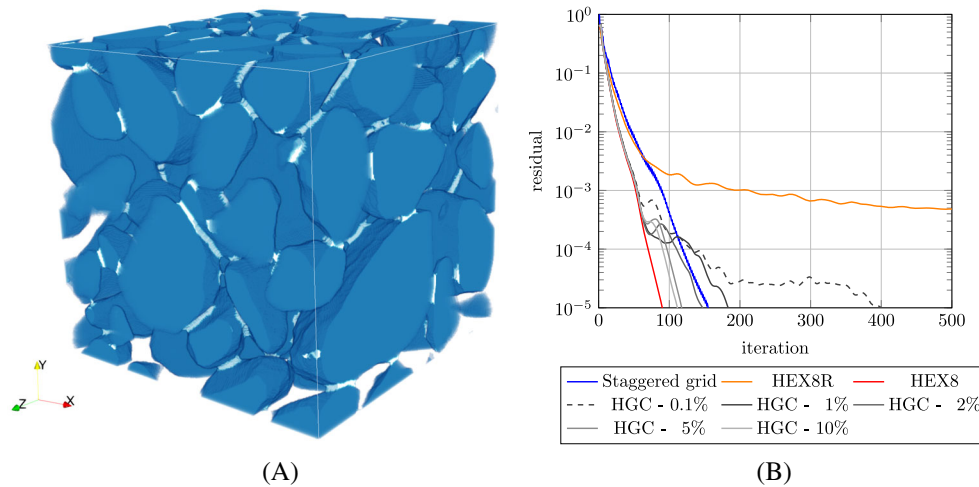
FIGURE 6 Microstructure and performance of different discretizations for a microstructure of bound sand. (A) Microstructure;[107,110] (B) residual versus iteration for 5% uniaxial extension in x-direction

linear elasticity. Thus, in the latter case, there are more possibilities to create global hourglass modes, and, subsequently, ill-conditioned linear and nonlinear systems.

When looking at the staggered grid discretization, a steady convergence behavior is observed in Figure 6B. Due to the staggered placement of the unknowns, hourglassing is completely avoided.

Similarly, in Figure 6B we observe that a sufficiently strong hourglass stabilization leads to a convergent linear CG. However, if the hourglass parameter is too small, $\rho = 0.1\%$ in the example, the iteration count may be rather high, that is, about 400. Still, convergence is ensured - in contrast to the HEX8R discretization. Yet, there is almost a factor of four in iteration count compared to 10% hourglass control (112 iterations). On the other hand, little is gained in terms of iteration count when considering full integration (91 iterations).

To conclude this investigation on porous microstructures, we take a look at the local solution fields on a cut-out of a section through the sand-binder structure in Figure 7B. We only compare the stabilized HEX8R discretization to the staggered grid, as the HEX8R did not converge. We observe in Figure 7A rather smooth solution fields devoid of oscillations or checkerboards. These properties are shared with the staggered grid discretization, see Figure 7B. Only in regions of high stresses (and sufficient magnification), differences in the solution fields between the two considered discretizations become visible.

To sum up, adding hourglass control to the HEX8R discretization does indeed stabilize the convergence behavior of this type of discretization for porous materials. Moreover, the local solution fields share the advantageous properties of both the HEX8R discretization and the discretization on a staggered grid.

4.3 | Computational effort

In this section, we wish to study the computational effort that comes with hourglass stabilization more closely. For this purpose, we consider a short-fiber reinforced composite. The microstructure, shown in Figure 8A, features 15 volume-% of short fibers in a cubic volume element with edge length of 288 μm and at an isotropic fiber-orientation state. The fibers have a length of 200 μm and a diameter of 10 μm. The geometry was generated by the SAM algorithm,[108] contains 229 fibers and is resolved by 192^3 voxels, that is, a mesh size $h = 1.5$ μm. We endow the fibers with the material properties of the polymer matrix and the E-glass fibers described in Table 1.

