Received: 1 August 2023 | Revised: 28 October 2023 | Accepted: 29 October 2023

DOI: 10.1002/nme.7399

RESEARCH ARTICLE

WILEY

Adaptive material evaluation by stabilized octree and sandwich coarsening in FFT-based computational micromechanics

Matthias Kabel[1] | **Matti Schneider**[1,2]

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

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

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

Funding information
German Federal Ministry for Economic Affairs and Climate Action, Grant/Award Number: 03LB3044B; HORIZON EUROPE European Research Council, Grant/Award Number: 101040238

Abstract

The computational efficiency of FFT-based computational micromechanics is deeply rooted in the underlying regular, that is, Cartesian, discretization. The bottleneck for most industrial applications is evaluating the typically rather expensive constitutive law on the regular grid. In the work at hand, we exploit coarsening strategies to evaluate the material law with the intention of speeding up the overall computation time while retaining the level of achieved accuracy. Inspired by wavelet-compression techniques, we form aggregates of voxels where the local strain tensors are close, and compute the stresses on these coarsened elements. If done naively, such a strategy will lead to intrinsic instabilities whose origin is apparent from a mathematical perspective. As a remedy, we introduce a stabilization technique which is inspired by hourglass control well-known for underintegrated finite elements. We introduce octree as well as sandwich coarsening, discuss the handling of internal variables, report on the efficient implementation of the concepts and demonstrate the effectiveness of the developed technology on simple as well as industrial examples.

KEYWORDS

adaptive wavelets, computational homogenization, FFT-based methods, inelasticity, nonlinear micromechanics, octree

1 | INTRODUCTION

1.1 | State of the art

The approach to computational micromechanics based on the fast Fourier transform (FFT), initially proposed by Moulinec–Suquet[1,2] and further developed by a vivid community, is widely regarded as one of the most efficient numerical strategy for this purpose. Almost all variants of FFT-based homogenization have in common that they operate on a regular grid and utilize the fast Fourier transform to evaluate Green's operator within a Lippmann–Schwinger type solution framework. The success of the strategy rests, on the one hand, on the rather efficient FFT implementations available,[3,4] and, on the other hand, on the intrinsic fusion of discretization and numerical solution strategy, for instance incorporating an intrinsic preconditioning strategy which renders the iteration count essentially independent of the resolution. Another key factor for the efficiency of these methods is their use of simple data structures. In fact, all data is stored in

arrays, and all operations (except for the FFT) involve simple loops over this data. As the already optimized FFT libraries are none of the user's concern, little extra computational overhead is created except for evaluating Green's operator, a linear operation, and computing the material law.

Over time, different flavors of FFT-based methods were developed. It was realized that rather general discretization schemes[5(section 2)] can be employed, as long as they operate on a regular grid, and used in conjunction with Lippmann–Schwinger type solvers[5(section 3)].

It appears difficult to extend FFT-based methods beyond the regular-grid setting, preserving the computational advantages of the schemes in the process. Eyre-Milton[6] were probably the first to exploit multiple grid levels in this context. Their approach, however, used hierarchical grids to speed up the solution process on the finest (still regular) grid by what might be called a cascadic multigrid method.[7]

Nkoumbou Kaptchouang and Gélébart[8] studied an FFT-based computational method with a rectangular local refinement which they solve with the local-defect correction method.

Only recently, Kaiser et al.[9] showed that FFT-based computational homogenization methods can also be used in an adaptive manner, removing the constraint of working with a regular grid a little bit. To be more precise, Kaiser et al.[9] employ interpolating wavelets[10] in two ways. For a start, they consider the Deslauriers-Dubuc wavelets[11] on a regular grid and derive the associated Eshelby–Green operator. In a second step, they show that they may also dispense with small wavelet coefficients when evaluating the constitutive law, potentially speeding up the computations significantly. Kaiser et al.[9] do not completely leave the terrain of regular-grid-solvers, as there is still a fine-scale regular grid in the background, where the FFT is computed on and which is used to apply the Eshelby–Green operator.

Please note the difference between the approaches of Nkoumbou Kaptchouang and Gélébart[8] and Kaiser et al.[9] Nkoumbou Kaptchouang and Gélébart[8] use a regular grid as the coarse grid and *refine* the grid, whereas Kaiser et al.[9] use the regular grid as the fine grid and consider *coarsening*. The difference in these approaches manifests when applying the FFT. For the LDC approach,[8] the FFT is applied on the coarse scale, and some extra strategy needs to be used to solve the fine scale. In contrast, the wavelet technique[9] uses FFT on the fine grid and seeks to avoid extraneous material evaluations due to the coarse grid, considering those as predominantly contributing to the overall runtime.

1.2 | Contributions

Inspired by the work of Kaiser et al.,[9] we study strategies to reduce the effort when computing the material law in FFT-based computational homogenization methods. However, we do not use wavelets for discretization, but retain traditional discretization schemes, reserving the compression for evaluating the constitutive law only.

Actually, due to the discontinuities in the strain fields and the complexity of industrial-scale microstrostructures, we wanted to rely on simple yet robust wavelets like the Haar wavelet.[12,13] However, we realized that using (tensor-product) Haar wavelets is equivalent to working with octrees.[14] Thus, we slowly moved away from wavelets altogether, and started to study general coarsening strategies for evaluating the constitutive law, see Section 2.1. Interestingly, there are close links to the transformation field analysis of Dvorak and co-workers,[15,16] but also to working with underintegrated elements in the context of the finite-element method.[17,18] We noticed that it is necessary to introduce a suitable stabilization term to avoid a pathological degeneration of the resulting nonlinear system of equations, very much like the introduction of hourglass stabilization[17,18] for underintegrated finite elements, see Section 2.2. It turns out that an elastic stabilization with the material tangent leads to the highest accuracy. Unfortunately, using the current material tangent is incompatible with retaining favorable numerical properties like monotonicity and symmetric material tangents, so that we use the material tangent of the previous load step. Working with inelastic material models involving internal variables imposes further restrictions, discussed in Section 2.3: in a nutshell, only refinement (and no coarsening) is allowed when going from one time step to the next.

We introduce flexible coarsening strategies based on octrees in Section 3.1, also working out a straightforward enhancement which we call sandwich coarsening. The integration into an existing FFT-based computational micromechanics code is discussed in Section 3.2. Based on the data structure of a linear octree,[19] the implementation is essentially array-based and we may preserve the original computational speed of regular-grid methods. Last but not least, we study the introduced techniques for a number of challenging micromechanical problems in Section 4.

2 | COARSENING IN COMPUTATIONAL HOMOGENIZATION

2.1 | The homogenization problem

Suppose a rectangular cell $Y \subseteq \mathbb{R}^d$ in $d = 2, 3$ dimensions is given. A periodic first-order homogenization problem at small strains seeks a periodic displacement-fluctuation field $\boldsymbol{u} \in H^1_\#(Y;\mathbb{R}^d)$, a local strain field $\boldsymbol{\varepsilon} \in L^2(Y;\mathrm{Sym}(d))$ and a local stress field $\boldsymbol{\sigma} \in L^2(Y;\mathrm{Sym}(d))$, s.t. kinematic compatibility

$$\boldsymbol{\varepsilon} = \overline{\boldsymbol{\varepsilon}} + \nabla^s \boldsymbol{u}, \tag{1}$$

involving a prescribed macroscopic strain $\overline{\boldsymbol{\varepsilon}}$ as well as the symmetrized gradient operator ∇^s, and the quasi-static stress equilibrium

$$\mathrm{div}\, \boldsymbol{\sigma} = 0 \tag{2}$$

are satisfied, where the divergence operator div arises as the (negative of the) adjoint of the symmetrized gradient operator, that is, the identity

$$\langle \mathrm{div}\, \boldsymbol{\sigma} \cdot \boldsymbol{v} \rangle_Y = -\langle \boldsymbol{\sigma} : \nabla^s \boldsymbol{v} \rangle_Y \quad \text{holds for all} \quad \boldsymbol{\sigma} \in L^2(Y;\mathrm{Sym}(d)) \quad \text{and} \quad \boldsymbol{v} \in H^1_\#(Y;\mathbb{R}^d). \tag{3}$$

Here, $\mathrm{Sym}(d)$ refers to the vector space of symmetric $d \times d$-tensors and by $\langle q \rangle_Y$ we denote the average of any integrable field q on the cell Y, that is,

$$\langle q \rangle_Y \equiv \fint_Y q(\boldsymbol{x})\, dx \equiv \int_Y q(\boldsymbol{x})\, dx \Big/ \int_Y dx. \tag{4}$$

The Equations (1) and (2) are complemented by a constitutive law. To illustrate our ideas, for the section at hand, we assume the material behavior to be governed by a function

$$\mathbb{s} : Y \times \mathrm{Sym}(d) \to \mathrm{Sym}(d), \tag{5}$$

s.t. strain and stress are related by the equation

$$\boldsymbol{\sigma} = \mathbb{s}(\cdot, \boldsymbol{\varepsilon}), \tag{6}$$

which is a shorthand notation for

$$\boldsymbol{\sigma}(\boldsymbol{x}) = \mathbb{s}(\boldsymbol{x}, \boldsymbol{\varepsilon}(\boldsymbol{x})) \quad \text{for almost every} \quad \boldsymbol{x} \in Y. \tag{7}$$

Practical stress functions satisfy a number of salient properties. These include the following.

1. The stress function $\mathbb{s}$ is measurable in the first variable $\boldsymbol{x}$.
2. The stress corresponding to the unloaded state is square integrable

$$\left\langle \|\mathbb{s}(\cdot, 0)\|^2 \right\rangle_Y < \infty. \tag{8}$$

3. The stress function is uniformly Lipschitz continuous in the strain, that is, there is a constant α_+, s.t. the inequality

$$\|\mathbb{s}(\boldsymbol{x}, \boldsymbol{\varepsilon}_1) - \mathbb{s}(\boldsymbol{x}, \boldsymbol{\varepsilon}_2)\| \leq \alpha_+ \|\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2\| \tag{9}$$

holds for almost every $\boldsymbol{x} \in Y$ and all $\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2 \in \mathrm{Sym}(d)$ where we use the Frobenius norm

$$\|\boldsymbol{\varepsilon}\| = \sqrt{\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon}}, \quad \boldsymbol{\varepsilon} \in \mathrm{Sym}(d). \tag{10}$$

These properties ensure that the stress function $\mathbb{s}$ gives rise to a well-defined Nemytskii operator on the Lebesgue space L^2, that is, the association

$$L^2(Y;\mathrm{Sym}(d)) \ni \boldsymbol{\varepsilon} \mapsto (Y \ni \boldsymbol{x} \mapsto \mathbb{s}(\boldsymbol{x},\boldsymbol{\varepsilon}(\boldsymbol{x})) \in \mathrm{Sym}(d)) \in L^2(Y;\mathrm{Sym}(d)) \tag{11}$$

is reasonable[20(section 2.1)]. We will suppress the $\boldsymbol{x}$-dependence in the operator $\mathbb{s}$ for simplicity of notation.

In addition to these properties, we assume that the stress operator satisfies a monotonicity condition, that is, there is a non-negative constant α_-, s.t. the inequality

$$(\mathbb{s}(\boldsymbol{x},\boldsymbol{\varepsilon}_1) - \mathbb{s}(\boldsymbol{x},\boldsymbol{\varepsilon}_2)) : (\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2) \geq \alpha_- \, \|\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2\|^2 \tag{12}$$

holds for all $\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2 \in \mathrm{Sym}(d)$ and almost every $\boldsymbol{x} \in Y$. If the inequality (12) holds for a positive constant α_-, we call the stress functions $\mathbb{s}$ *strongly* monotone. In case of vanishing α_-, we speak of a monotone stress operator. Monotone stress operators arise, for instance, when discretizing strain-hardening materials in time.

If the stress operator is strongly monotone and Lipschitz continuous, the homogenization problem

$$\mathrm{div}\, \mathbb{s}(\overline{\boldsymbol{\varepsilon}} + \nabla^s \boldsymbol{u}) = 0, \tag{13}$$

comprising the equations (1), (2), and (6), will be well-posed, that is, there exists a solution which is furthermore unique. In practice, we frequently encounter stress operators which are not monotone. Yet, it is helpful to investigate whether monotonicity properties are preserved by the constructions which we will undertake in the work at hand. In fact, such preservation properties are helpful when designing *stable* numerical algorithms.

Once the displacement-fluctuation field $\boldsymbol{u}$ solving the micromechanics problem (13) is found, the effective stress

$$\overline{\boldsymbol{\sigma}} = \left\langle \mathbb{s}(\overline{\boldsymbol{\varepsilon}} + \nabla^s \boldsymbol{u}) \right\rangle_Y \tag{14}$$

may be computed in post-processing.

2.2 | Stabilized coarsening

We would like to decompose the strain field $\boldsymbol{\varepsilon} \in L^2(Y;\mathrm{Sym}(d))$ into fine- and coarse-scale features. This decomposition will be achieved through a suitable orthogonal projector $\mathbb{P}$, that is, a bounded linear operator $\mathbb{P} \in L(L^2(Y;\mathrm{Sym}(d)))$ which is idempotent, that is, the condition

$$\mathbb{P} : \mathbb{P} : \boldsymbol{\varepsilon} = \mathbb{P} : \boldsymbol{\varepsilon} \tag{15}$$

is satisfied for each strain field $\boldsymbol{\varepsilon} \in L^2(Y;\mathrm{Sym}(d))$, and the Pythagorean Theorem

$$\|\boldsymbol{\varepsilon}\|_{L^2}^2 = \|\mathbb{P} : \boldsymbol{\varepsilon}\|_{L^2}^2 + \|\mathbb{Q} : \boldsymbol{\varepsilon}\|_{L^2}^2 \tag{16}$$

holds for each strain field $\boldsymbol{\varepsilon} \in L^2(Y;\mathrm{Sym}(d))$, expressed in terms of the complementary projector $\mathbb{Q} = \mathrm{Id} - \mathbb{P}$. We use the normalized L^2-norm

$$\|\boldsymbol{\varepsilon}\|_{L^2} = \langle \boldsymbol{\varepsilon} : \boldsymbol{\varepsilon} \rangle_Y^{\frac{1}{2}} \tag{17}$$

in Equation (16).

The projectors we consider will always have a specific form. We suppose that a finite number Y_α of open, mutually disjoint and nonempty subsets of the cell Y are given, s.t. the union of their closures cover the original cell Y. Writing χ_α for the associated characteristic function, that is, the function $\chi_\alpha : Y \to \{0,1\}$ which equals unity inside the domain Y_α and vanishes otherwise, we may construct a projection operator

$$\mathbb{P} : \boldsymbol{\varepsilon} = \sum_\alpha \langle \chi_\alpha \boldsymbol{\varepsilon} \rangle_Y / \langle \chi_\alpha \rangle_Y \, \chi_\alpha. \tag{18}$$

It is readily established that idempotency (15) and the Pythagorean Theorem (16) hold.

We call the set Y_α *clusters* or *cells*, where the latter name will turn out to be more appropriate for the chosen cluster shape, see Section 3.1.

With the projector $\mathbb{P}$ at hand, we may decompose any strain field $\boldsymbol{\varepsilon} \in L^2(Y;\mathrm{Sym}(d))$

$$\boldsymbol{\varepsilon} = \mathbb{P} : \boldsymbol{\varepsilon} + \mathbb{Q} : \boldsymbol{\varepsilon}, \quad \mathbb{Q} = \mathrm{Id} - \mathbb{P}, \tag{19}$$

into *coarse-scale* features $\mathbb{P} : \boldsymbol{\varepsilon}$ and *fine-scale* features $\mathbb{Q} : \boldsymbol{\varepsilon}$. With this decomposition of fields into coarse- and fine-scale features, it appears natural to define a *coarsened stress operator*

$$\tilde{\mathbb{s}}^{\mathtt{naive}} : L^2(Y;\mathrm{Sym}(d)) \to L^2(Y;\mathrm{Sym}(d)), \quad \boldsymbol{\varepsilon} \mapsto \mathbb{s}(\mathbb{P} : \boldsymbol{\varepsilon}), \tag{20}$$

which extracts the coarse-scale feature of the strain field and computes the corresponding stress field. This naive approach, however, does not lead to a numerically stable strategy. In fact, any strain field with vanishing fine parts, that is, whenever the condition $\mathbb{P} : \boldsymbol{\varepsilon} = 0$ holds, does not infer a stress, that is, gives rise to a zero-stress mode. In a sense, this situation is rather similar to hourglass modes emerging from using too few integration points for the ansatz functions of finite elements. To effectively suppress hourglass modes, *hourglass stabilization* techniques were established,[17,18] which add an additional *hourglass stiffness* to render the kernel of the finite-element stiffness matrix trivial.

