Received: 16 July 2019 | Revised: 27 November 2019 | Accepted: 28 November 2019

DOI: 10.1002/nme.6283

RESEARCH ARTICLE

WILEY

On Quasi-Newton methods in fast Fourier transform-based micromechanics

Daniel Wicht | **Matti Schneider** | **Thomas Böhlke**

Karlsruhe Institute of Technology, Institute of Engineering Mechanics, Karlsruhe, Germany

Correspondence
Thomas Böhlke, Karlsruhe Institute of Technology, Institute of Engineering Mechanics, Karlsruhe 76131, Germany.
Email: thomas.boehlke@kit.edu

Funding information
German Research Foundation (DFG), Grant/Award Number: "Integrated engineering of continuous-discontinuous long fiber reinforced polymer structures" (GRK 2078); German Research Foundation (DFG), Grant/Award Number: "Lamellar Fe-Al in situ composite materials" (BO 1466/12-2)

SUMMARY

This work is devoted to investigating the computational power of Quasi-Newton methods in the context of fast Fourier transform (FFT)-based computational micromechanics. We revisit FFT-based Newton-Krylov solvers as well as modern Quasi-Newton approaches such as the recently introduced Anderson accelerated basic scheme. In this context, we propose two algorithms based on the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method, one of the most powerful Quasi-Newton schemes. To be specific, we use the BFGS update formula to approximate the global Hessian or, alternatively, the local material tangent stiffness. Both for Newton and Quasi-Newton methods, a globalization technique is necessary to ensure global convergence. Specific to the FFT-based context, we promote a Dong-type line search, avoiding function evaluations altogether. Furthermore, we investigate the influence of the forcing term, that is, the accuracy for solving the linear system, on the overall performance of inexact (Quasi-)Newton methods. This work concludes with numerical experiments, comparing the convergence characteristics and runtime of the proposed techniques for complex microstructures with nonlinear material behavior and finite as well as infinite material contrast.

KEYWORDS

BFGS, composites, crystal viscoplasticity, FFT-based micromechanics, homogenization, Quasi-Newton methods

1 | INTRODUCTION

Fast Fourier transform (FFT)-based solution schemes[5,6] have become an established tool in computational micromechanics. Their popularity rests on their computational efficiency, the ability to treat nonlinear material behavior and the compatibility to imaging techniques such as microcomputed tomography. Owing to these advantages, FFT-based solvers have found widespread application in various fields, including the homogenization of polycrystals at small[7] and finite strains,[8] damage[9] and fracture mechanics,[10] fatigue prediction,[11] electromechanically coupled materials,[12] and concurrent multiscale simulations.[13,14]

[Correction added on 14 February 2020, after first online publication: Funding information has been updated.]

A considerable amount of research effort has been devoted to developing FFT-based solution algorithms. A limited set of solvers needs to be identified, which are computationally efficient and versatile enough to cover a broad spectrum of applications, for example, nonlinear material behavior and materials with high or infinite contrast. The original solver of Moulinec and Suquet,[6] the so-called basic scheme, was based on the Lippmann-Schwinger formulation of elasticity. It is characterized by several favorable properties such as a small memory footprint and a tangent-free treatment of nonlinear material behavior. However, its required iteration count is proportional to the material contrast. Thus, for certain practical applications such as the homogenization of plastifying or porous materials, the convergence behavior of the basic scheme can be exceedingly slow. To accelerate the solution process, Eyre and Milton[15] introduced a polarization-based scheme by reformulating the Lippmann-Schwinger equation. Damped versions of the algorithm were proposed by Michel et al[16] and Monchiet and Bonnet.[17] By interpreting the polarization-based schemes as versions of the Douglas-Rachford splitting, their applicability was extended to the nonlinear case.[18] Whereas these methods exhibit excellent performance for finitely contrasted media, it was shown that they may converge slower than the basic scheme for porous materials,[19] limiting their versatility. Zeman et al[20] and Brisard and Dormieux[21,22] applied Krylov-subspace solvers to FFT-based homogenization. These methods are extremely fast, but they are restricted to linear problems. However, they may enter inexact Newton-methods, constituting a powerful class of solvers with broad applications. In the context of FFT-based computational homogenization, Newton's method was combined with the conjugate-gradient (CG) solver in the small-[1] and finite-strain setting[2] and exhibited excellent performance. Due to the small number of required function evaluations, these schemes proved to be particularly powerful for problems with computationally expensive material laws, such as single-crystal plasticity,[3,11,23] whose evaluation dominates the overall runtime. In contrast to the aforementioned schemes, the Newton-CG solver requires the evaluation of the material's tangent stiffness for each voxel. This procedure can be computationally expensive for some material laws. Furthermore, the analytic derivation of the tangent can be tedious and its implementation may require considerable programming effort, and is thus prone to errors. This gave rise to applying Quasi-Newton methods in FFT-based micromechanics. Quasi-Newton schemes rely upon an approximation of the Hessian by generalizing the one-dimensional secant method and are thereby tangent free.[24] Schneider[19] used the Barzilai-Borwein method,[25] which approximates the Hessian by a multiple of the identity, to accelerate Moulinec-Suquet's basic scheme. Shantraj et al[3] pioneered using Anderson acceleration[26] in an FFT-based context. The algorithm is included in the software DAMASK[27] as the nonlinear GMRES method. More recently, Chen et al successfully adapted the Anderson acceleration to simulate damage initiation[28] and brittle fracture.[10] Originally developed to accelerate general fixed-point iterations, Anderson acceleration was linked to Quasi-Newton schemes by Fang and Saad.[29] More precisely, it was identified as a generalized multisecant form of the second Broyden method (or "bad Broyden method")[30] which approximates the Hessian in terms of a number m (called depth) of past iterates and gradients. Recently, Evans et al[31] proved that Anderson acceleration improved the first-order convergence rate for fixed-point iterations. Pollock and Rebholz[32] extended the analysis to the noncontractive setting and provided sharper residual bounds.