In a first step, we study the performance under uniaxial extension in x-direction and linear elastic material behavior. Comparing the different hexahedron discretizations, we observe that the computed average stress strongly differs between full and reduced integration, see Table 2. Indeed, full integration leads to a stress level almost 11% higher than for reduced integration. As expected, there is little difference between using six and eight integration points. With varying hourglass-stabilization parameter ρ, we may smoothly transit from the underintegrated to the fully integrated element.
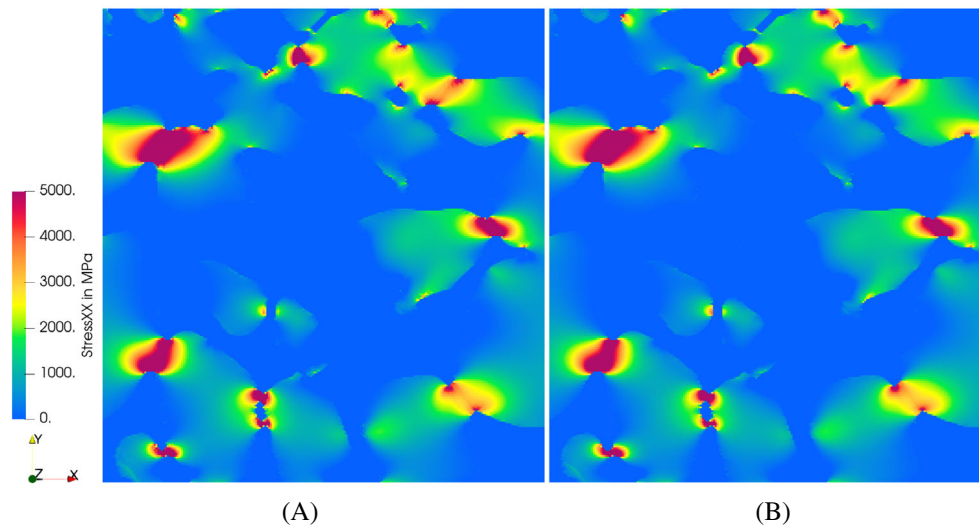
FIGURE 7 Local stress field in *xx*-direction under 5% uniaxial extension in *x*-direction for a slice of the sand-grain structure, see Figure 6A, and different discretizations. (A) 1% hourglass stabilization; (B) staggered grid[55]
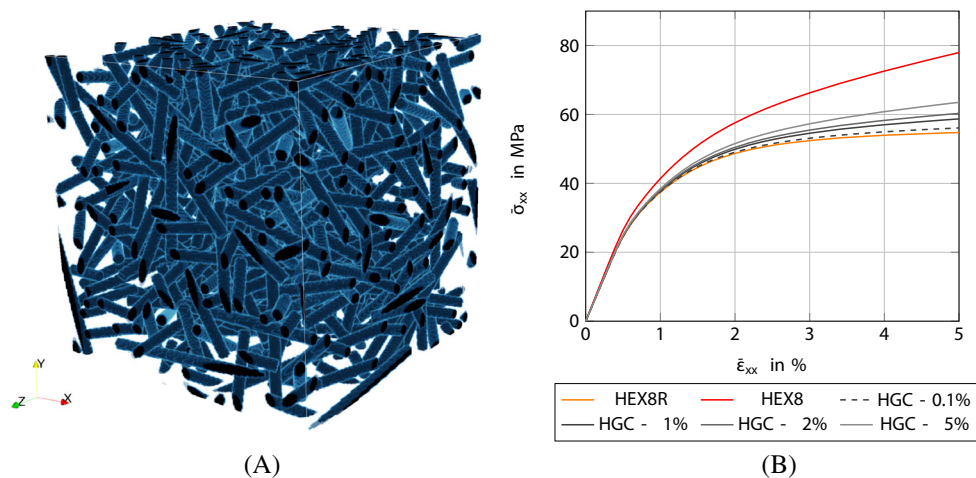

FIGURE 8 Microstructure and computed mechanical behavior under uniaxial extension in *x*-direction for a short-fiber reinforced composite. (A) Microstructure;[109] (B) stress–strain diagram

Taking a look at the iteration count required for the linear CG, we observe that all discretizations require exactly 36 iterations. Moreover, the runtimes for all the displacement-based implementations are roughly the same. Even splitting into individual contributions as the preconditioner step and the material evaluation, we observe little differences. This is remarkable, as the preconditioner for HEX8R is computed on the fly, whereas forming the preconditioner for the other considered discretizations is much more expensive. The similarity in runtime is a result of the ingenious precomputation and caching strategy of Leuschner and Fritzen.[59]

To put the displacement-based implementation into context, a similar strain-based implementation[44] of the HEX8R is included in Table 2, as well. As the residuals are matching, both the iteration count and the resulting effective stresses coincide. It is more interesting to look at the runtimes. The preconditioning takes 2.5 times longer for the strain-based implementation. This is to be expected, as six FFTs need to be performed instead of three. Moreover, computing the strain-based Green's operator is a little more involved than computing only the displacement version. Indeed, an additional divergence and a symmetrized gradient operation are needed, as well. In contrast, evaluating the constitutive law is much faster for the strain-based implementation than for the displacement-based implementation. This comes as no surprise, as applying the elastic stiffness matrix involves a 6×6 matrix, whereas applying the nodal-force matrix