We follow a similar strategy to stabilize the naively coarsened stress operator (20). We suppose that stabilization stiffnesses $\mathbb{C}^{\mathtt{stab}}$ are given, that is, a (measurable) mapping

$$\mathbb{C}^{\mathtt{stab}} : Y \to L(\mathrm{Sym}(d)), \tag{21}$$

s.t. $\mathbb{C}^{\mathtt{stab}}$ is a linear elastic stiffness tensor for every $\boldsymbol{x} \in Y$ which furthermore satisfies the inequalities

$$\alpha_- \, \|\boldsymbol{\varepsilon}\|^2 \leq \boldsymbol{\varepsilon} : \mathbb{C}^{\mathtt{stab}}(\boldsymbol{x}) : \boldsymbol{\varepsilon} \leq \alpha_+ \|\boldsymbol{\varepsilon}\|^2 \tag{22}$$

for every strain $\boldsymbol{\varepsilon} \in \mathrm{Sym}(d)$ and every point $\boldsymbol{x} \in Y$ with the constants $\alpha_\pm$ from the conditions (9) and (12). In fact, inspired by hourglass control, we want the elastic response of the coarsened stress operator to equal the elastic response of the fine stress operator (11), which requires the stiffness (21) to be heterogeneous. The inequalities (22) ensure that the conditioning of the resulting system of equations does not deteriorate.

With these preliminaries at hand, we define the *stabilized coarsened stress operator*

$$\tilde{\mathbb{s}} : L^2(Y;\mathrm{Sym}(d)) \to L^2(Y;\mathrm{Sym}(d)), \quad \boldsymbol{\varepsilon} \mapsto \mathbb{s}(\mathbb{P} : \boldsymbol{\varepsilon}) + \mathbb{C}^{\mathtt{stab}} : \mathbb{Q} : \boldsymbol{\varepsilon}, \tag{23}$$

which evaluates the original stress operator $\mathbb{s}$ on the coarsened strain features only and adds back the fine-scale features via the stabilization stiffness $\mathbb{C}^{\mathtt{stab}}$. The construction (23) will turn out to decrease computational effort provided evaluating the stress operator $\mathbb{s}$ is significantly more expensive than computing the linear elastic stress-strain relationship. Please notice that if the operator $\mathbb{s}$ has a symmetric material tangent, so will the operator $\tilde{\mathbb{s}}$.

For the stress operator (23) to have further salient properties it is necessary that the projector $\mathbb{P}$ respects both the heterogeneity of the original stress operator and of the stabilization stiffness $\mathbb{C}^{\mathtt{stab}}$, that is, the identities

$$\mathbb{s}(\mathbb{P} : \boldsymbol{\varepsilon}) = \mathbb{P} : \mathbb{s}(\mathbb{P} : \boldsymbol{\varepsilon}) \tag{24}$$

and

$$\mathbb{C}^{\mathtt{stab}} : \mathbb{Q} : \boldsymbol{\varepsilon} = \mathbb{Q} : \mathbb{C}^{\mathtt{stab}} : \mathbb{Q} : \boldsymbol{\varepsilon} \tag{25}$$

hold for all fields $\boldsymbol{\varepsilon} \in L^2(Y;\mathrm{Sym}(d))$. To get an intuition, the first condition (24) ensures that the stress field of a coarse-scale strain is purely coarse-scale, that is, has no fine-scale contributions. The complementary, the second condition (25) ensures that the stabilized stress of a fine-scale contribution is solely fine-scale.

For the special setup (18), the condition (24) entails that the decomposition $\{Y_\alpha\}$ conforms to the heterogeneity of the stress operator $\mathbb{s}$, that is, each cluster Y_α contains only a single material. For this case, the condition (25) is satisfied for all strain fields $\boldsymbol{\varepsilon}$ if and only if the stabilization stiffnesses $\mathbb{C}^{\mathtt{stab}}$ is constant on each cluster Y_α.

With these properties at hand, we may show the following characteristics:

1. The operator $\tilde{\mathbb{S}}$ is α_+-Lipschitz continuous.
2. The operator $\tilde{\mathbb{S}}$ is (strongly) monotone with constant α_+ provided the operator $\mathbb{S}$ has this property.
3. For any strain field $\boldsymbol{\varepsilon} \in L^2(Y;\mathrm{Sym}(d))$, the exact error expression

$$\mathbb{S}(\boldsymbol{\varepsilon}) - \tilde{\mathbb{S}}(\boldsymbol{\varepsilon}) = \int_0^1 \left[\frac{\partial \mathbb{S}}{\partial \boldsymbol{\varepsilon}}(\mathbb{P} : \boldsymbol{\varepsilon} + s\, \mathbb{Q} : \boldsymbol{\varepsilon}) - \mathbb{C}^{\mathtt{stab}}\right] : \mathbb{Q} : \boldsymbol{\varepsilon}\, ds \tag{26}$$

holds provided the right-hand side makes sense. In particular, we obtain the error estimate

$$\|\mathbb{S}(\boldsymbol{\varepsilon}) - \tilde{\mathbb{S}}(\boldsymbol{\varepsilon})\|_{L^2} \leq 2\alpha_+ \|\mathbb{Q} : \boldsymbol{\varepsilon}\|_{L^2}. \tag{27}$$

Loosely speaking, the first attribute quantifies the continuity properties of the operator $\tilde{\mathbb{S}}$, the second characteristic ensures that the coarsening preserves the monotonicity of the original stress operator, whereas the last feature guarantees that the error we make when evaluating $\tilde{\mathbb{S}}$ instead of $\mathbb{S}$ is small provided the small-scale features $\mathbb{Q} : \boldsymbol{\varepsilon}$ are small to begin with.

To preserve the flow of the reader, the validity of properties 1–3. is shown in the Appendix. We close this subsection with a small discussion of these properties. Before we start let us emphasize that the constitutive law

$$\boldsymbol{\sigma} = \tilde{\mathbb{S}}(\boldsymbol{\varepsilon}) \tag{28}$$

encoded by the coarsened stress operator (23) is *nonlocal*, in general, in contrast to the previous constitutive law (6). In fact, the projection operation, compare Equation (18), may involve averaging.

For the second property to hold, including the stabilization stiffness $\mathbb{C}^{\mathtt{stab}}$ is essential. In fact, the decomposition (19) has some similarities to the splitting of a strain field on a hexahedron into its mean and the remaining hourglass modes. In this context, neglecting the hourglass modes will also turn out to be detrimental, and stabilization in terms of appropriate hourglass control is imperative.[17,18,21]

The last property will also give us a hint how to select the stabilization stiffness $\mathbb{C}^{\mathtt{stab}}$. The error representation (26) suggests selecting

$$\mathbb{C}^{\mathtt{stab}} = \frac{\partial \mathbb{S}}{\partial \boldsymbol{\varepsilon}}(\mathbb{P} : \boldsymbol{\varepsilon}) \tag{29}$$

due to the Taylor expansion

$$\mathbb{S}(\boldsymbol{\varepsilon}) = \mathbb{S}(\mathbb{P} : \boldsymbol{\varepsilon}) + \frac{\partial \mathbb{S}}{\partial \boldsymbol{\varepsilon}}(\mathbb{P} : \boldsymbol{\varepsilon}) : \mathbb{Q} : \boldsymbol{\varepsilon} + O\big(\|\mathbb{Q} : \boldsymbol{\varepsilon}\|_{L^2}^2\big), \tag{30}$$

provided the second derivative of $\mathbb{S}$ is uniformly bounded. However, the prescription (29) is not feasible, as we defined the stabilizing stiffness $\mathbb{C}^{\mathtt{stab}}$ to be *independent* of the current strain $\boldsymbol{\varepsilon}$. The reason for the latter condition is that a dependence on the strain interferes with the monotonicity condition, that is, property 2. This mathematical caution was backed up by computational experiments where using the current material tangent for stabilization did, in fact, not turn out to be stable. Instead, with inelasticity and several loading steps in mind, we suggest to choose the material tangent of the *previous load step* $\boldsymbol{\varepsilon}^n$, that is,

$$\mathbb{C}^{\mathtt{stab}} = \frac{\partial \mathbb{S}^n}{\partial \boldsymbol{\varepsilon}}(\mathbb{P}^n : \boldsymbol{\varepsilon}^n), \tag{31}$$

where the superindex n indicates to the previous load step. For typical problems of small-strain inelasticity, the inequalities $\alpha_+^n \leq \alpha_+$ and $\alpha_-^n \geq \alpha_-$ hold, that is, the Lipschitz constant may only increase, whereas the monotonicity constant typically decreases upon increased loading.

2.3 | Handling internal variables

Typically, the stress function $\mathbb{S}$ defined in Equation (5) arises by an implicit time discretization of an inelastic evolution in practice. The previous considerations did not make the handling of internal variables explicit. We chose this path deliberately to focus on the relevant ideas.

The purpose of the section at hand is to fill this gap and discuss how to handle internal variables in case of *changing* coarsening. In fact, it will turn out that we may only refine the coarsening in order to retain a consistent evolution of the internal variables.

We consider a heterogeneous generalized standard material (GSM)[22,23] to govern the mechanical behavior of the material, given in the form of a Banach space Z of internal variables, a (continuously differentiable in $\boldsymbol{\varepsilon}$) free energy density

$$w : Y \times \mathrm{Sym}(d) \times Z \to \mathbb{R}_{\geq 0}, \quad (\boldsymbol{x}, \boldsymbol{\varepsilon}, \underline{z}) \mapsto w(\boldsymbol{x}, \boldsymbol{\varepsilon}, \underline{z}), \tag{32}$$

and a dissipation potential

$$\phi : Y \times Z \to \mathbb{R}_{\geq 0} \cup \{+\infty\}, \quad (\boldsymbol{x}, \underline{\dot{z}}) \mapsto \phi(\boldsymbol{x}, \underline{\dot{z}}), \tag{33}$$

For every $\boldsymbol{x} \in Y$, the extended-valued function $\phi(\boldsymbol{x}, \cdot)$ is assumed to be a proper and lower semicontinuous convex function with $\phi(\boldsymbol{x}, 0) = 0$.

The Cauchy stress corresponding to a GSM is given by the relation

$$\boldsymbol{\sigma}(\boldsymbol{x}) = \frac{\partial w}{\partial \boldsymbol{\varepsilon}}(\boldsymbol{x}, \boldsymbol{\varepsilon}(\boldsymbol{x}), \underline{z}(\boldsymbol{x})), \tag{34}$$

whereas the evolution of the internal variables $\underline{z} : Y \to Z$ is governed by Biot's equation

$$-\frac{\partial w}{\partial \underline{z}}(\boldsymbol{x}, \boldsymbol{\varepsilon}(\boldsymbol{x}), \underline{z}(\boldsymbol{x})) \in \partial \phi(\boldsymbol{x}, \underline{\dot{z}}(\boldsymbol{x})). \tag{35}$$

In case the dissipation potential is finite-valued and differentiable, the latter subdifferential inclusion becomes an ordinary differential equation

$$\frac{\partial w}{\partial \underline{z}}(\boldsymbol{x}, \boldsymbol{\varepsilon}(\boldsymbol{x}), \underline{z}(\boldsymbol{x})) + \frac{\partial \phi}{\partial \underline{\dot{z}}}(\boldsymbol{x}, \underline{\dot{z}}(\boldsymbol{x})) = 0. \tag{36}$$

In practice, it is often more convenient to re-write Biot's equation (35) to involve the rate of the internal variables explicitly. This may be accomplished in the form

$$\underline{\dot{z}}(\boldsymbol{x}) \in \partial \phi^* \left(\boldsymbol{x}, -\frac{\partial w}{\partial \underline{z}}(\boldsymbol{\varepsilon}(\boldsymbol{x}), \underline{z}(\boldsymbol{x})) \right), \tag{37}$$

utilizing the driving-force potential defined via Legendre-Fenchel duality

$$\phi^*(\boldsymbol{x}, \underline{y}) \equiv \sup_{\underline{z}} \langle \underline{y}, \underline{z} \rangle - \phi(\boldsymbol{x}, \underline{z}), \quad \underline{y} \in Z^*. \tag{38}$$

If Biot's equation (37) is replaced by an implicit Euler discretization in time with time step $\triangle t$,

$$\frac{\underline{z}(\boldsymbol{x}) - \underline{z}^n(\boldsymbol{x})}{\triangle t} \in \partial \phi^* \left(\boldsymbol{x}, -\frac{\partial w}{\partial \underline{z}}(\boldsymbol{x}, \boldsymbol{\varepsilon}(\boldsymbol{x}), \underline{z}(\boldsymbol{x})) \right), \tag{39}$$

where we use the superscript n for the previous time step and consider variables without superscript belonging to the *current* time step, it was realized[24,25] that the time-discretized *primal* Biot's equation

$$-\frac{\partial w}{\partial \underline{z}}(\boldsymbol{x}, \boldsymbol{\varepsilon}(\boldsymbol{x}), \underline{z}(\boldsymbol{x})) \in \frac{\partial \phi}{\partial \underline{\dot{z}}} \left(x, \frac{\underline{z}(\boldsymbol{x}) - \underline{z}^n(\boldsymbol{x})}{\triangle t} \right) \tag{40}$$

may be written as the Euler–Lagrange equation of the variational principle

$$\int_Y w(\boldsymbol{x}, \boldsymbol{\varepsilon}, \underline{z}) + \triangle t \; \phi\left(\boldsymbol{x}, \frac{\underline{z} - \underline{z}^n}{\triangle t}\right) dx \to \min_{\underline{z}: Y \to Z}. \tag{41}$$

In particular, the associated stress operator $\mathbb{s}$ derives from the condensed potential w^*

$$\mathbb{s}(\boldsymbol{\varepsilon}) = \frac{\partial w^*}{\partial \boldsymbol{\varepsilon}}(\boldsymbol{\varepsilon}) \quad \text{with} \quad w^*(\boldsymbol{\varepsilon}) = \inf_{\underline{z}: Y \to Z} \int_Y w(\boldsymbol{x}, \boldsymbol{\varepsilon}, \underline{z}) + \triangle t \; \phi\left(\boldsymbol{x}, \frac{\underline{z} - \underline{z}^n}{\triangle t}\right) dx. \tag{42}$$

Suppose we are in the setup (18)

$$\mathbb{P} : \boldsymbol{\varepsilon} = \sum_\alpha \langle \chi_\alpha \boldsymbol{\varepsilon} \rangle_Y / \langle \chi_\alpha \rangle_Y \, \chi_\alpha. \tag{43}$$

Let us define the associated projector

$$\underline{\underline{P}} \in L(L^1(Y; Z)), \quad \underline{z} \mapsto \sum_\alpha \left\langle \chi_\alpha \underline{z} \right\rangle_Y / \langle \chi_\alpha \rangle_Y \, \chi_\alpha. \tag{44}$$

The operator $\underline{\underline{P}}$ is a projector, but it is not orthogonal because the latter notion does not make sense on a Banach space. With the projector at hand, and using the projector (18) as well as the stabilization stiffness (21), we consider the condensed potential

$$\tilde{w}^*(\boldsymbol{\varepsilon}) = \inf_{\underline{z}: Y \to Z, \; \underline{\underline{P}} \underline{z} = \underline{z} \text{ a.e.}} \int_Y w(\boldsymbol{x}, \mathbb{P} : \boldsymbol{\varepsilon}, \underline{z}) + \frac{1}{2} \, \boldsymbol{\varepsilon} : \mathbb{Q} : \mathbb{C}^{\text{stab}} : \mathbb{Q} : \boldsymbol{\varepsilon} + \triangle t \; \phi\left(\boldsymbol{x}, \frac{\underline{z}(\boldsymbol{x}) - \underline{z}^n(\boldsymbol{x})}{\triangle t}\right) dx \tag{45}$$

and the associated stress operator

$$\tilde{\mathbb{s}}(\boldsymbol{\varepsilon}) \equiv \frac{\partial \tilde{w}^*}{\partial \boldsymbol{\varepsilon}}(\boldsymbol{\varepsilon}). \tag{46}$$

More explicitly, we obtain the representation

$$\tilde{\mathbb{s}}(\boldsymbol{\varepsilon}) = \frac{\partial w}{\partial \boldsymbol{\varepsilon}}(\cdot, \mathbb{P} : \boldsymbol{\varepsilon}, \underline{z}) + \mathbb{C}^{\text{stab}} : \mathbb{Q} : \boldsymbol{\varepsilon}, \tag{47}$$

where the internal-variable field $\underline{z} : Y \to Z$ solves the following variant of Biot's equation

$$-\frac{\partial w}{\partial \underline{z}}(\cdot, \mathbb{P} : \boldsymbol{\varepsilon}, \underline{z}) \in \frac{\partial \phi}{\partial \dot{\underline{z}}}\left(\cdot, \frac{\underline{z} - \underline{z}^n}{\triangle t}\right) \tag{48}$$

with $\underline{\underline{P}} \, \underline{z}(\boldsymbol{x}) = \underline{z}(\boldsymbol{x})$ almost everywhere. In the previous derivations, we implicitly used the assumption that the free energy density w and the dissipation potential ϕ are homogeneous on each cluster χ_α.