Motivated by the mentioned work on Quasi-Newton methods, we focus our attention on the powerful and popular Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm.[33-36] We revisit its basics in the framework of (inexact) Newton methods in Section 2. Both Newton and Quasi-Newton methods require appropriate globalization strategies to ensure global convergence. Often, this is realized by a backtracking line search using appropriate conditions for the acceptance of the step size. However, applying the classical Wolfe conditions[37] to FFT-based micromechanics is not feasible, as function evaluations are not available in this setting in general, as the condensed potential[38] of the material law carries no physical meaning and is therefore not computed. Hence, we propose using the line-search conditions proposed by Dong,[4] which solely rely upon gradient evaluations, cf. Section 2.3. Another aspect which is of major importance for the overall performance of inexact (Quasi-)Newton methods is the choice of the forcing term, that is, the accuracy to which the linear system is solved. To this end, we revisit the forcing-term strategies of Eisenstat and Walker,[39] cf. Section 2.4. In Section 3, we turn our attention to Newton and Quasi-Newton methods as applied in the context of FFT-based micromechanics. After revisiting the Newton-CG method and Anderson acceleration, two possible uses of the BFGS update formula in the FFT-based setting are proposed. First, we investigate the limited-memory version of the BFGS algorithm (L-BFGS) by Nocedal[40] which only stores the m last differences of iterates and gradients for its Hessian, similar to Anderson acceleration. A second algorithm is derived, using the BFGS-update formula to approximate the local material tangent for every voxel instead of the Hessian of the global system. In analogy to the Newton-CG method, the resulting linear system is solved using CGs. Hence, we refer to the method as BFGS-CG. Last but not least, we compare the performance of the investigated solution algorithms and the impact of the different forcing-term choices for nonlinear problems with finite and infinite material contrast, cf. Section 4.

2 | NEWTON AND QUASI-NEWTON METHODS

2.1 | Newton's method

Let V be a Hilbert space with an associated inner product $V \times V \to \mathbb{R}$, $(x, y) \mapsto \langle x, y\rangle_V$ and the induced norm $\|x\|_V = \sqrt{\langle x, x\rangle_V}$. Suppose a twice continuously differentiable function $f : V \to \mathbb{R}$ is given. Its gradient $\nabla f : V \to V$ is defined by

$$Df(x)[v] = \langle \nabla f(x), v\rangle_V, \qquad v \in V, \tag{1}$$

where $Df : V \to V'$ denotes the differential of f and V' is the continuous dual of V. For a minimization problem

$$f(x) \to \min_{x \in V}, \tag{2}$$

critical points of f are characterized by

$$\nabla f(x) = 0. \tag{3}$$

Newton's method iteratively updates an initial guess $x_0 \in V$ by the formula

$$x_{n+1} = x_n + \xi_n, \quad \text{where} \quad \xi_n \in V \quad \text{solves} \quad D\nabla f(x_n)[\xi_n] = -\nabla f(x_n), \tag{4}$$