TABLE 2 Comparison of different discretizations for 5% uniaxial extension in *x*-direction of the short-fiber structure, see Figure 8A, the linear conjugate gradient method and linear elastic material behavior

Discretization	$\bar{\sigma}_{xx}$ in MPa	Iterations	Runtime in s		
			Preconditioning	Material	Total
HEX8	274.19	36	8.3	31.1	54.2
HEX6	273.51	36	8.6	31.2	55.0
HEX8R	255.34	36	8.6	30.5	54.0
HGC—0.1%	255.72	36	8.3	31.4	54.3
HGC—1%	256.77	36	8.3	31.0	54.2
HGC—2%	257.49	36	8.4	31.2	54.5
HGC—5%	259.01	36	8.2	31.0	54.2
HGC—10%	260.78	36	8.3	29.3	52.9
HEX8R (strain)	255.34	36	19.3	2.3	37.3

TABLE 3 Comparison of different discretizations for 5% uniaxial extension in *x*-direction of the short-fiber structure, see Figure 8A, with von-Mises elastoplastic matrix, a single load step and the nonlinear conjugate gradient method

Discretization	$\bar{\sigma}_{xx}$ in MPa	Iterations	Runtime in s		
			Preconditioning	Material	Total
HEX8	76.93	584	143.8	2410.5	2652.1
HEX6	75.73	592	145.9	1901.0	2141.5
HEX8R	54.12	863	203.7	695.9	1017.6
HGC—0.1%	55.46	786	194.4	995.5	1299.8
HGC—1%	58.01	765	189.3	964.8	1262.1
HGC—2%	59.58	1247	311.2	1575.9	2056.4
HGC—5%	62.74	534	132.6	672.9	883.2
HGC—10%	66.34	456	112.7	575.8	756.2
HEX8R (strain)	54.12	863	436.4	200.0	902.3

requires dealing with a 24 × 24-matrix. Thus, there is a factor 16 in size between these two matrices. Indeed, we observe a factor of roughly 15 between the material-evaluation runtimes for the displacement-based and the strain-based implementation. All in all, the strain-based implementation requires only two thirds of the runtime of the displacement-based implementation.

These considerations should be taken with a grain of salt, as we are comparing apples and oranges. The author decided to include these results anyway, as the reader may wonder about precisely such runtimes. Actually, both implementations were done by the same individual and with the same software framework (Python + Cython extensions + OpenMP) and the same "tricks". Clearly, there might be optimizations that are either not accessible by the chosen programming language or are specific to matrix-free finite-element evaluation that may be explored in subsequent works.

After these linear considerations,

we turn our attention to physically nonlinear problems. More precisely, we consider a von-Mises elastoplastic matrix material with linear-exponential hardening, see Table 1 for the material parameters, and apply 5% uniaxial strain in a single load step. In this way, we avoid some of the additional complications inferred by multiple load steps. The results are recorded in Table 3. We observe a strong discrepancy between the stresses for full and reduced integration. The axial stresses for the former are 42% higher than for the latter. Yet, this large relative difference is actually a consequence of the differences in the initial elastic response. Indeed, the absolute difference between the elastic axial stresses is 18.9 MPa, and the differences in the inelastic case are 22.8%.

TABLE 4 Comparison of different discretizations for 5% uniaxial extension in *x*-direction of the short-fiber structure, see Figure 8A, with von-Mises elastoplastic matrix with 50 equidistant load steps and the nonlinear conjugate gradient method

Discretization	Iterations	Runtime in min		
		Preconditioning	Material	Total
HEX8	3231	13.4	200.9	233.9
HEX6	3251	13.5	158.2	188.0
HEX8R	3928	15.7	48.3	73.0
HGC—0.1%	3822	15.9	76.6	103.0
HGC—1%	3652	15.2	72.9	98.3
HGC—2%	3572	14.8	71.8	96.7
HGC—5%	3397	14.1	67.7	91.6
HGC—10%	3197	13.3	64.1	86.8
HEX8R (strain)	3849	33.0	13.4	66.4

As for the linear elastic case, the hourglass parameter may be used to increase the computed stresses from the levels of reduced integration to those of full integration. Taking a look at the iteration counts, we observe that the reduced integration takes about 48% more iterations to converge than using full integration. Hourglass control leads to iteration counts which roughly conform with this range, but there are outliers. These fluctuations in the iteration count compared to the linear case are a consequence of the nonlinear solver used. Indeed, the adaptive parameter selection of the nonlinear CG[38] is much more sensitive than linear CG.