Usually the dual version of Biot's equation (37)

$$\frac{\underline{z} - \underline{z}^n}{\triangle t} \in \frac{\partial \phi^*}{\partial \dot{\underline{z}}}\left(\cdot, -\frac{\partial w}{\partial \underline{z}}(\cdot, \mathbb{P} : \boldsymbol{\varepsilon}, \underline{z})\right) \tag{49}$$

is computationally more convenient. These developments may be stated informally as follows: the internal variables are assumed to be homogeneous on each cluster χ_α. Their evolution is driven by the cluster-wise average of the strains. The resulting stresses (47) involve the stresses associated to the GSM on this cluster plus the elastic stresses associated to the stabilization stiffness and the strain fluctuations of the cluster. From this perspective, the proposed technology generalizes Dvorak's transformation-field analysis (TFA).[15,16,26] The latter describes the evolution of cluster-wise averages of internal variables, but does not account for stabilization terms. Moreover, we do not restrict to Hooke-type constitutive laws which is typical for TFA.

Last but not least let us remark that the presented framework naturally allows for a refinement of the clusters, that is, the condition

$$\underline{\underline{P}}\,\underline{\underline{P}}^n = \underline{\underline{P}} \tag{50}$$

must hold, but does not naturally support coarsening, as heterogeneous internal variables within a cluster also necessitate heterogeneous driving forces. In fact, when refining a cluster, the previous value of the internal variable is copied to all subclusters.

As a very last remark, let us stress that the refinement-only strategy (50) in the setup (18) together with the stabilization choice (31) implies that the stabilization stiffness $\mathbb{C}^{\mathtt{stab}}$ has no influence on the average stress, that is, the identity

$$\langle \tilde{\mathbb{S}}(\boldsymbol{\varepsilon})\rangle_Y = \langle \mathbb{S}(\mathbb{P} : \boldsymbol{\varepsilon})\rangle_Y \tag{51}$$

holds. This simple observation may be used to save a few computations.

3 | OCTREE-SANDWICH COARSENING AND COMPUTATIONAL ASPECTS

3.1 | Octree coarsening with sandwich extension

To construct the supercells required in the coarsening strategy described in Section 2.2, we rely on octrees.[14] An octree is a tree structure where each internal node has exactly eight children. As usual in computer graphics, we consider each node of the octree to correspond to a (possibly large) voxel and its children to represent a nonoverlapping decomposition of the large voxel into eight congruent smaller voxels. In case the octree is not a fully populated tree, considerable savings in memory occupancy can be achieved when storing binary voxel data.

For the application at hand, we consider an octree representation of the microstructure geometry. This representation can change from one load step to the next. Following the principles established in Section 2.3, however, the octree representation is only allowed to be refined in this process. In fact, coarsening an octree would require to coarsen the internal variables, as well, and such a procedure may cause some mayhem.

The construction of our octree follows three simple principles:

1. The octree should resolve the material heterogeneity.
2. Linear elastic phases are always fully resolved.
3. The octree resolves the heterogeneity of the strain field.

The first principle means that a necessary condition for combining eight voxels into a supercell is that all eight voxels have the same material behavior. To ensure that principle 3 is satisfied, we investigate the local strain field and determine whether the heterogeneity of the strain field is resolved by the current octree. For this purpose, we fix a tolerance level $\mathtt{tol} \in (0, 1)$ and investigate each nodal octree cell $\mathtt{C}$ in the octree $\mathcal{T}$. Such a cell needs to be refined if there is a subvoxel $\mathtt{V}$, s.t. some strain component $\varepsilon_{ij}(\mathtt{V})$ differs from the ij-component of the nodal-cell average $\langle\boldsymbol{\varepsilon}\rangle_{\mathtt{C}}$ by more than a relative tolerance $\mathtt{tol}$, that is, if the inequality

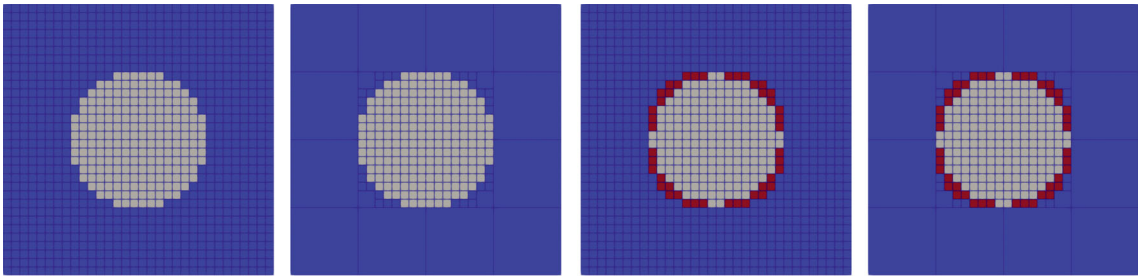
$$\left|\varepsilon_{ij}(\mathtt{V}) - \langle\varepsilon_{ij}\rangle_{\mathtt{C}}\right| > \mathtt{tol}\ \|\langle\boldsymbol{\varepsilon}\rangle_{\mathtt{C}}\|_{\ell^\infty} \tag{52}$$

holds with the maximum norm

$$\|\langle\boldsymbol{\varepsilon}\rangle_{\mathtt{C}}\|_{\ell^\infty} = \max_{k,l=1}^{3} |\langle\varepsilon_{kl}\rangle_{\mathtt{C}}|. \tag{53}$$

This rule implies, in particular, that composite voxels[27] cannot be coarsened. In fact, composite voxels are typically implemented as a separate material law, involving volume fraction and normal data per voxel.

At the beginning of the simulation, we construct a coarsened octree representation of the microstructure based on these geometric considerations alone. The results are illustrated in Figure 1, which shows a cross section of a single



(A) No composite voxels (B) With composite voxels

FIGURE 1 Setup of the initial octree for a single circular inclusion with and without composite voxels. The composite voxels were obtained by downsampling from a higher resolution.

spherical inclusion (compare Section 4.2 below). Without composite voxels, there are two phases, a blue and a white phase. The nonlinear matrix phase (in blue) is coarsened, the elastic inclusion (gray) remains fully resolved as a consequence of principle 2 above. In fact, principle 2 is actually a consequence of performance considerations, as the elastic stabilization procedure (23) would only infer computational overhead for coarsened cells in the elastic phase.

Figure 1 also shows the resulting octree in case composite voxels are used. In this case, there is a third phase (in red), corresponding to the composite voxels. In this case, there is less potential for coarsening due to the additional constraints imposed by the presence of the composite voxels. In Section 4, however, it will become clear that activating composite voxels may still be a good idea, as they typically lead to more accurate solution fields which moreover turn out to be smoother, permitting to retain a coarser octree due to principle 3 above.

To get an idea about the workings of the procedure, we take a look at Figure 2, which shows the evolution of the octree in the center column. For the details of the simulation, we refer to Section 4.2. After convergence of the solution algorithm and before moving to the next time step, the octree is refined iteratively until a desired accuracy is achieved. Some tinkering revealed that $\texttt{tol} = 10\%$ is a reasonable tolerance for the refinement criterion (52), and we will use this value for the entire manuscript.

Figure 2 shows that the octree is significantly refined after the first load step. Please note that the figure shows the cell averages on the octree cells only. Of course, due to the elastic stabilization term (23), the strain field is also heterogeneous within such an octree cell. However, for the sake of visualization, we subtracted these fluctuations. In particular, if the first step is elastic, the octree coarsening does not infer an error compared to a computation at full voxel resolution.

At least visually, we see little difference between the fully resolved strain field and the octree-coarsened version, building some confidence in the presented octree-coarsening strategy. In addition to these qualitative considerations, precise quantitative evaluations are available, and we refer to Section 4 for details.

Before discussing the implementation, we will discuss a simple modification of octree coarsening that turns out to be critical for industrial applications. For a given cell with eight subvoxels, an octree gives you only two options: either the subvoxels are merged or the full count of eight subvoxels is considered. In practical applications, such a harsh strategy cannot exploit situations where some (but not all) subvoxels share a similar value. Such an event occurs quite frequently at a material interface, where not only the heterogeneity decomposes into two categories, but also the strain values are similar at the two sides of the interface. This observation motivated Linden et al.[28] to introduce left-intermediate-right (LIR) trees, which consider a whole variety of possible decompositions of a cubic agglomeration of eight voxels. To avoid some of the technicalities, we consider a specific subset of such an LIR tree. We investigate the case that the cell is cut in half along one of the coordinate axes. We call this strategy sandwich coarsening due to the apparent resemblance with the food.

We integrate sandwich coarsening as an elementary add-on to sandwich coarsening by augmenting the coarsening check (52) by similar queries along each of the coordinate axes.

Such sandwich-coarsened cells are shown in Figure 2 in the right column. Due to the spherical geometry, no difference to the octree case is evident for the initial tree. However, after the first time step, the differences become quite apparent. Interestingly, this simple trick permits to save a factor of four in the number of cells (and, thus, nonlinear material evaluations) in case it is applicable.

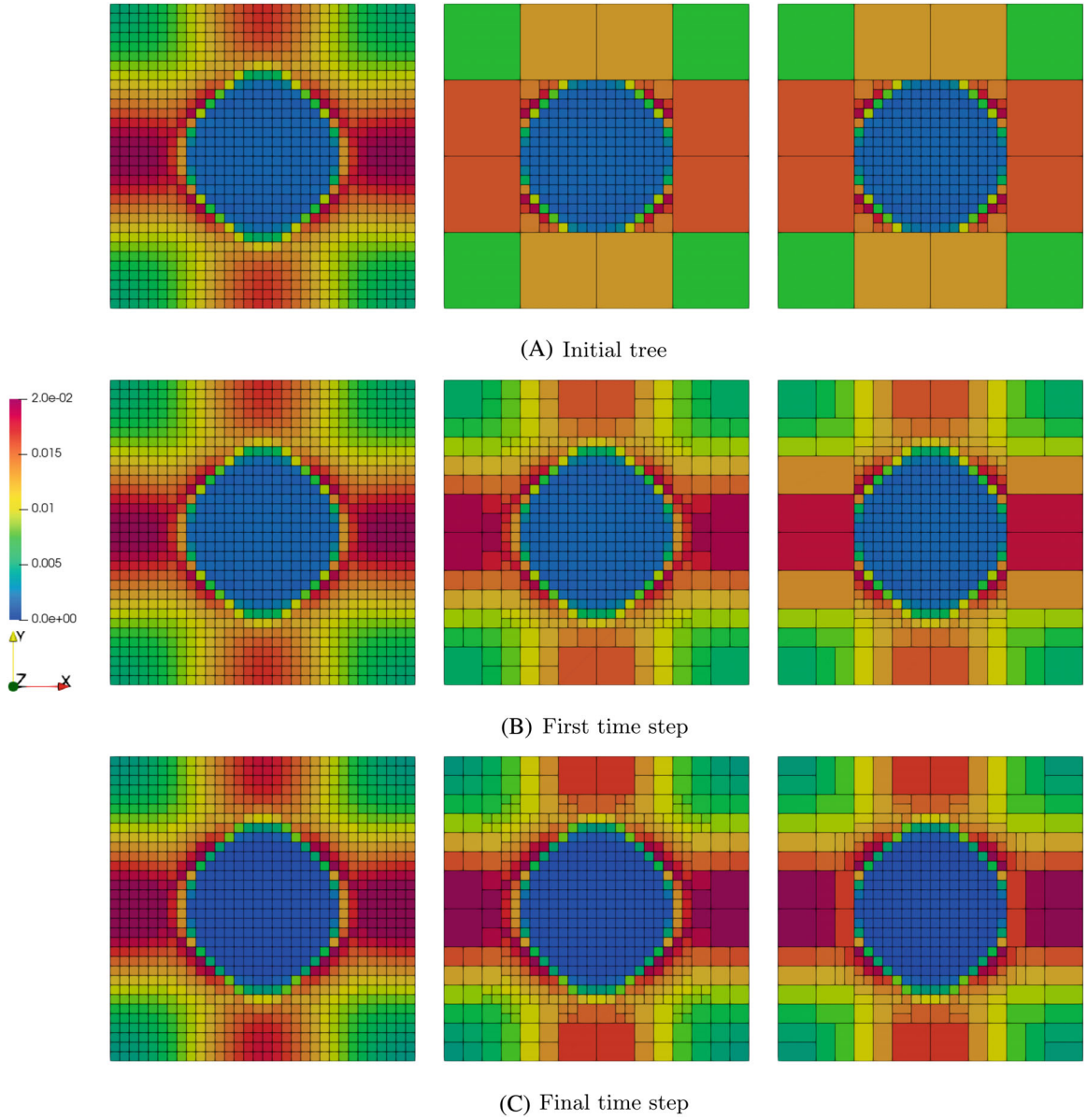


FIGURE 2 Mises norm of the local strain field for original voxel mesh (left) with octree (middle) and sandwich coarsening (right) when using laminate composite voxels.

3.2 | Implementation

Of central importance to the software implementation is the employed data structure. Classical FFT-based computational micromechanics software is based on voxel grids which are stored as (three-dimensional) arrays. Most solution algorithms[5(sections 3.2–3.4)] are extensions of the basic scheme

$$\boldsymbol{\tau}^k = \mathbb{s}(\boldsymbol{\varepsilon}^k) - \mathbb{C}^0 : \boldsymbol{\varepsilon}^k, \tag{54}$$

$$\boldsymbol{\varepsilon}^{k+1} = \overline{\boldsymbol{\varepsilon}} - \Gamma^0 : \boldsymbol{\tau}^k, \tag{55}$$

introduced by Moulinec–Suquet.[1,2] The two essential operations are the computation of the stress polarization (54) and the application (55) of the Eshelby–Green operator $\Gamma^0 = \nabla^s(\mathrm{div}\ \mathbb{C}^0\nabla^s)^{\dagger}\mathrm{div}$ for a given reference material $\mathbb{C}^0$. Both operations are based on loops over arrays, whereas the fast Fourier transform and its inverse are used to convert from real to Fourier space and back, exploiting the locality of the action of the Eshelby–Green operator in real space. In fact, the line

(54) involves a loop over all voxels in real space, and in the line (55), a loop over the relevant frequencies in Fourier space (which can also be brought into the form of a rectangular array) is executed.

The basic data structure which we use for the implementation of the octree/sandwich coarsening is a linear octree.[19] Instead of storing a tree structure, linear octrees only store a unique identifier of each leaf in the tree in an array. Then, storing data in the octree is as simple as storing data in an array, and the unique identifier is used to connect the data in the array to the spatial location the data belongs to. For the implementation at hand, we rely upon Morton encoding.[29] The underlying idea is the following. The eight subcells of an octree node can be encoded by three bits. Thus, the path from the root to the leaf of an octree may be encoded by a sequence of such bits in a unique fashion. Queries for parent and children nodes admit efficient implementation based on proper bit-wise manipulations,[30] available in programming languages like `C++`.

So, after generating the initial octree, we store its topology in a `C++`-vector containing the corresponding Morton codes. In case the octree needs to be refined, we change the Morton code of the parent cell to the corresponding code of its first child, and append the remaining seven children to the end of the vector. In addition, we store the material color for each octree leaf and allocate memory for the internal variables in the process. In case of refinement, the corresponding `C++`-vector is reshaped, copying the old values of the internal variables to all eight children.

We deal with sandwich coarsening by adding a further flag indicating the direction of splitting, so that voxels and the upper/lower parts of the sandwich may be distinguished. The handling of this extension is straightforward.

Alg. 1 provides a pseudo-code for the implementation of the coarsened basic scheme. The basic scheme is only modified when computing the polarization (54). For every leaf of the octree or sandwich cell, the cell-wise average strain is computed first. Then, the stress response is calculated for this cell, updating the internal variables if needed. Subsequently, the calculated stress is prolongated to all subvoxels, adding the coarsening stabilization (23) in the process. Please note that we compute the tangent of the previous load step on the fly, as we store the previous strain field as part of the internal variables. In particular, no extra storage is needed for this procedure.

Applying the Eshelby–Green operator (55) is not modified.