where $D\nabla f : V \to L(V, V)$ denotes the Hessian of f and and $L(V, V)$ denotes the space of linear mappings $V \to V$. Let $x^* \in X$ be a solution to Equation (3). Suppose that $D\nabla f(x^*)$ is an isomorphism and $D\nabla f$ is Lipschitz continuous in a neighborhood of x^*. Then, if x_0 is sufficiently close to x^*, the Newton iteration (4) converges, and if $D\nabla f$ is locally Lipschitz, it does so with quadratic rate.[41]

To obtain global convergence, the Newton iteration (4) has to be modified, for instance, by damping, that is, with $\alpha_n \in (0, 1]$,

$$x_{n+1} = x_n + \alpha_n \xi_n, \quad \text{where} \quad \xi_n \in V \quad \text{solves} \quad D\nabla f(x_n)[\xi_n] = -\nabla f(x_n). \tag{5}$$

The damping factor α_n is chosen by a line-search procedure, for instance, by an approximate line search involving the Wolfe conditions[37]

$$f(x_n + \alpha_n \xi_n) \le f(x_n) + c_1\ \alpha_n \langle \nabla f(x_n), \xi_n\rangle_V, \tag{6}$$

and

$$\langle \nabla f(x_n + \alpha_n \xi_n), \xi_n\rangle_V \ge c_2 \langle \nabla f(x_n), \xi_n\rangle_V, \tag{7}$$

for fixed constants $0 < c_1 < c_2 < 1$.

For large-scale applications, the equation $D\nabla f(x_n)[\xi_n] = -\nabla f(x_n)$ for the Newton increment can often only be solved iteratively up to a prescribed precision, leading to an inexact, damped Newton method

$$x_{n+1} = x_n + \alpha_n \xi_n, \quad \text{where} \quad \xi_n \in V \quad \text{solves} \quad \|D\nabla f(x_n)[\xi_n] + \nabla f(x_n)\|_V \le \gamma_n \|\nabla f(x_n)\|_V. \tag{8}$$

The choice of γ_n is crucial, as its order of convergence (as $n \to \infty$) is linked to the convergence of x_n to x^*, see Dembo et al.[42] More precisely, if γ_n is uniformly less than one, x_n converges to x^* linearly. Furthermore, assuming Lipschitz continuity of $D\nabla f(x_n)$ in a neighborhood of x^*, $\gamma_n \le C\|x_n - x^*\|_V$ is necessary to obtain quadratic convergence. However, "asymptotic quadratic convergence is achievable, but only with effort on the part of the inner, linear iterative method, which is usually unwarranted when overall time to solution is the metric," cf. Knoll and Keyes.[43] General-purpose strategies for the choice of γ_n were proposed by Eisenstat and Walker[39] and are discussed in Section 2.4.

Despite the computational power of Newton's method, there are several practical disadvantages.

1. Programming the second derivatives of a function can be tedious, and doing it efficiently is often challenging. These problems can be partly overcome by automatic differentiation techniques.[44]
2. If V is m-dimensional and the equation for the Newton increment is solved directly, $O(m^3)$ operations are required. For large m, this can be excessive. If the Hessian is sparse, iterative solvers can be used to reduce the computational complexity to $O(m^2)$.
3. For inexact Newton methods, the optimal choice of the Newton forcing term $\{\gamma_n\}$ in Equation (8) is difficult. Although general-purpose strategies have been developed,[39] the following problem remains. Suppose you wish to find a δ-critical point, that is, to find a solution to the inequality

$$\|\nabla f(x)\|_V \leq \delta,$$

and your current iterate x_n almost satisfies the inequality. How accurate do you have to solve for the increment to ensure that x_{n+1} is δ-critical?

Points 1 and 3 motivated the development of Quasi-Newton methods which we shall discuss next.