It is interesting to study the runtimes for this examples. Indeed, for inelastic material behavior, the trick with the precomputed force-displacement matrix (25) does not apply, and the constitutive law needs to be evaluated on each integration point. In particular, the HEX8 element needs to evaluate eight integration points instead of six for HEX6 within the matrix material. Thus, the material evaluation for the latter is about 25% faster than for eight integration points. For the total runtime, this leads to a speed up of 24%.

For reduced integration, only a single integration point needs to be processed for inelastic materials. Thus, although roughly 50% more iterations are required, this discretization takes less than half of the runtime to be solved compared to eight- and six-point integration. It is also interesting to compare HEX8R to a minimum level of hourglass control. For $\rho = 0.1\%$, there is little difference in the effective stresses and the iteration counts, yet there is a 43% increase in runtime for the material evaluation. Indeed, the von-Mises elastoplastic law is rather inexpensive, and adding the additional linear term $\underline{\underline{K}}_e^{\text{stab}} \underline{u}_e$ in Equation (48) infers a substantial increase in computing time. Factoring out the different iteration count, there is little difference in relative material-evaluation runtime between different levels of hourglass control.

To conclude the discussion of the runtimes, let us investigate the difference between the strain- and the displacement-based implementation in Table 3. We observe that the difference in total runtime is not as significant as for linear elasticity. Indeed, there is still a factor of roughly two for the runtimes of preconditioning, whereas there is only a factor 3.5 difference in material evaluation. This difference is expected to become even smaller for more complex constitutive behaviors.

Last but not least

we consider 5% uniaxial extension, distributed over 50 equidistant load steps. The computed stress-strain diagrams are shown in Figure 8B. As before, there is an absolute difference of above 20 MPa between the full and the reduced integration, and the hourglass-control parameter serves as an interpolation device. The iteration counts recorded in Table 4 again show an increase for reduced integration compared to full integration, and increasing the hourglass-control parameter does decrease the iteration counts. There is a factor of roughly three between full and reduced integration in terms of runtime. Using hourglass control incurs an overhead of about one third in terms of runtime. Due to the many load steps, the iteration counts of the strain- and the displacement-based implementation do not match anymore, as the former stores the last field of the last time step in single and the latter in double precision. The runtimes, however, differ by less than 10%.

To sum up, we observed that hourglass control may relieve the computational burden of fully integrated hexahedron elements significantly. Yet, it does not match the performance of the underintegrated element. Still, if either artefact-free local solution fields are sought or infinite material contrast is of interest, for example, in case of pores and for progressively damaging/strongly plastifying materials, it may be worth investing in a sufficient level of hourglass control.

5 | CONCLUSION

Computational micromechanics based on the FFT draws its strength from selecting solution method and discretization scheme appropriately to exploit synergy effects between both. In contrast to the solution schemes, the discretization techniques do not seem to have reached a satisfying state. Indeed, each of the used discretization schemes come with at least one drawback which make them unsuitable for particular applications. It is therefore desirable to invest scientific effort in a discretization scheme which performs reasonably well for a wide range of scenarios.

This work started off with a discussion of the shortcomings of finite-element discretizations on a regular grid. This discussion could be surprising at first glance, as finite elements are ubiquitous in computational mechanics. However, finite element discretizations draw their strength from their geometric flexibility. If they are used on a regular grid, they are stripped of their main advantage. Indeed, on regular grids, finite-difference discretizations are typically less expensive than finite-element discretizations because the safeguards put in place by the latter to ensure consistency upon element deformation are not necessary. Moreover, higher-order convergence achievable by elements with higher-order shape functions cannot be exploited for composite materials, in general, as the interfaces between the materials limit the regularity of the displacement field, and higher regularity is necessary for higher-order convergence.

For these reasons, we revisited the underintegrated trilinear hexahedron finite element and imported the hourglass-stabilization technology well-known for (mostly explicit) structural simulations with finite elements to FFT-based computational micromechanics. Such an approach can be made efficient by using a displacement-based implementation with a cached preconditioner.

Working with Newton's method and a stored algorithmic tangent severely limits the applicability of FFT-based methods when storage considerations play a role, for example, for industrial-scale applications. This is particularly pronounced for FE discretizations with multiple integration points. As an alternative, it is possible to transfer modern gradient- and fast-gradient-type solution methods available for strain-based implementations to the displacement-based setting. A simplified implementation of the linear conjugate-gradient method emerges by accounting for mixed boundary conditions in a natural way. Moreover, an extension to mixed uniform boundary conditions[109] is immediate.