Algorithm 1. Basic scheme for coarsened discretization (multiple load steps)

$\mathcal{T} \leftarrow$ `BuildOctree`[VoxelGeometry] ▷ Build octree coarsening of voxel mesh in nonlinear phases
$\boldsymbol{\varepsilon} \leftarrow 0$ ▷ Initialization of $\boldsymbol{\varepsilon}$
for $n \in \{1, \dots, n_{\max}\}$ **do** ▷ Load-step control
  repeat
    $\boldsymbol{\varepsilon} \leftarrow$ `MSiterate`$[\boldsymbol{\varepsilon}, \mathbb{C}_n^0, \overline{\boldsymbol{\varepsilon}}_n, \mathcal{T}]$ ▷ Reference material $\mathbb{C}^0$ may be adjusted
  until Convergence
  for $c \in \{1, \dots, c_{\max}\}$ **do**
    $\mathcal{T} \leftarrow$ `RefineOctree`$[\mathcal{T}, \boldsymbol{\varepsilon}]$
  end for
end for
return $\boldsymbol{\varepsilon}$

`MSiterate`$[\boldsymbol{\varepsilon}, \mathbb{C}^0, \overline{\boldsymbol{\varepsilon}}, \mathcal{T}]$

for $\mathtt{C} \in \mathcal{T}$ **do** ▷ Loop over cells
  $\boldsymbol{\varepsilon}_{\mathtt{C}} \leftarrow \langle \boldsymbol{\varepsilon} \rangle_{\mathtt{C}}$ ▷ Computing average strain $\boldsymbol{\varepsilon}_{\mathtt{C}}$ on the cell $\mathtt{C}$
  $\boldsymbol{\sigma}_{\mathtt{C}} \leftarrow \mathbb{s}(\boldsymbol{\varepsilon}_{\mathtt{C}}) - \mathbb{C}^0 : \boldsymbol{\varepsilon}_{\mathtt{C}}$ ▷ Computing the cell-wise stress $\boldsymbol{\sigma}_{\mathtt{C}}$ and update of internal variables
  for $\mathtt{V} \in \mathtt{C}$ **do** ▷ Loop over subvoxels
    $\boldsymbol{\varepsilon}(\mathtt{V}) \leftarrow \boldsymbol{\sigma}_{\mathtt{C}} + \mathbb{C}^{\mathtt{stab}} : (\boldsymbol{\varepsilon}(\mathtt{V}) - \boldsymbol{\varepsilon}_{\mathtt{C}}) - \mathbb{C}^0 : \boldsymbol{\varepsilon}(\mathtt{V})$ ▷ Prolongation and polarization computation
  end for
end for
$\boldsymbol{\varepsilon} \leftarrow \mathrm{FFT}(\boldsymbol{\varepsilon})$ ▷ Fast Fourier transform
$\boldsymbol{\varepsilon}(\boldsymbol{\xi}) \leftarrow -\hat{\Gamma}^0(\boldsymbol{\xi}) : \boldsymbol{\varepsilon}(\boldsymbol{\xi}), \boldsymbol{\xi} \neq 0$ ▷ Application of Γ^0 in Fourier space
$\boldsymbol{\varepsilon}(0) \leftarrow \overline{\boldsymbol{\varepsilon}}$ ▷ Setting the mean strain in Fourier space
$\boldsymbol{\varepsilon} \leftarrow \mathrm{FFT}^{-1}(\boldsymbol{\varepsilon})$ ▷ Inverse Fast Fourier transform
return $\boldsymbol{\varepsilon}$

When dealing with multiple load steps, we proceed as follows. An initial octree/sandwich structure is set up in preprocessing based on geometric considerations. For each load step, the solution algorithm is used to find the solution of the cell problem (13) on a fixed tree. After convergence, the refinement criterion (52) is checked, triggering a tree refinement if necessary. Then sandwich cells are split into their four subvoxels and octree cells are split into two sandwich cells. Here the sandwich direction is chosen in such a way that the error reduction is maximal. As it may happen that the tree needs to be refined by adding more than one level, we actually repeat the refinement check a couple of times, up to the specified maximum coarsening depth $c_{\max}$. Keep in mind that this refinement procedure is typically negligible in runtime compared to the application of the solution algorithms.

We discuss the integration of the proposed coarsening strategy into the basic scheme only, as the extension to solvers which augment the basic scheme like fast gradient methods[31-33] and (Quasi-)Newton-type schemes[34-36] is straightforward. For the article at hand, we rely upon the nonlinear conjugate gradient method[37] where we limit the factor β in line 15 of Alg. 1 in Schneider[37] by one to increase robustness of the algorithm.

4 | COMPUTATIONAL INVESTIGATIONS

4.1 | Used materials, models and parameters

For our computational investigations, we simulate creep and tensile tests on volume elements comprising a polypropylene (PP) matrix reinforced with E-glass inclusions. For both cases, the E-glass inclusions are assumed to show an isotropic linear elastic material response which is specified by the material parameters shown in Table 1. For the tensile tests, the polypropylene matrix is treated as an elasto-viscoplastic material. The considered GSM involves the viscoplastic strain tensor $\boldsymbol{\varepsilon}_{\mathrm{vp}}$ and an isotropic hardening variable α as the internal variables, the Helmholtz free energy density

$$w(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}_{\mathrm{vp}}, \alpha) = \frac{1}{2}\left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\mathrm{vp}}\right) : \mathbb{C}^{\mathrm{e}} : \left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\mathrm{vp}}\right) + \int_0^{\alpha} \mathcal{K}(q)\, \mathrm{d}q \tag{56}$$

with an isotropic linear elastic stiffness tensor C^{e} and the force potential

$$\phi^*(\boldsymbol{A}_{\mathrm{vp}}, A_\alpha) = \frac{\sigma_d\ \dot{\varepsilon}_0}{m+1} \left\langle \frac{\left\| \mathrm{dev} \boldsymbol{A}_{\mathrm{vp}} \right\|_{\mathrm{eq}} + A_\alpha}{\sigma_d} \right\rangle_+^{m+1}, \tag{57}$$

where dev denotes the deviatoric part of a tensor, $\|\cdot\|_{\mathrm{eq}}$ refers to the Mises norm, $\langle\cdot\rangle_+$ stands for the Mcauley bracket. The viscous effects are described by the drag stress σ_d, the rate-sensitivity m exponent and the reference strain rate $\dot{\varepsilon}_0$. The stress tensor and the driving forces compute as

$$\boldsymbol{\sigma} = \mathbb{C}^{\mathrm{e}} : \left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\mathrm{vp}}\right), \quad \boldsymbol{A}_{\mathrm{vp}} = \boldsymbol{\sigma} \quad \text{and} \quad A_\alpha = -\mathcal{K}(\alpha). \tag{58}$$

We consider a linear-exponential hardening function

$$\mathcal{K}(\alpha) = \sigma_Y + a_1(1 - \exp(-a_2\alpha)). \tag{59}$$

For the studied example, the nonlinear parameters of the PP matrix are taken from Grimm-Strele et al.[38] and are reproduced in Table 2.

For the creep tests, the PP matrix is modeled as a linear viscoelastic solid, more precisely a generalized Maxwell model, which may be formulated as a GSM.[39] The corresponding linear viscoelastic material parameters of the Polypropylene

TABLE 1 Isotropic linear elastic material parameters of E-glass.

Parameter	Unit	Value
Young's modulus (E)	GPa	72
Poissons' ratio (ν)	-	0.22

TABLE 2 Material parameters for the viscoplastic material model of PP.[38]

Behavior	Parameters
Elastic	$E = 1.058$ GPa, $\nu = 0.45$
Hardening	$\sigma_Y = 3.982$ MPa, $a_1 = 14.01$ MPa, $a_2 = 220.578$
Viscoplastic	$\dot{\varepsilon}_0$=1/s, $\sigma_d = 9.682$ MPa, $m = 5.835$

TABLE 3 Viscoelastic material parameters[42] for polypropylene (PP): Relaxed Young's modulus $E_r = 320$MPa, Poisson's ratio $\nu = 0.44$ as well as normalized relaxation coefficients γ_j and relaxation times τ_j for each Maxwell element.

#	1	2	3	4	5	6
γ_j	0.190	0.100	0.168	0.167	0.205	0.245
τ_j in s	0.116×10^{-8}	0.574×10^{-8}	0.285×10^{-7}	0.141×10^{-6}	0.702×10^{-6}	0.348×10^{-5}
#	7	8	9	10	11	12
γ_j	0.492	0.587	0.725	0.885	0.485	0.726
τ_j in s	0.173×10^{-4}	0.857×10^{-4}	0.425×10^{-3}	0.211×10^{-2}	0.105×10^{-1}	0.520×10^{-1}
#	13	14	15	16	17	18
γ_j	0.414	0.468	0.370	0.316	0.255	0.207
τ_j in s	0.258×10^{0}	0.128×10^{1}	0.635×10^{1}	0.315×10^{2}	0.156×10^{3}	0.775×10^{3}
#	19	20	21	22	23	24
γ_j	0.159	0.133	0.087	0.112	0.004	0.168
τ_j in s	0.385×10^{4}	0.191×10^{5}	0.947×10^{5}	0.470×10^{6}	0.233×10^{7}	0.116×10^{8}

were obtained by approximating a measured master curve[40] with 24 Maxwell elements by using the Levenberg–Marquardt algorithm. The resulting parameters are summarized in Table 3.

We use the direct implementation of Kaliske et al.[41] instead of the automatic-differentiation technique of Blühdorn et al.[43] as the latter leads to a higher computational effort on CPUs.

The algorithms described in this article were integrated into the software FeelMath and run in a modified version of the software GeoDict2023.[44] The plan of our studies is the following. We consider two creep and one tensile experiment. The first creep test is performed on a PP reinforced by a single spherical E-glass inclusion to investigate an essentially optimal scenario for our coarsening approach—with 6.5%, the volume fraction of the inclusion is chosen to be comparatively small, and significant coarsening is possible. Subsequently, we will consider a creep test for a cross-ply microstructure, which features a significantly higher filler fraction with 30%. For the final tensile test, we will investigate the microstructure of a short-fiber reinforced PP at an industrial volume fraction. This microstructure leads to strain peaks at the fiber ends and appears to represent a challenge for the introduced coarsening strategy. All these simulations are performed using the Willot's rotated staggered grid,[45] since it has symmetric solution fields in contrast to the staggered grid discretization.[46]

4.2 | A single spherical inclusion

Similar to the work of Kaiser et al.,[9] we start our computational investigations on a microstructure with a single spherical inclusion. For a creep test subjected to 1% tensile loading with 50 load steps equidistantly spaced on a logarithmic scale, we investigate the accuracy and computational performance of the sandwich coarsening presented in Section 3. To be more precise, we compare six different scenarios in the Figures 4 as well as 5 and the remainder of this section.

The first and simplest case considers a voxelization (V) of the analytically described spherical inclusion (solid lines) and serves as the reference. This approach is state-of-the-art for many FFT-based solvers. The results for a resolution of 256^3 voxels serve as the reference, and four coarser resolutions, 16^3, 32^3, 64^3, and 128^3 voxels, are considered, see Figure 3.

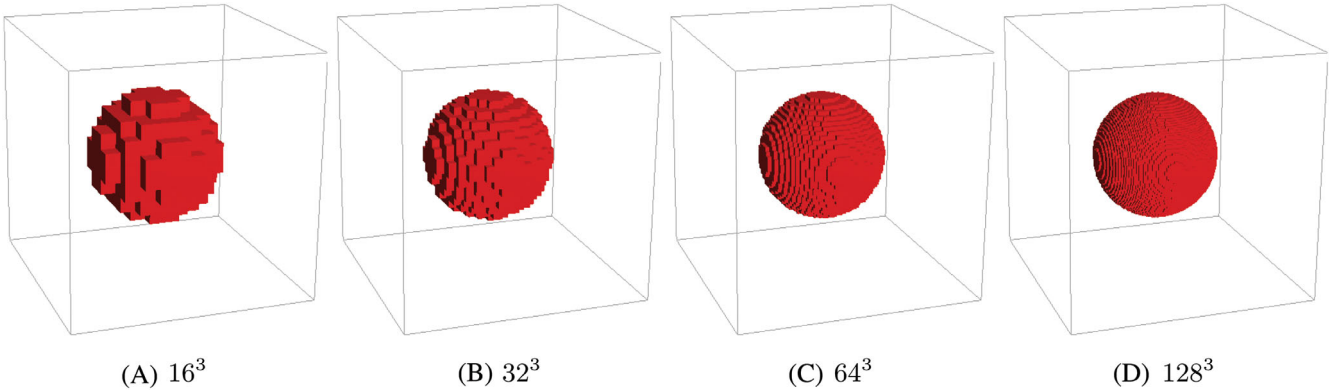


FIGURE 3 Microstructure of the spherical inclusion at different resolutions.

The second approach involves the proposed sandwich-coarsening strategy applied to the voxelized structures (V-Sandwich) at different resolutions (dashed lines). Depending on the evolution of the local strain field, the number of cells increases, possibly up to a fully resolved voxel microstructure.

Our third variant concerns using laminate composite voxels (CV)[27] (dotted lines) in interface elements. This strategy represents the current state-of-the-art of FFT-based solvers provided a subvoxel-scale representation of the material interface is available. To estimate the volume fractions and the normals for the laminate composite voxels, we rely on highest considered resolution, 256^3 voxels, and utilize the algorithm described in Kabel et al.[47] with downsampling factors 16, 8, 4, and 2.

In addition, we combine the laminate composite voxels with the sandwich coarsening (CV-Sandwich) as our fourth variant (dash-dotted lines). In this case, the potential decrease in the number of cells is slightly lower than for the pure sandwich coarsening because the laminate composite voxels at the material interface cannot be coarsened into sandwich cells. In fact, the interface voxels constitute a third material, and coarsening is only allowed to take place in homogeneous regions.

Finally, for the fifth variant, we use composite boxels (CB)[48] (densely dotted lines) and combine it with the sandwich coarsening (CB-Sandwich) (densely dash-dotted lines), giving rise to the sixth variant considered. Composite boxels were introduced to extend the composite-voxel method to regular meshes with general cuboid elements ("boxels"). In fact, the composite-voxel method employing laminate mixing requires the volume fractions of the constituens within the element, together with an estimate for the normal to the interphase separating the material. Whereas calculating volume fractions on a regular mesh is straightforward, finding a suitable approximation to the normal is not that trivial. The original work[47] proposed to consider the direction of the vector connecting the centroid of the dominating phase of the element to the centroid of the element. This strategy, however, did not work well for boxels with significant anisotropy. Therefore, Keshav et al.[48] proposed to smooth the characteristic function of the dominant phase and to compute the normal by averaging the finite-difference gradient of this smoothed characteristic function. This technique of image processing type also turned out to be beneficial for cube-shaped elements compared to the traditional normal-estimation technique.[47] This is the reason for including the technique in this work.

The effect of sandwich coarsening for this example is shown in Figure 2.

Before investigating the potential gain in computational speed and reduction the memory footprint, we first assess the additional error inferred by working on a coarser resolution and by employing both composite voxels/boxels and sandwich coarsening. The computation on the full resolution of 256^3 voxels serves as our reference.

For the considered scenarios, the computed effective stress in loading direction is shown in Figure 4. We observe that the initial stress relaxes over time, reaching an eventual steady state at slightly less than 5 MPa. As the different approaches turn out to be rather close and inspecting their difference for the absolute stress value appears difficult, we investigate the relative error in the axial stress over time, also shown in Figure 4.

As a general trend, we observe that the error in the effective stress roughly reduces by a factor of two when doubling the resolution of a single axis of the cell, conforming to the theoretical predictions.[49,50] Even at the coarsest considered resolution, the relative stress error does not exceed 3%.

Let us take a look at the coarsest resolution with 16^3 voxels in more detail. We observe that using composite voxels/boxels decreases the stress error by about a factor three in the initial stage and by about 25% in the later stages of the

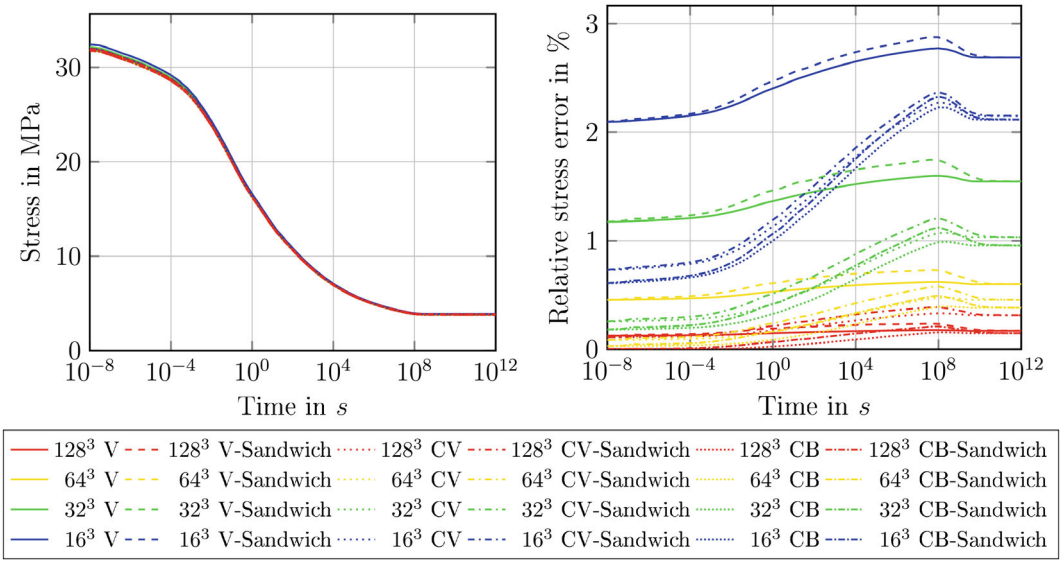


FIGURE 4 Stress-strain curves and relative error for the spherical inclusion. The considered coarsening approaches are voxelization (V) at lower resolution, sandwich-coarsening applied to the voxelized structures (V-Sandwich), voxelization at lower resolution with laminate composite voxels (CV) at the material interface, combination of the laminate composite voxels with sandwich coarsening (CV-Sandwich), voxelization at lower resolution with composite boxels (CB) at the material interface and a combination of composite boxels with sandwich coarsening (CB-Sandwich).

loading. For single spherical inclusion, there is actually little difference between composite voxels and boxels, as the estimation of the interface normals profits from the comparatively high resolution. Activating sandwich coarsening increases the stress error only mildly, both without and with composite voxels/boxels. In fact, there is little deviation observed in the initial and final stages of the loading due to the choice of elastic stabilization term (31). The smallness of the inferred error implies that, for the case at hand, the benefits of using sandwich coarsening may be exploited without compromising the accuracy.