2.2 | From Newton to BFGS

Quasi-Newton methods replace the Hessian $D\nabla f(x_n)$ in the linear equation

$$D\nabla f(x_n)[\xi_n] = -\nabla f(x_n), \tag{9}$$

by an approximation B_n which is required to fulfill the secant condition

$$y_n = B_{n+1}s_n, \quad \text{where} \quad s_n = x_{n+1} - x_n, \quad \text{and} \quad y_n = \nabla f(x_{n+1}) - \nabla f(x_n). \tag{10}$$

Among the most powerful Quasi-Newton methods is the BFGS algorithm,[33-36] which recursively updates an approximation of the Hessian

$$B_{n+1} = B_n + \frac{y_n \otimes \langle y_n, \cdot\rangle_V}{\langle y_n, s_n\rangle_V} - \frac{B_n s_n \otimes \langle B_n s_n, \cdot\rangle_V}{\langle s_n, B_n s_n\rangle_V}, \tag{11}$$

for a given $B_0 \in L(V, V)$. If the operator B_0 is self-adjoint and positive definite, the subsequent $B_n \in L(V, V)$ will inherit the symmetry and positive definiteness property. Alternatively, an update formula corresponding to Equation (11) is available for the inverse of the Hessian $H_n = B_n^{-1}$

$$H_{n+1} = \left(\mathrm{Id} - \frac{y_n \otimes \langle s_n, \cdot\rangle_V}{\langle y_n, s_n\rangle_V}\right) H_n \left(\mathrm{Id} - \frac{s_n \otimes \langle y_n, \cdot\rangle_V}{\langle y_n, s_n\rangle_V}\right) + \frac{s_n \otimes \langle s_n, \cdot\rangle_V}{\langle y_n, s_n\rangle_V}. \tag{12}$$

With this formula at hand, $\xi_n = -H_n\nabla f(x_n)$ can be computed without solving the linear system (9). Thus, the damped BFGS method may be rewritten

$$x_{n+1} = x_n - \alpha_n H_n \nabla f(x_n). \tag{13}$$

Global superlinear convergence of the BFGS method (13) with inexact line search respecting the Wolfe conditions (6) and (7) and uniformly convex and Lipschitz-continuous objective functions in finite dimensions has been established by Powell.[45] In the general Hilbert space setting, only linear convergence[46,47] can be expected, see Griewank[47] for counterexamples. If the Hessian at the critical point x^* and the inverse H_0^{-1} of the initial approximation of the Hessian differ by a compact linear operator, superlinear convergence can be established.[47] More generally, superlinear convergence is characterized by Dennis and Moré.[48] However, their criterion is difficult to verify for a particular problem at hand.

The BFGS method still keeps the Hessian (or its inverse) in memory. In particular, due to the rank-two update, B_n quickly becomes fully populated, restricting the method's utility for large-scale applications. Nocedal[40] introduced a

limited-memory version of BFGS (L-BFGS) depending on a positive integer m, such that only the last m differences of iterates s_n and gradients y_n are kept in storage for updating the inverse Hessian. More precisely, for any n, and $l = 0, \dots, m-1$, Nocedal proposed the formula

$$H_n^{m-l} = \left(\mathrm{Id} - \frac{y_{n-l} \otimes \langle s_{n-l}, \cdot \rangle_V}{\langle y_{n-l}, s_{n-l} \rangle_V}\right) H_n^{m-l-1} \left(\mathrm{Id} - \frac{s_{n-l} \otimes \langle y_{n-l}, \cdot \rangle_V}{\langle y_{n-l}, s_{n-l} \rangle_V}\right) + \frac{s_{n-l} \otimes \langle s_{n-l}, \cdot \rangle_V}{\langle y_{n-l}, s_{n-l} \rangle_V}, \tag{14}$$

for some initial approximation H_n^0, and where we formally set y_n and s_n to zero for $n < 0$. Typically, the initial approximation is chosen as a multiple of the identity $H_n^0 = \theta_n \mathrm{Id}$. A common choice for the scaling factor is given by $\theta_n = \langle s_{n-1}, y_{n-1} \rangle_V / \langle y_{n-1}, y_{n-1} \rangle_V$, cf. Shanno and Puah[49] and Liu and Nocedal,[50] corresponding to the Barzilai-Borwein stepsize.[25] The damped L-BFGS iteration reads

$$x_{n+1} = x_n - \alpha_n H_n^m \nabla f(x_n). \tag{15}$$

How to implement the update (15) in the context of FFT-based micromechanics is discussed in Section 3.3. For strongly convex and Lipschitz-continuous objective functions, convergence of L-BFGS under the Wolfe conditions (6) and (7) in finite dimensions V was established by Liu and Nocedal.[50] In contrast to BFGS, the convergence to x^* is only linear.