In the computational investigations, we observed that a well-chosen hourglass control stabilizes the underintegrated trilinear finite element for porous materials and does remove checkerboard artifacts from the solution fields. Using only a single integration point per element is key to rather low runtimes for inelastic and nonlinear constitutive laws. Thus, the hourglass-stabilized element consistently outperformed fully integrated elements both in terms of runtime and accuracy.

To sum up, the hourglass-stabilized trilinear finite element may be seamlessly integrated into FFT-based computational micromechanics, provided a displacement-based implementation is used. It represents a strong candidate for a flexible discretization technique in computational micromechanics. However, the method requires to select the hourglass-stabilization parameter. In the studies at hand, choosing 1% of stabilization led to good results. Yet, further effort may be invested into this.

DATA AVAILABILITY STATEMENT
The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

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

REFERENCES

1. Elliott JC, Dover SD. X-ray microtomography. *J Microsc*. 1982;126(2):211-213.
2. Spowart JE, Mullens HE, Puchalla BT. Collecting and analyzing microstructures in three dimensions: a fully automated approach. *JOM*. 2003;55(10):35-37.
3. Chawla N, Ganesh VV, Wunsch B. Three-dimensional (3D) microstructure visualization and finite element modeling of the mechanical behavior of SiC particle reinforced aluminum composites. *Scr Mater*. 2004;51(2):161-165.
4. Bansal RK, Kubis A, Hull R, Fitz-Gerald J. High-resolution three-dimensional reconstruction: a combined scanning electron microscope and focused ion-beam approach. *J Vac Sci Technol B*. 2006;24(2):554-561.
5. Adams B, Olson T. The mesostructure - Property linkage in polycrystals. *Prog Mater Sci*. 1998;43(1):1-87.
6. Guedes JM, Kikuchi N. Preprocessing and postprocessing for materials based on the homogenization method with adaptive finite element methods. *Comput Methods Appl Mech Eng*. 1990;83(2):143-198.
7. Chevalier Y, Pahr D, Allmer H, Charlebois M, Zysset PK. Validation of voxel-based FE method for prediction of the uniaxial apparent modulus of human trabecular bone using macroscopic mechanical tests and nanoindentation. *J Biomech*. 2007;40:3333-3340.
8. Pahr D, Zysset PK. A comparison of enhanced continuum FE with micro FE models of human vertebral bodies. *J Biomech*. 2009;42:455-462.
9. Arbenz P, van Lenthe GH, Mennel U, Müller R, Sala M. A scalable multi-level preconditioner for matrix-free μ-finite element analysis of human bone structures. *Int J Numer Methods Eng*. 2008;73(7):927-947.
10. Zhu Y, Sifakis E, Teran J, Brandt A. An efficient parallelizable multigrid framework for the simulation of elastic solids. *ACM Trans Graph*. 2010;29(2):1-18.
11. Bedrossian J, Brecht JV, Zhu S, Sifakis E, Teran J. A second order virtual node method for Poisson interface problems on irregular domains. *J Comput Phys*. 2010;229:6405-6426.
12. Naouar N, Vidal-Sallé E, Schneider J, Marie E, Boisse P. Meso-scale FE analyses of textile composite reinforcement deformation based on X-ray computed tomography. *Compos Struct*. 2014;116:156-176.
13. Kabel M, Merkert D, Schneider M. Use of composite voxels in FFT-based homogenization. *Comput Methods Appl Mech Eng*. 2015;294:168-188.
14. Wintiba B, Vasiukov D, Panier S, Lomov SV, Kamel KEM, Massart TJ. Automated reconstruction and conformal discretization of 3D woven composite CT scans with local fiber volume fraction control. *Compos Struct*. 2020;248:112438.
15. Keshav S, Fritzen F, Kabel M. FFT-based homogenization at finite strains using composite boxels (ComBo). arXiv:2204.13624, 2022:1-35.
16. Fritzen F, Böhlke T, Schnack E. Periodic three-dimensional mesh generation for crystalline aggregates based on Voronoi tessellations. *Adv Eng Softw*. 2009;43:701-713.
17. Schneider K, Klusemann B, Bargmann S. Automatic three-dimensional geometry and mesh generation of periodic representative volume elements for matrix-inclusion composites. *Adv Eng Softw*. 2016;99:177-188.
18. Moës N, Cloirec M, Cartraud P, Remacle J-F. A computational approach to handle complex microstructure geometries. *Comput Methods Appl Mech Eng*. 2003;192(28–30):3163-3177.
19. Kafafy R, Lin T, Lin Y, Wang J. Three-dimensional immersed finite element methods for electric field simulation in composite materials. *Int J Numer Methods Eng*. 2005;64:940-972.
20. Kim H-Y, Kim H-G. A novel adaptive mesh refinement scheme for the simulation of phase-field fracture using trimmed hexahedral meshes. *Int J Numer Methods Eng*. 2021;122:1493-1512.
21. Matveev MY, Brown LP, Long AC. Efficient meshing technique for textile composites unit cells of arbitrary complexity. *Compos Struct*. 2020;254:112757.
22. Zangmeister T, Andrä H, Müller R. Comparison of XFEM and Voxelbased FEM for the approximation of discontinuous stress and strain at material interfaces. *Tech Mech*. 2013;33(2):131-141.
23. Lang C, Makhija D, Doostan A, Maute K. A simple and efficient preconditioning scheme for heaviside enriched XFEM. *Comput Mech*. 2014;54:1357-1374.
24. Tian R, Wen L. Improved XFEM - An extra-DOF free, well-conditioning, and interpolating XFEM. *Comput Methods Appl Mech Eng*. 2020;285:639-658.
25. Moulinec H, Suquet P. A fast numerical method for computing the linear and nonlinear mechanical properties of composites. *C R Acad Sci Ser II*. 1994;318(11):1417-1423.
26. 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.
27. Zeller R, Dederichs PH. Elastic constants of polycrystals. *Phys Status Solidi*. 1973;55(2):831-842.
28. Kröner E. Bounds for effective elastic moduli of disordered materials. *J Mech Phys Solids*. 1977;25(2):137-155.
29. Mura T. *Micromechanics of Defects in Solids*. Martinus Nijhoff; 1987.
30. 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.
31. 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.
32. 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.