The observations made for the coarsest resolution and using composite voxels/boxels as well as sandwich coarsening carry over to the two next highest resolutions with little change. The only difference is that the overall error level is lower. However, at the finest considered resolution of 128^3 voxels, the positive influence of using composite voxels/boxels is less palpable. The reason is found in the inaccurate normal and volume-fraction data. In fact, a resolution of 256^3 is used to estimate these critical quantities for the composite voxels and only eight subvoxels are available for 128^3 voxels, limiting the realized volume fractions and normals considerably.[47]

The potential gain with respect to memory usage and runtime will be investigated in the following.

Figure 5 shows the number of cells and the used memory as a function of time for the considered creep loading of the single spherical matrix inside the viscoelastic matrix. Let us first take a look at the number of cells for the coarsest resolution. When using sandwich coarsening, the number of cells increases quickly in the initial state of the loading, but stabilizes eventually. Compared to the regular grid, using the sandwich coarsening reduces the number of cells by about 15%. If sandwich coarsening is applied to geometries with composite voxels/boxels, the number of cells will be reduced compared to a sandwich-coarsened pure-voxel microstructure. This result appears counter-intuitive at first, as the composite voxels at the interface preclude a coarsening at these sites. However, the additional factor at play is the increased smoothness of the local strain field resulting from the laminate composite voxels. In fact, avoiding to use composite voxels results in a stronger heterogeneity of the strain field, in particular at material interfaces. This increased heterogeneity precludes a coarsening from taking place. Please note that using composite voxels without coarsening requires the same cell count as a fully resolved voxel grid, that is, the corresponding curves are overlaid in the plot at hand.

For the next highest resolution, 32^3 voxels, the observations are qualitatively similar. However, the number of cells is reduced by slightly more than 50% when activating the sandwich coarsening. For both 64^3 and 128^3 voxels, the cell count may be reduced by about 90% when using sandwich coarsening. In particular, we observe a tremendous potential to reduce the cell count for these fine resolutions.

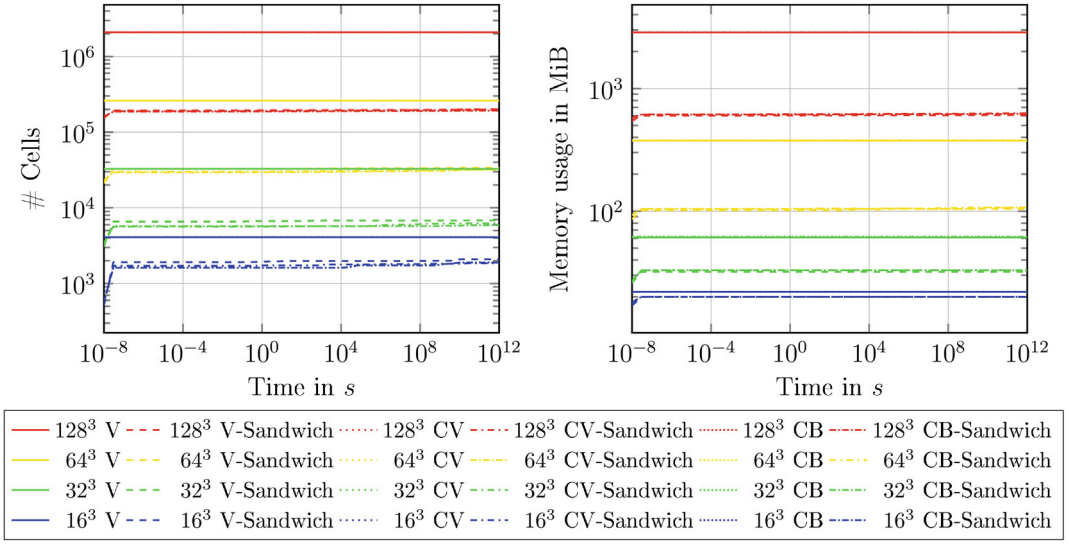


FIGURE 5 Memory usage and number of cells for spherical inclusion. The considered coarsening approaches are voxelization (V) at lower resolution, sandwich-coarsening applied to the voxelized structures (V-Sandwich), voxelization at lower resolution with laminate composite voxels (CV) at the material interface, combination of the laminate composite voxels with sandwich coarsening (CV-Sandwich), voxelization at lower resolution with composite boxels (CB) at the material interface and a combination of composite boxels with sandwich coarsening (CB-Sandwich).

Aside from considering the cell count, we also take a look at the consumed memory, see Figure 5. Using composite voxels/boxels increases the required storage. On the one hand, geometrical information, that is, the volume fraction and the estimated normal, need to be stored for each composite voxel. Moreover, it is computationally convenient to store the laminate-jump vector ($\boldsymbol{a}$ in equation (2.13) of Kabel et al.[27]) in each composite voxel, that is, an additional vector internal variable needs to be stored. For the material model at hand, a generalized Maxwell model with 24 elements, the increase in memory consumption induced by using composite voxels is negligible, however. Thus, it is hard to discern two curved corresponding to using composite voxels or not in the corresponding plot. This is seen, for instance at the coarsest resolution. Sandwich coarsening leads to a reduction of the consumed memory by 75% for this resolution.

Higher resolutions provide similar results with the exception that combining composite boxels with sandwich coarsening reduces the memory consumption compared to the combined composite-voxels & sandwich coarsening strategy. The reason appears to be hidden in the improved-accuracy calculation of the normal enabled by composite boxels. More than 75% of memory can be saved when using sandwich coarsening, and the best results are realized in combination with composite boxels.

In addition to the memory consumption, we also investigate the convergence speed and the elapsed runtime of the different approaches. In fact, in FFT-computational micromechanics, it is not uncommon that a faster solver may be designed at the expense of additional memory only. We investigate the iteration count per load step in Figure 6 for nonlinear CG[37] with β-limitation. We observe a peak at the first iteration, and a reduced iteration count at about twenty for the subsequent time steps which is a result of the extrapolated initial guess for the subsequent time step. For longer times, the iteration counts increase slightly, reaching a peak close to the end of the loading. Close to the end of the considered time window, the iteration counts drop significantly.

First and foremost, we observe that the iteration counts remain moderate for the entire considered time window. In fact, no more than eighty iterations are required at any time step. This feature is a consequence of the employed stabilization term (23). Of course, there are differences in the individual iteration counts, both caused by different resolutions, whether composite voxels/boxels are used or not and on the activation status of the sandwich coarsening. Clearly, using a fully voxelized structure tends to lead to lower iteration counts, in general. However, the sandwich coarsening strategy does not increase the iteration count beyond what is observed for enriching interfaces with composite voxels. The reason for this phenomenon is the following. Apart from the convergence rate of the solver, the number of iterations is also influenced by the starting point of the iteration, more precisely by the distance of the starting point of the iteration to the solution. It is well-known from computational practice that using composite voxels/boxels typically infers higher iteration counts than just using a regular mesh without interface voxels. Using composite voxels/boxels or sandwich coarsening

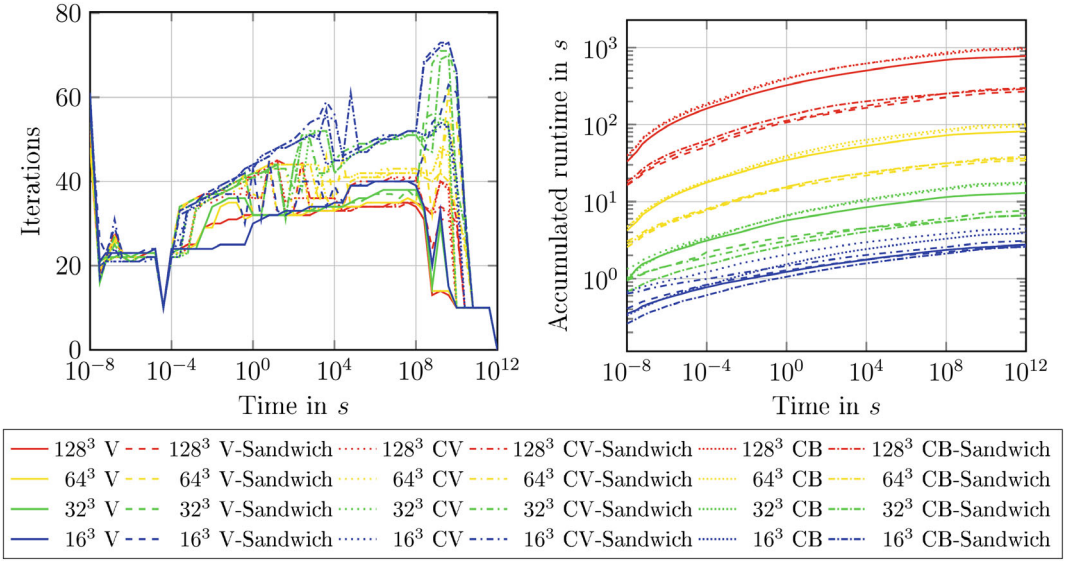


FIGURE 6 Number of iterations for convergence and accumulated runtime for the spherical inclusion. The considered coarsening approaches are voxelization (V) at lower resolution, sandwich-coarsening applied to the voxelized structures (V-Sandwich), voxelization at lower resolution with laminate composite voxels (CV) at the material interface, combination of the laminate composite voxels with sandwich coarsening (CV-Sandwich), voxelization at lower resolution with composite boxels (CB) at the material interface and a combination of composite boxels with sandwich coarsening (CB-Sandwich).

leads to a solution with a higher degree of complexity (due to the presence of the interface voxels), and we surmise that the starting point of the iteration is typically further away from such a solution than from the solution without interface voxels.

Taking a look at the accumulated runtime in Figure 6, we observe that using composite voxels increases the runtime due to the more complex material evaluation required in the composite voxels. As the number of interface voxels is not too large, the increase of using composite voxels/boxels is not too strong. In contrast, the runtime is significantly reduced, by about a factor of four, whenever the sandwich coarsening is activated. Interestingly, there is little influence of whether composite voxels/boxels are used. Apparently, the additional effort in evaluating the material law in the composite voxels/boxels is counterbalanced by the fewer cells required by the coarsening as the result of the smoother solution fields.

Our findings are neatly summarized in Figure 7.

For the highly resolved microstructure with a single spherical inclusion, the sandwich approach allows to reduce the number of material law evaluation by one order of magnitude. For the studied linear viscoelastic material law, this allows memory wise to either double the resolution for computations with similar memory footprint or to alternatively run the simulation in 25% of the runtime needed without sandwich coarsening. In particular, Figure 7 shows that 1% accuracy can be achieved for the relaxation test with only 2‰of the voxels when using laminate composite voxels/boxels and with 0.3%0when using additionally sandwich coarsening compared to the full 256^3 voxel resolution.

4.3 | A cross-ply

The next example we consider is a cross-ply, that is, two continuously reinforced fiber layers which are stacked atop each other and which are characterized by an angle of 90° relative to each other. In fact, continuously reinforced materials often show a strongly anisotropic material behavior. Using several layers with incrementally shifted fiber angle serves as a clever way to reduce the anisotropy of the mechanical properties, at least transverse to the stacking direction.

We consider a fiber-volume fraction of 30% which is typical for industrial applications. A resolution of 512^3 voxels serves as our reference, and we study four coarser discretization levels, as shown in Figure 8. We apply a creep loading in essentially the same way as we did for the single spherical inclusion, except that we apply the load to the cross ply in 45° direction,[51,52] that is, the weakest direction of the composite at hand, to generate maximum inelastic deformation. We

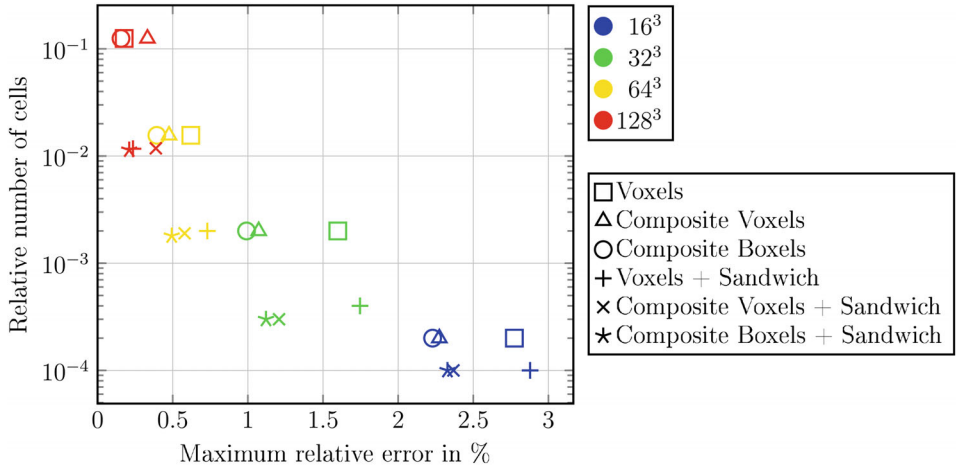


FIGURE 7 Accuracy versus cell-number reduction for the creep test of the spherical inclusion. The considered coarsening approaches are voxelization (Voxels) at lower resolution, sandwich-coarsening applied to the voxelized structures (Voxels+Sandwich), voxelization at lower resolution with laminate composite voxels (Composite Voxels) at the material interface, combination of the laminate composite voxels with sandwich coarsening (Composite Voxels + Sandwich), voxelization at lower resolution with composite boxels (Composite Boxels) at the material interface and a combination of composite boxels with sandwich coarsening (Composite Boxels + Sandwich).

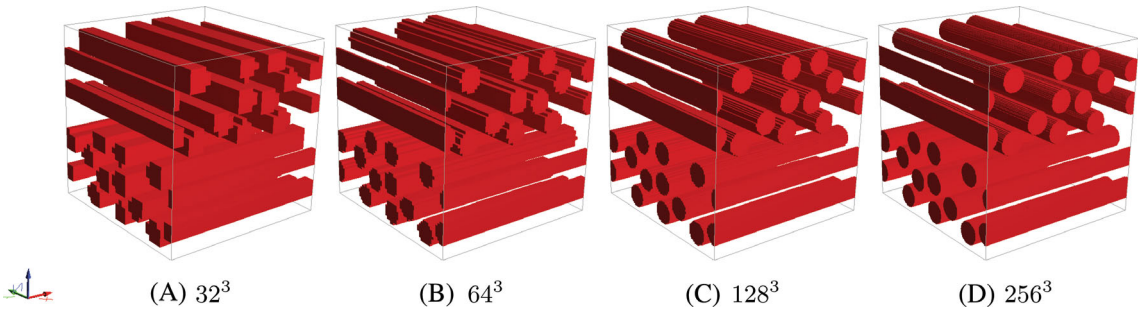


FIGURE 8 Microstructure of the cross ply at different resolutions.

investigate the same six coarsening approaches as for the single spherical inclusion at the four resolutions 32^3, 64^3, 128^3, and 256^3.

A diagram plotting the number of considered cells versus the achieved error relative to the full 512^3 resolution is shown in Figure 9. The observed results are completely different from the single spherical inclusion. For the coarsest resolution of 32^3 voxels, using composite voxels/boxels actually has a negative effect on the overall accuracy, irrespective of whether sandwich coarsening is used or not. The reason is found in the microstructure, for instance shown in Figure 10. Due to the high filler content, the matrix between the fibers is resolved only with a few voxels. More precisely, for the resolution with 256^3 voxels shown in Figure 10, there are fibers where only four voxels resolve the inter-fiber distance. Thus, only half a voxel resolves the diameter of this region on the 32^3 voxel-image. In particular, the composite voxel/boxel would be required to resolve essentially three phases—the two fibers which are close as well as the intermediate matrix region. Due to the inherently linear approximation of the interface within a composite voxel and the subsequent homogeneous approximation of the strain fields in the phases, laminate composite voxels/boxels cannot resolve the strain peak shown in Figure 10, which is responsible for the lower accuracy observed in Figure 9 at the coarsest considered resolution. Activating sandwich coarsening has little effect for this discretization level. Apparently, due to the high volume fraction and the associated close packing of the fibers there is simply no potential for coarsening. In turn, there is no reduction in the cell count, and, consequently, no change in accuracy.

The situation changes for the next higher resolution of 64^3 voxels. Composite boxels lead to a significant increase in accuracy, reducing the (maximum) relative error by almost a factor of two. Interestingly, the considered composite voxels do not lead to such an improvement, enhancing the accuracy of the plain-voxel approach only insignificantly. Recall that

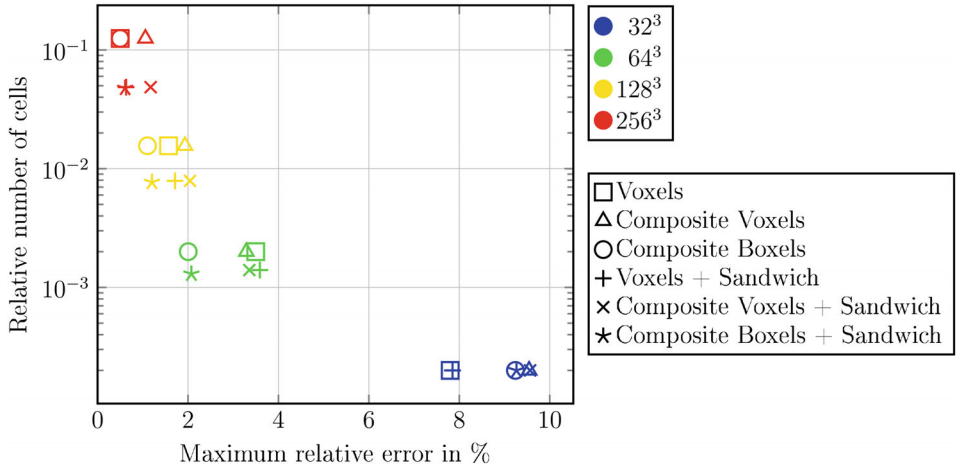


FIGURE 9 Accuracy versus cell-number reduction for the creep test of the cross ply. The considered coarsening approaches are voxelization (Voxels) at lower resolution, sandwich-coarsening applied to the voxelized structures (Voxels+Sandwich), voxelization at lower resolution with laminate composite voxels (Composite Voxels) at the material interface, combination of the laminate composite voxels with sandwich coarsening (Composite Voxels + Sandwich), voxelization at lower resolution with composite boxels (Composite Boxels) at the material interface and a combination of composite boxels with sandwich coarsening (Composite Boxels + Sandwich).

the critical difference of composite voxels and composite boxels for the article at hand is the way the interface normal is estimated. For composite voxels, this estimation is performed by connecting the centroid of a phase (e.g., of the matrix phase) to the centroid of the voxel, and normalizing the vector to unity.[53] Unfortunately, this procedure is not convergent under grid refinement.[54] The alternative strategy of Keshav et al.[48(section 5.2)] is based on Laplacian smoothing, and was shown to be more accurate. This gain in accuracy manifests for the study at hand, as well. Activating sandwich coarsening reduces the cell count by less than ten percent and does not lead to a significantly increased error, irrespective of whether composite voxels/boxels are used or not.

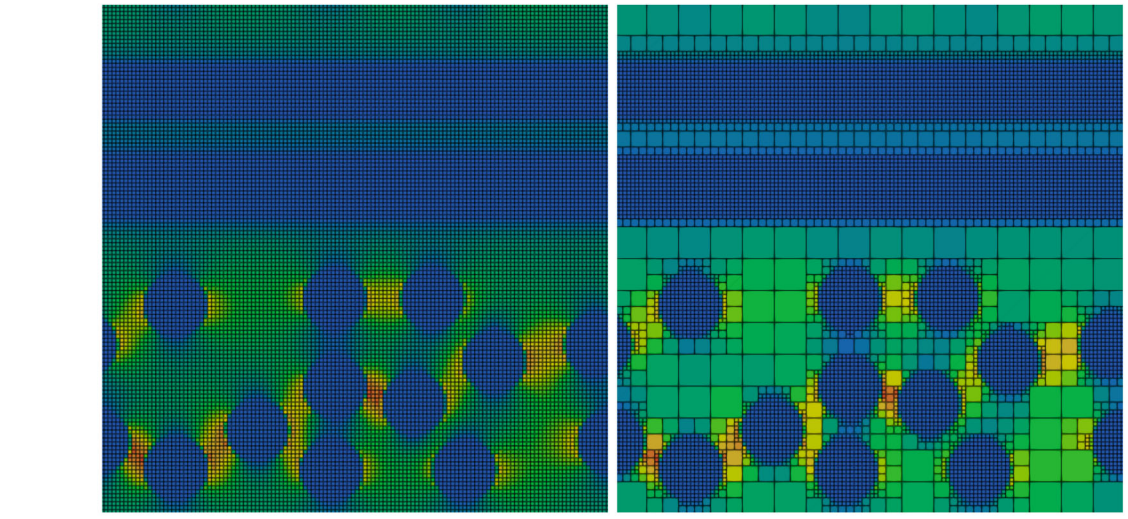
For the 128^3 voxel resolution, using composite voxels actually reduces the accuracy (but only slightly) compared to using no composite voxels at all. The composite boxels lead to a slight increase in accuracy compared to a regular voxel grid. The gain in accuracy, however, is less significant for this discretization level. This phenomenon is caused by the finer discretization, where the number of voxels at the interface, the thus the influence of the composite voxels/boxels, is decreased. The inferior accuracy of the composite voxels is caused by the lack of convergence under grid refinement of the normal-estimation technique.[54] Activating sandwich coarsening reduces the cell count while more or less preserving the level of accuracy. At 128^3 voxels, about 25% savings in the cell count is possible for sandwich coarsening.

For the highest considered resolution with 256^3 voxels, the accuracy of composite boxels and a fully resolved voxel grid coincide, essentially due to the poorly estimated volume fractions and normals. In fact, only eight subvoxels are available for downsampling. The normal-estimation approach for composite voxels is not successful, at all, whereas the composite voxels cannot profit from accurate volume fractions. For all considered scenarios, activating sandwich coarsening permits to save 50% of the cells, also preserving the accuracy levels.

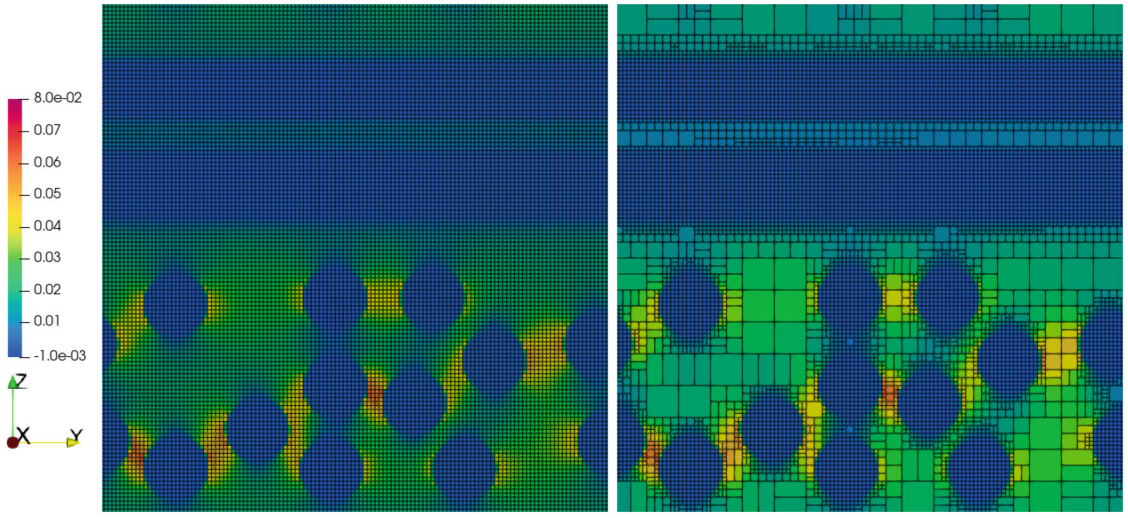
To sum up, we observe different "regimes" for the effectivity of the considered approaches. In case the discretization is too coarse, there is no hope to obtain accurate results. If the inter-fiber distance is resolved by at least a single voxel, composite voxel techniques will turn out to be useful. More precisely, accurately estimated normals, for instance via the technique described in Keshav et al.[48(section 5.2)], are critical for these methods. In this regime, sandwich coarsening is not effective, essentially due to the close fiber packing and the induced strong heterogeneity in the strain field. In fact, there is not much potential for coarsening. In the next regime, the discretization is fine enough to permit coarsening *and* producing an accurate solution, but where the influence of composite voxels is less severe, essentially due to the low interface-voxel count.

Last but not least, let us take a further look at Figure 9. We observe that 1% accuracy is obtained by composite boxels with 2% of the voxels of the comparison solution or with 8%0 sandwich cells.

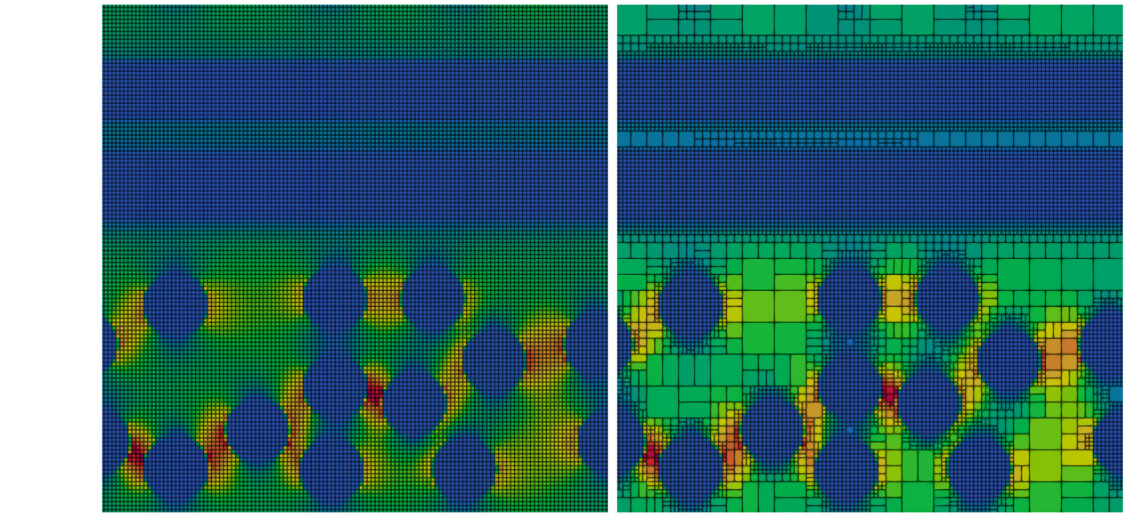
The local shear strains for a resolution of 256^3 voxels are shown in Figure 10. We consider a view from the front and compare the influence of using sandwich coarsening coupled to the composite-boxel technology. We chose this setup because it led to the most accurate solutions. In fact, we compare the solution fields and the meshes for three time steps:



(A) First time step



(B) Second time step



(C) End of simulation

FIGURE 10 Front view of local shear strain $\|\mathrm{dev}\boldsymbol{\epsilon}\|$ in x-direction for the cross ply: full resolution (left) versus sandwich coarsening with composite boxels (right).

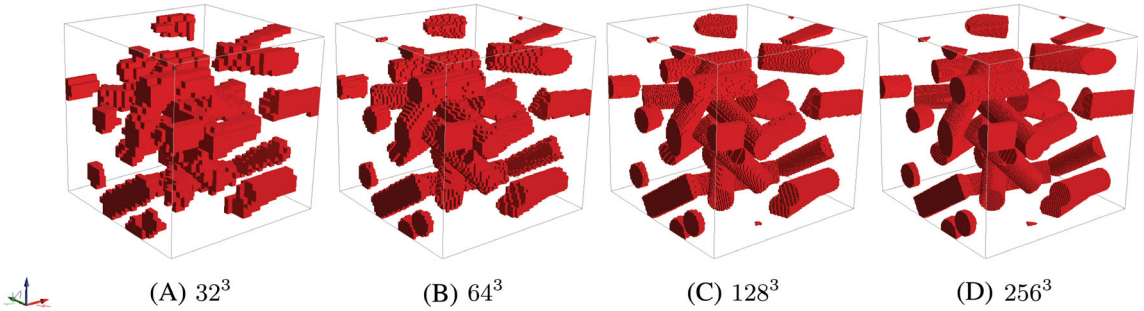


FIGURE 11 Microstructure of the fiber reinforced plastic at different resolutions.

the initial time step, the second time step and the final, 50th, time step. On the right-hand side, the local solution field associated to the sandwich-coarsened mesh is shown. Please note that we only show the average strain in the sandwich cells to emphasize the cell structure. The correct local strain field is actually heterogeneous in the cell due to the elastic stabilization term (23).

We observe a rather close agreement of the solution fields for the different time steps, illuminating the results we discussed earlier which were focused on the quality of the effective quantities. Concerning the sandwich coarsening, we start by looking at the first time step shown in Figure 10A. The fibers remain fully resolved as their behavior is linear elastic and the proposed coarsening strategy does not lead to a reduced memory consumption and runtime in this case. Due to the close fiber packing resulting from the industrial filler fraction, there is little potential to coarsen. In fact, due to the cylindrical geometry of the fibers, large (coarsened) boxes simply do not have enough space between the fibers.

The second time step shown in Figure 10B permits to assess the additional refinement required to accurately resolve the local strain field, in addition to the geometric constraints governing the first time step. Refinement is dominant for inter-fiber regions, essentially due to the comparatively large shear strains occurring in the vicinity of the fibers resulting from the deformation of the rather stiff individual fibers relative to each other, inducing a significant inter-fiber stresses inside the PP matrix.

At this point it also becomes apparent why it makes sense to consider sandwich coarsening instead of the simpler octree-coarsening strategy. When dealing with strongly anisotropic objects like fibers, more often than not the mechanical deformations essentially follow the fiber direction. Thus, there is a particular direction where the strains are close and sandwich coarsening applies. If sandwich coarsening is possible, only two instead of eight subcells emerge when decomposing a coarse voxel. Previous studies indicated that there is little gain in octree coarsening for fiber microstructures at industrial volume fractions, rendering the presented sandwich-coarsening strategy critical for reducing the cell count.

4.4 | Fiber structure

Our last example concerns a short-fiber reinforced composite microstructure. More precisely, we consider a commercially available polypropylene matrix with 30 weight percent glass-fiber reinforcements, which corresponds to a filler-volume fraction of 13%. We use short fibers with an aspect ratio, that is, the length-to-diameter ratio, of four. These comparatively short fibers are used to permit some degree of coarsening at all. The microstructure was generated with the SAM algorithm[55] and a prescribed second-order fiber-orientation tensor

$$\boldsymbol{A} = \operatorname{diag}(0.7, 0.2, 0.1), \tag{60}$$

employing the exact closure approximation.[56,57]

Aside from the fully resolved microstructure with 512^3 voxels, we consider four coarser discretization levels, comprising 32^3, 64^3, 128^3, and 256^3 voxels, see Figure 11. This time we are interested in the viscoplastic response of the composite, relying upon the viscoplastic model described in Section 4.1 which supplements the isotropic and linearly elastic E-glass fibers. Please notice that the parameters of the material model were fitted to the experimental data of a commercially used polypropylene matrix, see Grimm-Strele et al.[38] for details. In fact, the material model serves as a true challenge for the considered composite/voxel methods and coarsening strategies.

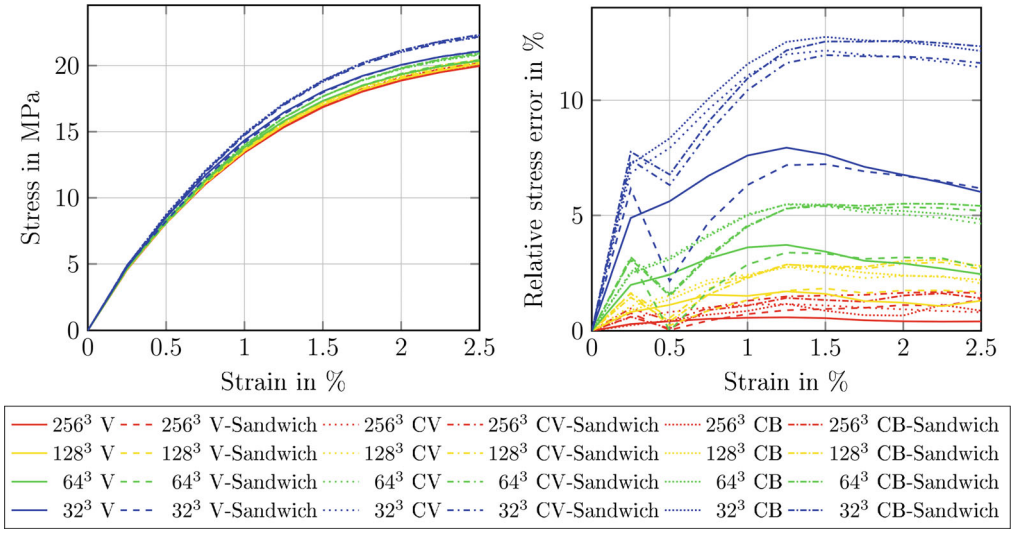


FIGURE 12 Stress-strain curves and relative error for the short-fiber reinforced plastic. The considered coarsening approaches are voxelization (V) at lower resolution, sandwich-coarsening applied to the voxelized structures (V-Sandwich), voxelization at lower resolution with laminate composite voxels (CV) at the material interface, combination of the laminate composite voxels with sandwich coarsening (CV-Sandwich), voxelization at lower resolution with composite boxels (CB) at the material interface and a combination of composite boxels with sandwich coarsening (CB-Sandwich).