33. 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.

34. Brisard S, Dormieux L. FFT-based methods for the mechanics of composites: a general variational framework. *Comput Mater Sci*. 2010;49(3):663-671.

35. 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.

36. 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.

37. 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.

38. Schneider M. A dynamical view of nonlinear conjugate gradient methods with applications to FFT-based computational micromechanics. *Comput Mech*. 2020;66:239-257.

39. Lahellec N, Michel JC, Moulinec H, Suquet P. Analysis of inhomogeneous materials at large strains using fast Fourier transforms. Proceedings of the IUTAM Symposium on Computational mechanics of Solid Materials at Large Strains; Vol. 108, 2003:247-258; Springer.

40. 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.

41. 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.

42. 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 Solids Struct*. 2019;161:111-126.

43. Wicht D, Schneider M, Böhlke T. On Quasi-Newton methods in FFT-based micromechanics. *Int J Numer Methods Eng*. 2020;121(8):1665-1694.

44. Schneider M. On the Barzilai-Borwein basic scheme in FFT-based computational homogenization. *Int J Numer Methods Eng*. 2019;118(8):482-494.

45. Givois F, Kabel M, Gauger N. QFT-based homogenization. arXiv:2207.12949, 2022:1-21.

46. Schneider M. Convergence of FFT-based homogenization for strongly heterogeneous media. *Math Methods Appl Sci*. 2015;38(13):2761-2778.

47. Vondřejc J, Zeman J, Marek I. An FFT-based Galerkin method for homogenization of periodic media. *Comput Math Appl*. 2014;68(3):156-173.

48. Mishra N, Vondřejc J, Zeman J. A comparative study on low-memory iterative solvers for FFT-based homogenization of periodic media. *J Comput Phys*. 2016;321:151-168.

49. Bellis C, Suquet P. Geometric variational principles for computational homogenization. *J Elast*. 2019;137:119-149.

50. 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.

51. 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.

52. Willot F. Fourier-based schemes for computing the mechanical response of composites with accurate local fields. *Comptes Rendus Mécanique*. 2015;343:232-245.

53. Vidyasagar A, Tan WL, Kochmann DM. Predicting the effective response of bulk polycrystalline ferroelectric ceramics via improved spectral phase field methods. *Mech Phys Solids*. 2017;106:133-151.

54. Vidyasagar A, Tutcuoglu AD, Kochmann DM. Deformation patterning in finite-strain crystal plasticity by spectral homogenization with application to magnesium. *Comput Methods Appl Mech Eng*. 2018;335:584-609.

55. Schneider M, Ospald F, Kabel M. Computational homogenization of elasticity on a staggered grid. *Int J Numer Methods Eng*. 2016;105(9):693-720.

56. 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.

57. Eloh KS, Jacques A, Berbenni S. Development of a new consistent discrete Green operator for FFT-based methods to solve heterogeneous problems with eigenstrains. *Int J Plast*. 2019;116:1-23.

58. 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.

59. Leuschner M, Fritzen F. Fourier-accelerated nodal solvers (FANS) for homogenization problems. *Comput Mech*. 2018;62:359-392.

60. Ladecký M, Leute RJ, Falsafi A, et al. Optimal FFT-accelerated finite element solver for homogenization. arXiv:2203.02962, 2022:1-30.

61. Caforio F, Imperiale S. A high-order spectral element fast Fourier transform for the Poisson equation. *SIAM J Sci Comput*. 2019;41(5):A2747-A2771.

62. Tu F, Jiao Y, Zhou X, Cheng Y, Tan F. The implementation of B-splines to Hashin and Shtrikman variational principle based FFT method for the homogenization of composite. *Int J Solids Struct*. 2020;191–192:133-145.

63. Saenger EH, Gold N, Shapiro SA. Modeling the propagation of elastic waves using a modified finite-difference grid. *Wave Motion*. 2000;31:77-92.

64. Saenger EH, Bohlen T. Finite-difference modeling of viscoelastic and anisotropic wave propagation using the rotated staggered grid. *Geophysics*. 2004;69:583-591.

65. Schneider M. Lippmann-Schwinger solvers for the computational homogenization of materials with pores. *Int J Numer Methods Eng*. 2020;121(22):5017-5041.

66. Grimm-Strehle H, Kabel M. FFT-based homogenization with mixed uniform boundary conditions. *Int J Numer Methods Eng*. 2021;122(23):7241-7265.

67. Lirkov ID, Margenov SD, Vassilevski PS. Circulant block-factorization preconditioners for elliptic problems. *Computing*. 1994;53(1):59-74.

68. Bakhvalov NS, Knyazev AV. Fictitious domain methods and computation of homogenized properties of composites with a periodic structure of essentially different components. In: Marchuk GI, ed. *Numerical Methods and Applications*. CRC Press; 1994:221-276.

69. Bakhvalov NS, Knyazev AV, Parashkevov RR. Extension theorems for Stokes and Lamé equations for nearly incompressible media and their applications to numerical solution of problems with highly discontinuous coefficients. *Numer Linear Algebra Appl*. 2002;9(2):115-139.

70. Flanagan DP, Belytschko T. A uniform strain hexahedron and quadrilateral with orthogonal hourglass control. *Int J Numer Methods Eng*. 1981;17:679-706.

71. 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.

72. Belytschko T, Bindeman LP. Assumed strain stabilization of the eight node hexahedral element. *Comput Methods Appl Mech Eng*. 1993;105:225-260.

73. Matouš K, Geers MGD, Kouznetsova VG, Gillman A. A review of predictive nonlinear theories for multiscale modeling of heterogeneous materials. *J Comput Phys*. 2017;330:192-220.

74. Pellerin J, Verhetsel K, Remacle J-F. There are 174 subdivisions of the hexahedron into tetrahedra. *ACM Trans Graph*. 2018;37(6):266.

75. Simo JC, Rifai MS. A class of mixed assumed strain methods and the method of incompatible modes. *Int J Numer Methods Eng*. 1990;29:1595-1638.

76. Smith IM, Kidger DJ. Properties of the 20-node brick. *Int J Numer Anal Methods Geomech*. 1991;15:871-891.

77. Rannacher R, Turek S. Simple nonconforming quadrilaterial Stokes element. *Numer Methods Part Differ Equ*. 1992;8:97-111.

78. Schneider M. A review of non-linear FFT-based computational homogenization methods. *Acta Mech*. 2021;232:2051-2100.

79. 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.

80. Lucarini S, Segurado J. DBFFT: a displacement based FFT approach for non-linear homogenization of the mechanical behavior. *Int J Eng Sci*. 2019;114:103131.

81. Bellis C, Moulinec H, Suquet P. Eigendecomposition-based convergence analysis of the Neumann series for laminated composites and discretization error estimation. *Int J Numer Methods Eng*. 2020;121:201-232.

82. Ortiz L, Stainier L. The variational formulation of viscoplastic constitutive updates. *Comput Methods Appl Mech Eng*. 1999;171:419-444.

83. Miehe C. Strain-driven homogenization of inelastic microstructures and composites based on an incremental variational formulation. *Int J Numer Methods Eng*. 2002;55:1285-1322.