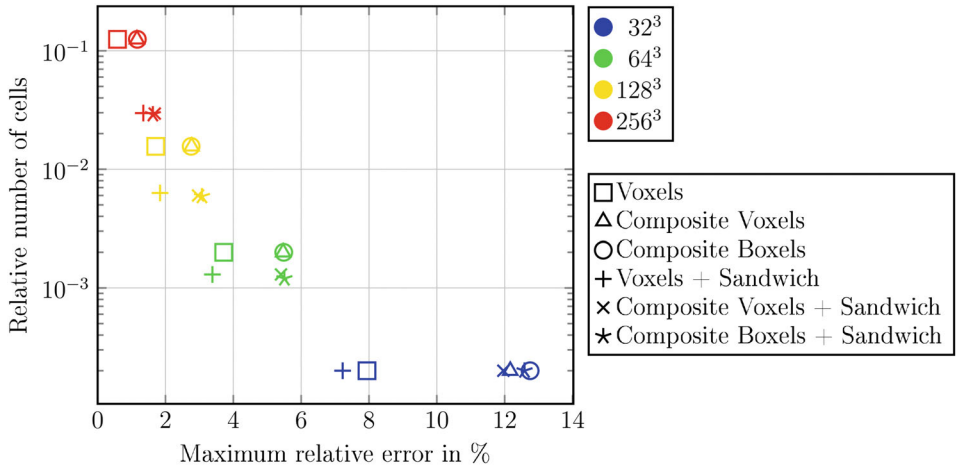
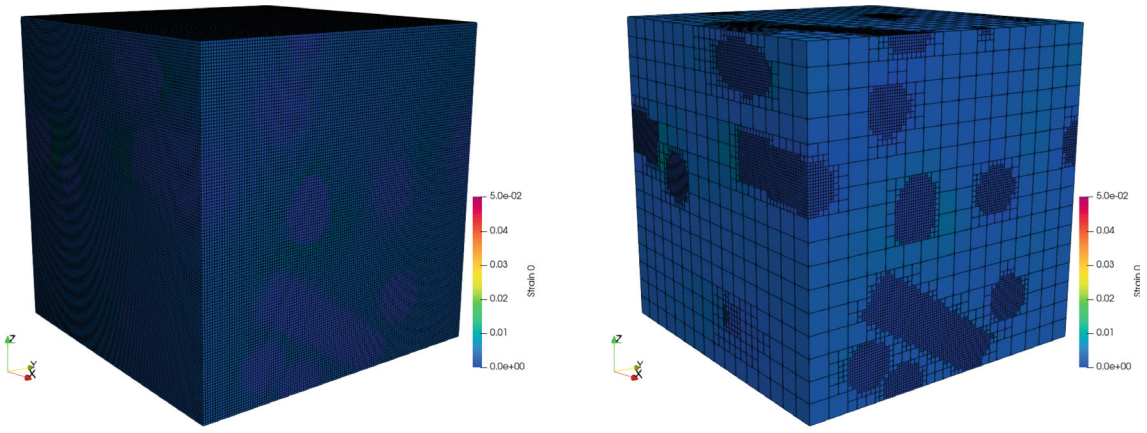


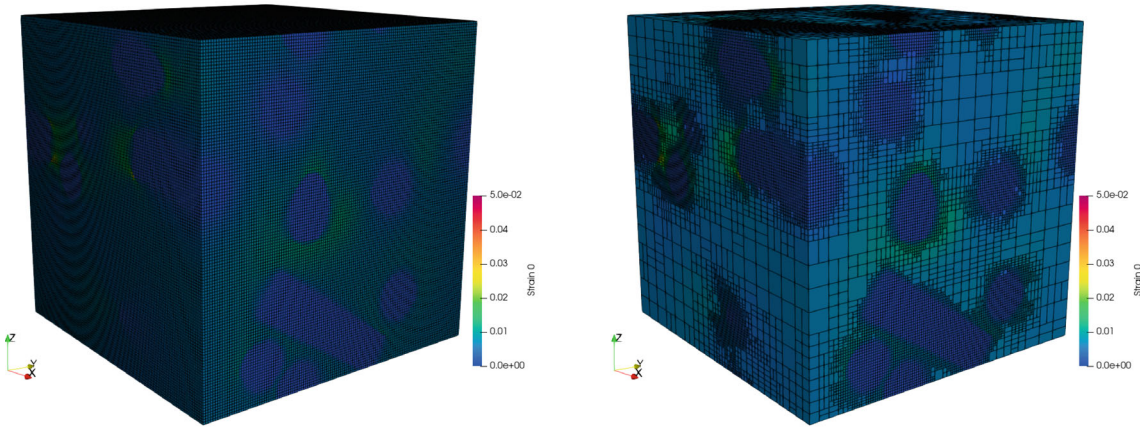
FIGURE 13 Accuracy versus cell-number reduction for the tensile test of the fiber reinforced plastic. The considered coarsening approaches are voxelization (Voxels) at lower resolution, sandwich-coarsening applied to the voxelized structures (Voxels+Sandwich), voxelization at lower resolution with laminate composite voxels (Composite Voxels) at the material interface, combination of the laminate composite voxels with sandwich coarsening (Composite Voxels + Sandwich), voxelization at lower resolution with composite boxels (Composite Boxels) at the material interface and a combination of composite boxels with sandwich coarsening (Composite Boxels + Sandwich).

Figure 12 shows the computed stress response up to 2.5% uniaxial strain loading, distributed over ten equidistant load steps with a strain rate of $1.7 \times 10^{-4} s^{-1}$. We observe that the different scenarios show a high degree of similarity for the first one or two time steps only, that is, in the elastic regime up to about 0.5% macroscopic strain. For a higher degree of loading, strong deviations of the stress predictions at coarse resolutions emerge compared to the reference curve. These discrepancies become more apparent when studying the relative errors in the axial stress, shown in the right part of Figure 12.

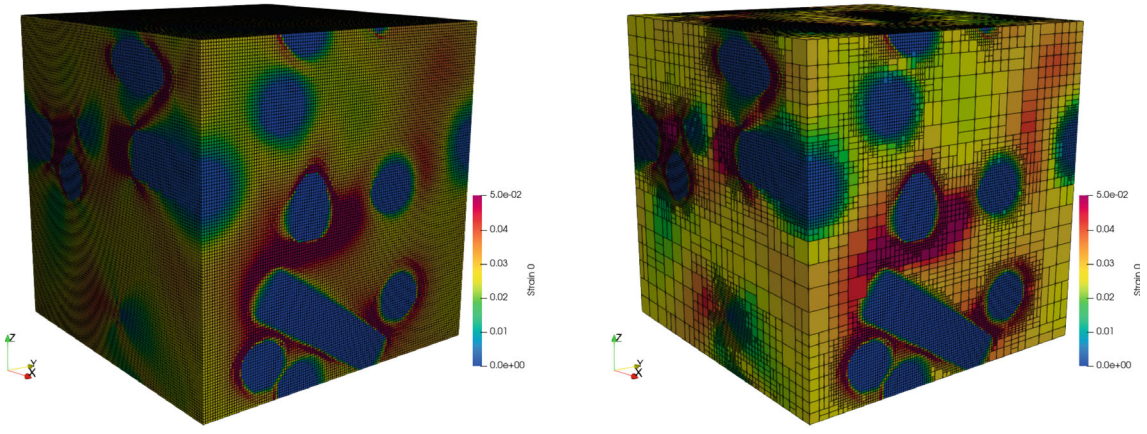
For the coarsest resolution considered, 32^3 voxels, all strategies produce a stress response that exceeds a relative error of 5% except for the initial elastic steps and an outlier at 0.5% loading. Taking a closer look at the fully resolved voxel



(A) Initial octree



(B) First time step



(C) End of simulation

FIGURE 14 Local Mises strain for FRP: voxel mesh versus sandwich coarsening.

model, we observe an increase of the relative error up to 1.25% strain, together with a subsequent decrease of the error. It appears as if the resolution is simply too coarse to provide a mechanical response with reasonable accuracy. Using either composite voxels or composite boxels leads to a consistently higher error than the plain-voxel model. In contrast to the latter, the error increases steadily for higher loading and composite voxels/boxels—with the exception of a slight dip towards the end of the loading window. Apparently, the benefits of composite voxels/boxels cannot be exploited for such a coarse resolution. The influence of activating sandwich coarsening on the results is not easily discerned.

For a resolution with 64^3 voxels, the plain-voxel model provides a relative error below 4%, whereas composite voxels/boxels come with an increased error level at around 6%. It appears that the composite voxels have trouble resolving the emerging strongly plastified regions irrespective of the utilized normal-estimation technique. Concerning an additional sandwich coarsening, its activating preserves the error level.

Rather similar conclusions may be drawn for the higher resolutions, although the emerging error levels are different for the various considered scenarios and combinations. Remarkably, we do not observe different regimes as we did for the cross-ply. This fact may hint at a lack of mesh convergence at 512^3 voxels.

To weigh accuracy against cell count, we study Figure 13 more closely. A maximum relative error of 2% can be obtained using only 5%0 of the cells corresponding to the full 512^3 resolution when using sandwich coarsening is active. Without the coarsening, 1.5% of the voxels of the reference mesh are required to reach a similar accuracy. Thus, we can save a factor of three in this regard when using sandwich coarsening.

Last but not least, the local solution fields on the 256^3 voxel resolution are shown in Figure 14, together with the cells of the sandwich coarsened mesh. Notice that we did not activate composite voxels/boxels as they did not lead to an improved accuracy.

When inspecting the initial step, shown in Figure 14A, the purely geometric preprocessing is unable to coarsen the matrix significantly. This appears a bit odd when inspecting the meshes on the surfaces of the volume only. However, due to the presence of the short fibers, placing larger volumes is actually prohibited. In this sense, short-fiber geometries turn out to be rather difficult for such coarsening strategies. In fact, more sophisticated numerical strategies, for example, adaptive finite elements, would run into essentially the same problem: the strong anisotropy and the high filler fraction of the fibers limits the effectiveness of classical adaptive meshing strategies. In fact, relying upon the simpler octree coarsening would lead to little reduction of the cell count, and the considered sandwich coarsening turns out to be a critical advantage.

After the first time step, see Figure 14B, a strong refinement is necessary in regions where fibers get close. In fact, the strong nonlinearity of the viscoplastic material makes this degree of refinement necessary. The final time step, see Figure 14C, does not appear to increase the cell count significantly, as probably expected from a loading scenario without a change of loading direction.

All in all, the (cell-wise averages of the) strain field are captured accurately on the coarsened grid, increasing confidence in the selected parameters for the refinement criterion (52).

5 | SUMMARY AND CONCLUSION

For typical industrial-scale computational micromechanics simulations, most time is spent on evaluating the nonlinear constitutive laws. Recently, Kaiser et al.[9] showed that interpolating wavelets can be integrated into the framework of FFT-based computational micromechanics, enabling spatial adaptivity to be exploited in this context.

The work at hand aimed at simplifying and, at the same time, extending the range of applicability of this adaptive technology. More precisely, we abstracted the concept of stress coarsening and demonstrated that such a strategy needs to be stabilized by a suitable term in a similar fashion as for hourglass control. Also, we identified the material tangent to be a suitable elasticity tensor for the stabilization, coming with an implied accuracy guarantee as an essential bonus.

We showed that octrees give rise to a powerful coarsening strategy which is moreover readily integrated into existing FFT-based computational homogenization codes. Moreover, we demonstrated that using sandwich coarsening can be quite useful to reduce the number of cells further. The computational experiments revealed the potential of the introduced piece of technology for reducing both runtime and memory footprint, in particular for highly resolved microstructures where high accuracy is sought. A convenient feature of the discussed strategies is that they are not only intrinsically compatible with composite voxel/boxel techniques, but may even profit from them, because their use leads to smoother solution fields which warrant a lower degree of refinement is necessary. However, we also noticed that there is a limitation on the possible coarsening for industrial microstrostructures with an inherently high complexity. In fact, it does not come

as a surprise that coarsening cannot be effective in the presence of high complexity—unless the complexity is introduced artificially, for example, by resolving a single inclusion on an excessively fine grid.

Future works could further study the refinement criterion (52), improve upon the initial tree setup or introduce different coarsening options.

ACKNOWLEDGMENTS

MK acknowledges support from the German Federal Ministry for Economic Affairs and Climate Action within the TTP-LB project DigiLaugBeh (03LB3044B). Support from the European Research Council within the Horizon Europe program—project 101040238—is gratefully acknowledged by MS. We thank H. Grimm-Strele for constructive feedback on an earlier version of the manuscript. We are grateful to the anonymous reviewers for their encouraging and detailed feedback on the manuscript. Open Access funding enabled and organized by Projekt DEAL.

DATA AVAILABILITY STATEMENT

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

ORCID

Matthias Kabel https://orcid.org/0000-0001-8869-2387
Matti Schneider https://orcid.org/0000-0001-7017-3618

REFERENCES

1. Moulinec H, Suquet P. A fast numerical method for computing the linear and nonlinear mechanical properties of composites. *Comptes Rendus de l'Académie Des Sciences Série II*. 1994;318(11):1417-1423.
2. Moulinec H, Suquet P. A numerical method for computing the overall response of nonlinear composites with complex microstructure. *Comput Methods Appl Mech Eng*. 1998;157:69-94.
3. Frigo M, Johnson SG. The design and implementation of FFTW3. *Proceedings of the IEEE*. Vol 93. IEEE; 2005:216-231.
4. Dalcin L, Mortensen M, Keyes DE. Fast parallel multidimensional FFT using advanced MPI. *J Parallel Distribut Comput*. 2019;128:137-150.
5. Schneider M. A review of non-linear FFT-based computational homogenization methods. *Acta Mech*. 2021;232:2051-2100.
6. 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.
7. Bornemann F, Deuflhard P. The Cascadic multigrid method for elliptic problems. *Numer Math*. 1996;75:135-152.
8. Nkoumbou Kaptchouang NB, Gélébart L. Multiscale coupling of FFT-based simulations with the LDC approach. *Comput Methods Appl Mech Eng*. 2022;394:114921.
9. Kaiser T, Raasch T, Remmers JJC, Geers MGD. A wavelet-enhanced adaptive hierarchical FFT-based approach for the efficient solution of microscale boundary value problems. *Comput Methods Appl Mech Eng*. 2023;409:115959.
10. Donoho DL. *Interpolating Wavelet Transforms*. Preprint. Vol 2. Department of Statistics, Stanford University; 1992:1-54.
11. Deslauriers G, Dubuc S. Symmetric iterative interpolation processes. *Constr Approx*. 1989;5:49-68.
12. Haar A. Zur Theorie der orthogonalen Funktionensysteme. *Math Ann*. 1910;69:331-371.
13. Nakassima G, Gameiro M. A framework for rigorous computational methods using Haar wavelets for differential equations. arXiv preprint, arXiv:2304.14536. 2023;1-33.
14. Samet H. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley; 1990.
15. Dvorak GJ. Transformation field analysis of inelastic composite materials. *Proc Math Phys Sci*. 1900;1992(437):311-327.
16. Dvorak G, Benveniste Y. On transformation strains and uniform fields in multiphase elastic media. *Proc Royal Soc A*. 1992;437:291-310.
17. Flanagan DP, Belytschko T. A uniform strain hexahedron and quadrilateral with orthogonal hourglass control. *Int J Numer Methods Eng*. 1981;17:679-706.
18. Reese S, Wriggers P. A stabilization technique to avoid hourglassing in finite elasticity. *Int J Numer Methods Eng*. 2000;48:79-109.
19. Gargantini I. Linear octtrees for fast processing of three-dimensional objects. *Computer Vision Graphics Image Proc*. 1982;20:365-374.
20. Schneider M. Lippmann-Schwinger solvers for the computational homogenization of materials with pores. *Int J Numer Methods Eng*. 2020;121(22):5017-5041.
21. Schneider M. Voxel-based finite elements with hourglass control in FFT-based computational homogenization. *Int J Numer Methods Eng*. 2022;123(24):6286-6313.
22. Halphen B, Nguyen QS. Sur les matériaux standards generalisés. *J Mécanique*. 1975;14:508-520.
23. Nguyen QS. *Stability and Nonlinear Solid Mechanics*. Wiley; 2000.
24. Ortiz M, Stainier L. The variational formulation of viscoplastic constitutive updates. *Comput Methods Appl Mech Eng*. 1999;171:419-444.
25. 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.

26. Chaboche JL, Kanouté P, Roos A. On the capabilities of mean-field approaches for the description of plasticity in metal matrix composites. *Int J Plast*. 2005;21:1409-1434.
27. Kabel M, Fink A, Schneider M. The composite voxel technique for inelastic problems. *Comput Methods Appl Mech Eng*. 2017;322:396-418.
28. Linden S, Wiegmann A, Hagen H. The LIR space partitioning system applied to the stokes equations. *Graph Model*. 2015;82:58-66.
29. Morton GM. *A Computer Oriented Geodetic Data Base; and a New Technique in File Sequencing*. tech rep.; IBM Ltd.; 1966.
30. Hasbestan JJ, Senocak I. Binarized-octree generation for Cartesian adaptive mesh refinement around immersed geometries. *J Comput Phys*. 2018;368:179-195.
31. 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.
32. Schneider M. On the Barzilai-Borwein basic scheme in FFT-based computational homogenization. *Int J Numer Methods Eng*. 2019;118(8):482-494.
33. 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.
34. 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.
35. 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.
36. Wicht D, Schneider M, Böhlke T. On quasi-Newton methods in FFT-based micromechanics. *Int J Numer Methods Eng*. 2020;121(8):1665-1694.
37. Schneider M. A dynamical view of nonlinear conjugate gradient methods with applications to FFT-based computational micromechanics. *Comput Mech*. 2020;66:239-257.
38. Grimm-Strele H, Kabel M, Andrä H, et al. Efficient characterization and modelling of the nonlinear behaviour of LFT for crash simulations. In: Chinesta F, Abgrall R, Allix O, Kaliske M, eds. *Wccm-ECCOMAS2020*. SCIPEDIA; 2021:1-12.
39. Fritzen F, Böhlke T. Reduced basis homogenization of viscoelastic composites. *Compos Sci Technol*. 2013;76:84-91.
40. Steiner K. *MISES-FOK: Multiskalenintegrierende Struktureigenschaftssimulation der Faserorientierung für faserverstärkte Kunststoffe Im Automobil-Und Flugzeugbau: Abschlussbericht Zum Forschungsvorhaben 03x0513F Im BMBF-Rahmenprogramm WING: Bearbeitungszeitraum Vom 01.03.2007 Bis 31.12.2011*. tech. rep.; Fraunhofer-Institut für Techno-und Wirtschaftsmathematik ITWM; 2012.
41. Kaliske M, Rothert H. Formulation and implementation of three-dimensional viscoelasticity at small and finite strains. *Comput Mech*. 1997;19(3):228-239.
42. Staub S, Andrä H, Kabel M. Fast FFT based solver for rate-dependent deformations of composites and nonwovens. *Int J Solids Struct*. 2018;154:33-42.
43. Blühdorn J, Gauger NR, Kabel M. AutoMat: automatic differentiation for generalized standard materials on GPUs. *Comput Mech*. 2022;69(2):589-613.
44. Math2Market GmbH, "*GeoDict*." http://www.geodict.de, 2023.
45. Willot F. Fourier-based schemes for computing the mechanical response of composites with accurate local fields. *Comptes Rendus Mécanique*. 2015;343:232-245.
46. Schneider M, Ospald F, Kabel M. Computational homogenization of elasticity on a staggered grid. *Int J Numer Methods Eng*. 2016;105(9):693-720.
47. Kabel M, Merkert D, Schneider M. Use of composite voxels in FFT-based homogenization. *Comput Methods Appl Mech Eng*. 2015;294:168-188.
48. Keshav S, Fritzen F, Kabel M. FFT-based homogenization at finite strains using composite Boxels (ComBo). *Comput Mech*. 2022;71:191-212.
49. Schneider M, Wicht D. Superconvergence of the effective Cauchy stress in computational homogenization of inelastic materials. *Int J Numer Methods Eng*. 2023;124(4):959-978.
50. Schneider M. On the effectiveness of the Moulinec-Suquet discretization for composite materials. *Int J Numer Methods Eng*. 2023;124(14):3191-3218.
51. Kabel M, Fliegener S, Schneider M. Mixed boundary conditions for FFT-based homogenization at finite strains. *Comput Mech*. 2016;57(2):193-210.
52. Grimm-Strele H, Kabel M. FFT-based homogenization with mixed uniform boundary conditions. *Int J Numer Methods Eng*. 2021;122:7241-7265.
53. Merkert D, Andrä H, Kabel M, Schneider M, Simeon B. *An Efficient Algorithm to Include Sub-Voxel Data in FFT-Based Homogenization for Heat Conductivity*. Springer; 2015:267-279.
54. Lendvai J, Schneider M. Assumed strain methods in micromechanics, laminate composite voxels and level sets. *Int J Numer Methods Eng*. 2023;1-37.
55. 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.

56. Montgomery-Smith S, He W, Jack D, Smith D. Exact tensor closures for the three-dimensional Jeffery's equation. *J Fluid Mech*. 2011;680:321-335.
57. Montgomery-Smith S, Jack D, Smith DE. The fast exact closure for Jeffery's equation with diffusion. *J Non-Newtonian Fluid Mech*. 2011;166:343-353.

How to cite this article: Kabel M, Schneider M. Adaptive material evaluation by stabilized octree and sandwich coarsening in FFT-based computational micromechanics. *Int J Numer Methods Eng*. 2024;125(5):e7399. doi: 10.1002/nme.7399

APPENDIX . ARGUMENTS FOR THE PROPERTIES OF THE COARSENED STRESS

The purpose of this appendix is to show that the stress operator (23)

$$\tilde{\mathbb{s}} : L^2(Y;\mathrm{Sym}(d)) \to L^2(Y;\mathrm{Sym}(d)), \quad \boldsymbol{\varepsilon} \mapsto \mathbb{s}(\mathbb{P} : \boldsymbol{\varepsilon}) + \mathbb{C}^{\mathtt{stab}} : \mathbb{Q} : \boldsymbol{\varepsilon}, \tag{A1}$$

defined in Section 2.2 inherits a number of salient properties from the original stress operator $\mathbb{s}$ provided the eigenvalues bounds (22)

$$\alpha_- \, \|\boldsymbol{\varepsilon}\|^2 \leq \boldsymbol{\varepsilon} : \mathbb{C}^{\mathtt{stab}}(\boldsymbol{x}) : \boldsymbol{\varepsilon} \leq \alpha_+ \|\boldsymbol{\varepsilon}\|^2 \tag{A2}$$

are satisfied and the the compatibility conditions (24)

$$\mathbb{s}(\mathbb{P} : \boldsymbol{\varepsilon}) = \mathbb{P} : \mathbb{s}(\mathbb{P} : \boldsymbol{\varepsilon}) \tag{A3}$$

and (25)

$$\mathbb{C}^{\mathtt{stab}} : \mathbb{Q} : \boldsymbol{\varepsilon} = \mathbb{Q} : \mathbb{C}^{\mathtt{stab}} : \mathbb{Q} : \boldsymbol{\varepsilon} \tag{A4}$$

hold. For a start, we assume that the stress operator $\mathbb{s}$ is α_+-Lipschitz continuous, that is, the estimate

$$\|\mathbb{s}(\boldsymbol{\varepsilon}_1) - \mathbb{s}(\boldsymbol{\varepsilon}_2)\|_{L^2} \leq \alpha_+ \|\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2\|_{L^2}, \quad \boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2 \in L^2(Y;\mathrm{Sym}(d)) \tag{A5}$$

is valid. We want to show that this inequality holds for the modified stress operator $\tilde{\mathbb{s}}$ defined in Equation (23), as well. For this purpose, we write

$$\tilde{\mathbb{s}}(\boldsymbol{\varepsilon}_i) = \mathbb{s}(\mathbb{P} : \boldsymbol{\varepsilon}_i) + \mathbb{C}^{\mathtt{stab}} : \mathbb{Q} : \boldsymbol{\varepsilon}_i = \mathbb{P} : \mathbb{s}(\mathbb{P} : \boldsymbol{\varepsilon}_i) + \mathbb{Q} : \mathbb{C}^{\mathtt{stab}} : \mathbb{Q} : \boldsymbol{\varepsilon}_i, \quad i = 1, 2, \tag{A6}$$

where we used the compatibility conditions (A3) and (A4). Consequently, we may express the difference in the form

$$\tilde{\mathbb{s}}(\boldsymbol{\varepsilon}_1) - \tilde{\mathbb{s}}(\boldsymbol{\varepsilon}_2) = \mathbb{P} : \big[\mathbb{s}(\mathbb{P} : \boldsymbol{\varepsilon}_1) - \mathbb{s}(\mathbb{P} : \boldsymbol{\varepsilon}_2)\big] + \mathbb{Q} : \mathbb{C}^{\mathtt{stab}} : \mathbb{Q} : [\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2]. \tag{A7}$$

Taking squared L^2-norms on both hands of the equation and using the Pythagorean Theorem (16), a consequence of the fact that $\mathbb{P}$ and $\mathbb{Q}$ are complementary orthogonal projectors, yields the equation

$$\|\tilde{\mathbb{s}}(\boldsymbol{\varepsilon}_1) - \tilde{\mathbb{s}}(\boldsymbol{\varepsilon}_2)\|_{L^2}^2 = \big\|\mathbb{P} : \big[\mathbb{s}(\mathbb{P} : \boldsymbol{\varepsilon}_1) - \mathbb{s}(\mathbb{P} : \boldsymbol{\varepsilon}_2)\big]\big\|_{L^2}^2 + \big\|\mathbb{Q} : \mathbb{C}^{\mathtt{stab}} : \mathbb{Q} : [\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2]\big\|_{L^2}^2. \tag{A8}$$

In view of the compatibility conditions (A3) and (A4), we may get rid of the $\mathbb{P}$- and $\mathbb{Q}$-prefactors, respectively, to arrive at the identity

$$\|\tilde{\mathbb{s}}(\boldsymbol{\varepsilon}_1) - \tilde{\mathbb{s}}(\boldsymbol{\varepsilon}_2)\|_{L^2}^2 = \|\mathbb{s}(\mathbb{P} : \boldsymbol{\varepsilon}_1) - \mathbb{s}(\mathbb{P} : \boldsymbol{\varepsilon}_2)\|_{L^2}^2 + \big\|\mathbb{C}^{\mathtt{stab}} : \mathbb{Q} : [\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2]\big\|_{L^2}^2. \tag{A9}$$

Due to the estimates (A5) and (A2), we obtain the inequality

$$\|\tilde{\mathbb{S}}(\boldsymbol{\varepsilon}_1) - \tilde{\mathbb{S}}(\boldsymbol{\varepsilon}_2)\|_{L^2}^2 \leq \alpha_+^2 \big(\|\mathbb{P} : \boldsymbol{\varepsilon}_1 - \mathbb{P} : \boldsymbol{\varepsilon}_2\|_{L^2}^2 + \|\mathbb{Q} : [\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2]\|_{L^2}^2 \big). \tag{A10}$$

By the Pythagorean Theorem (16), we are thus led to the inequality

$$\|\tilde{\mathbb{S}}(\boldsymbol{\varepsilon}_1) - \tilde{\mathbb{S}}(\boldsymbol{\varepsilon}_2)\|_{L^2}^2 \leq \alpha_+^2 \|\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2\|_{L^2}^2, \tag{A11}$$

which, after taking the square root, just encodes the α_+-Lipschitz continuity of the stress operator $\tilde{\mathbb{S}}$.

After establishing the continuity, we move towards the monotonicity properties. We assume α_--strong monotonicity of the stress operator $\mathbb{S}$, that is, the inequality

$$\langle (\mathbb{S}(\boldsymbol{\varepsilon}_1) - \mathbb{S}(\boldsymbol{\varepsilon}_2)) : (\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2) \rangle_Y \geq \alpha_- \, \|\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2\|_{L^2}^2, \quad \boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2 \in L^2(Y; \mathrm{Sym}(d)). \tag{A12}$$

Multiplying the expression (A7) by the difference $\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2$, we are led to the identity

$$\begin{aligned} \langle (\tilde{\mathbb{S}}(\boldsymbol{\varepsilon}_1) - \tilde{\mathbb{S}}(\boldsymbol{\varepsilon}_2)) : (\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2) \rangle_Y &= \big\langle (\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2) : \mathbb{P} : \big[\mathbb{S}(\mathbb{P} : \boldsymbol{\varepsilon}_1) - \mathbb{S}(\mathbb{P} : \boldsymbol{\varepsilon}_2) \big] \big\rangle_Y \\ &\quad + \big\langle [\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2] : \mathbb{Q} : \mathbb{C}^{\mathtt{stab}} : \mathbb{Q} : [\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2] \big\rangle_Y, \end{aligned} \tag{A13}$$

valid for all $\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2 \in L^2(Y; \mathrm{Sym}(d))$. In view of the estimates from below (22) and (A12), we derive the inequality

$$\langle (\tilde{\mathbb{S}}(\boldsymbol{\varepsilon}_1) - \tilde{\mathbb{S}}(\boldsymbol{\varepsilon}_2)) : (\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2) \rangle_Y \geq \alpha_- \big(\|\mathbb{P} : (\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2)\|_{L^2}^2 + \|\mathbb{Q} : (\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2)\|_{L^2}^2 \big). \tag{A14}$$

The Pythagorean Theorem (16) finally implies the desired monotonicity inequality

$$\langle (\tilde{\mathbb{S}}(\boldsymbol{\varepsilon}_1) - \tilde{\mathbb{S}}(\boldsymbol{\varepsilon}_2)) : (\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2) \rangle_Y \geq \alpha_- \, \|\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2\|_{L^2}^2, \tag{A15}$$

establishing property 2. in Section 2.2.

We conclude this appendix by establishing the integral representation (26)

$$\mathbb{S}(\boldsymbol{\varepsilon}) - \tilde{\mathbb{S}}(\boldsymbol{\varepsilon}) = \int_0^1 \Big[\frac{\partial \mathbb{S}}{\partial \boldsymbol{\varepsilon}} (\mathbb{P} : \boldsymbol{\varepsilon} + s \, \mathbb{Q} : \boldsymbol{\varepsilon}) - \mathbb{C}^{\mathtt{stab}} \Big] : \mathbb{Q} : \boldsymbol{\varepsilon} \, ds \tag{A16}$$

of the difference between the actions of the two considered stress operators $\mathbb{S}$ and $\tilde{\mathbb{S}}$. If the stress operator $\mathbb{S}$ is continuously differentiable, the Fundamental Theorem of Calculus implies the representation

$$\mathbb{S}(\boldsymbol{\varepsilon}_1) = \mathbb{S}(\boldsymbol{\varepsilon}_2) + \int_0^1 \frac{\partial \mathbb{S}}{\partial \boldsymbol{\varepsilon}} (\boldsymbol{\varepsilon}_2 + s(\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2)) : (\boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2) \, ds, \quad \boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2 \in L^2(Y; \mathrm{Sym}(d)). \tag{A17}$$

For a given strain field $\boldsymbol{\varepsilon} \in L^2(Y; \mathrm{Sym}(d))$, we set

$$\boldsymbol{\varepsilon}_1 = \boldsymbol{\varepsilon} \quad \text{and} \quad \boldsymbol{\varepsilon}_2 = \mathbb{P} : \boldsymbol{\varepsilon}, \quad \text{that is,} \quad \boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_2 = \mathbb{Q} : \boldsymbol{\varepsilon}, \tag{A18}$$

and obtain the representation

$$\mathbb{S}(\boldsymbol{\varepsilon}) - \mathbb{S}(\mathbb{P} : \boldsymbol{\varepsilon}) = \int_0^1 \frac{\partial \mathbb{S}}{\partial \boldsymbol{\varepsilon}} (\mathbb{P} : \boldsymbol{\varepsilon} + s \, \mathbb{Q} : \boldsymbol{\varepsilon}) : \mathbb{Q} : \boldsymbol{\varepsilon} \, ds, \quad \boldsymbol{\varepsilon} \in L^2(Y; \mathrm{Sym}(d)), \tag{A19}$$

that is, after subtracting the term $\mathbb{C}^{\mathtt{stab}} : \mathbb{Q} : \boldsymbol{\varepsilon}$, we conclude the validity of the error representation (A16). The estimate (27) follows directly by applying the triangle inequality to the error representation (A16) and using the upper bounds (A2) and (A5).

DOI: 10.1002/nme.7399
URN: urn:nbn:de:hbz:465-20240710-153021-9