84. Blühdorn J, Gauger NR, Kabel M. AutoMat: automatic differentiation for generalized standard materials on GPUs. *Comput Mech*. 2022;69:589-613.

85. Kabel M, Fliegener S, Schneider M. Mixed boundary conditions for FFT-based homogenization at finite strains. *Comput Mech*. 2016;57(2):193-210.

86. Kabel M. Mixed strain/stress gradient loadings for FFT-based computational homogenization methods. *Comput Mech*. 2022;70:281-308.

87. Suquet PM. Elements of homogenization for inelastic solid mechanics. In: Sanchez-Palencia E, Zaoui A, eds. *Homogenization Techniques for Composite Media*. Springer; 1987:1157-1171.

88. Schneider M. On the mathematical foundations of the self-consistent clustering analysis for non-linear materials at small strains. *Comput Methods Appl Mech Eng*. 2019;354:783-801.

89. Hughes TJR. *The Finite Element Method. Linear Static and Dynamic Finite Element Analysis*. Dover; 2000.

90. Thomson W. Elements of a mathematical theory of elasticity. *Philos Trans R Soc*. 1856;146:481-498.

91. Mandel J. Généralisation de la théorie de plasticité de WT Koiter. *Int J Solids Struct*. 1965;1:273-295.

92. Dunavant DA. Efficient symmetrical cubature rules for complete polynomials of high degree over the unit cube. *Int J Numer Methods Eng*. 1986;23:397-407.

93. Willot F, Abdallah B, Pellegrini Y-P. Fourier-based schemes with modified Green operator for computing the electrical response of heterogeneous media with accurate local fields. *Int J Numer Methods Eng*. 2014;98:518-533.

94. Rudin W. *Fourier Analysis on Groups*. Interscience Publishers; 1962.

95. Barzilai J, Borwein JM. Two-point step size gradient methods. *IMA J Numer Anal*. 1988;8:141-148.

96. Golub GH, Van Loan CF. *Matrix Computations*. 4th ed. The John Hopkins University Press; 2013.

97. Fletcher R, Reeves C. Function minimization by conjugate gradients. *Comput J*. 1964;7:149-154.

98. Dai YH. Convergence analysis of nonlinear conjugate gradient methods. In: Wang Y, Yang C, Yagola AG, eds. *Optimization and Regularization for Computational Inverse Problems and Applications*. Springer; 2011:1157-1171.

99. Stainier L, Ortiz M. Study and validation of a variational theory of thermo-mechanical coupling in finite visco-plasticity. *Int J Solids Struct*. 2010;47:155-168.

100. Wu L, Adam L, Doghri I, Noels L. An incremental-secant mean-field homogenization method with second statistical moments for elasto-visco-plastic composite materials. *Mech Mater*. 2017;114:180-200.

101. Daphalapurkar NP, Wang F, Fu B, Lu H, Komanduri R. Determination of mechanical properties of sand grains by nanoindentation. *Exp Mech*. 2011;51:719-728.

102. Wichtmann T, Triantafyllidis T. On the influence of the grain size distribution curve on P-wave velocity, constrained elastic modulus M_{max} and Poisson's ratio of quartz sands. *Soil Dyn Earthq Eng*. 2010;30(8):757-766.

103. Sanditov DS, Mantatov VV, Sanditov BD. Poisson ratio and plasticity of glasses. *Tech Phys*. 2009;54(4):594-596.

104. Despande VS, Fleck NA, Ashby MF. Effective properties of the octet-truss lattice material. *J Mech Phys Solids*. 2001;49(8):1747-1769.

105. Schneider M. On non-stationary polarization methods in FFT-based computational micromechanics. *Int J Numer Methods Eng*. 2021;122(22):6800-6821.

106. Lucarini S, Cobian L, Voitus A, Segurado J. Adaptation and validation of FFT methods for homogenization of lattice based materials. *Comput Methods Appl Mech Eng*. 2022;388:114223.

107. Schneider M, Hofmann T, Andrä H, et al. Modeling the microstructure and computing effective elastic properties of sand core materials. *Int J Solids Struct*. 2018;143:1-17.

108. Schneider M. The sequential addition and migration method to generate representative volume elements for the homogenization of short fiber reinforced plastics. *Comput Mech*. 2017;59:247-263.

109. Grimm-Strehle 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.

110. Ettemeyer F, Lechner P, Hofmann T, et al. Digital sand core physics: predicting physical properties of sand cores by simulations on digital microstructures. *Int J Solids Struct*. 2020;188-189:155-168.