2.3 | The line-search procedure of Dong

Global convergence of Newton's method and (L-)BFGS depends on a flexible line-search procedure. Exact line search is typically infeasible in practice, because evaluating the gradient of the objective function involves nonlinear, and often quite costly, operations. Thus, approximate line-search procedures ensuring sufficient decrease per iteration are mandatory, involving, for instance, the Wolfe conditions (6) and (7). In particular, using the Wolfe conditions as criterion for the line search is crucial for ensuring global convergence of the (L-)BFGS method. Satisfying the Wolfe conditions guarantees that the curvature condition

$$\langle y_n, s_n \rangle_V > 0,$$

holds, which is necessary for the positive definiteness of the iterates B_n, cf. Section 6.1 in Nocedal and Wright.[24]

For FFT-based micromechanics (to be discussed in Section 3), function evaluations are not available, in general. The reason is that, in contrast to the stress, the Helmholtz free energy or the dissipation potential, the condensed potential f for the nonlinear material law, relating strains and stresses, has no physical meaning (because it depends on the time discretization and mixes the Helmholtz free energy and the dissipation potential). In particular, the Wolfe condition (6) cannot be evaluated per se. As a workaround, Dong[4] proposed to replace the first Wolfe condition (6) by the inequality

$$\langle \nabla f(x_n + \alpha_n d_n), d_n \rangle_V \leq c_1 \langle \nabla f(x_n), d_n \rangle_V, \tag{16}$$

which implies Equation (6) if the gradient $\nabla f : X \to X$ is monotone, that is, it satisfies

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle_V \geq 0, \quad x, y \in V.$$

In mechanics, the latter is equivalent to the monotonicity of the stress, considered as a function of the strain.

2.4 | Strategies for choosing the forcing term

For inexact Newton-methods, the choice of the forcing term $\{\gamma_n\}$ in Equation (8) is crucial for the overall efficiency of the scheme. At iterates $\{x_n\}$ far away from the solution, ∇f and its linear approximation may disagree significantly. Thus, solving the linear system (9) to a high accuracy may waste computational effort without substantially improving the overall convergence behavior.[39] This is commonly called oversolving. Setting γ_n to a moderate constant value, for example,

$\gamma_n = 0.1$ as suggested by Kelly,[51] can be reasonable but may not be optimal for all problems. Eisenstat and Walker[39] propose more involved strategies, taking ∇f into account. Their first strategy, named choice 1, reads

$$\gamma_n = \left| \frac{\|\nabla f(x_n)\|_V - \|D\nabla f(x_{n-1})[\xi_{n-1}] + \nabla f(x_{n-1})\|_V}{\|\nabla f(x_{n-1})\|_V} \right|, \tag{17}$$

with an initial value $\gamma_0 \in [0, 1)$. This choice directly measures the disagreement between the gradient and its linear approximation. Thus, the value of γ_n decreases, as the Newton iterates $\{x_n\}$ approach the solution of the system. Eisenstat and Walker's[39] alternative choice 2 is given by

$$\gamma_n = \lambda \left(\frac{\|\nabla f(x_n)\|_V}{\|\nabla f(x_{n-1})\|_V} \right)^{\alpha}, \tag{18}$$

with parameters $\lambda \in [0, 1]$ and $\alpha \in (1, 2]$. The ratio of consecutive residua provides a measure for the convergence rate between the current and last iteration. Hence, close to the solution, where a faster convergence behavior is expected, γ_n decreases. Setting the parameter $\alpha = \frac{1+\sqrt{5}}{2}$ results in a comparable convergence order for choices 1 and 2. In addition, Eisenstat and Walker suggest a safeguard for each choice to prevent a premature decrease of γ_n far away from the solution. This is achieved by limiting the decrease of γ_n by a factor of γ_{n-1} above a certain threshold. The safeguard for choice 1 reads

$$\gamma_n^{\text{safe}} = \begin{cases} \max\left(\gamma_n, \gamma_{n-1}^{(1+\sqrt{5})/2}\right), & \text{if } \gamma_{n-1}^{(1+\sqrt{5})/2} > 0.1, \\ \gamma_n, & \text{otherwise,} \end{cases} \tag{19}$$

and the safeguard for choice 2 is given by

$$\gamma_n^{\text{safe}} = \begin{cases} \max\left(\gamma_n, \lambda\gamma_{n-1}^{\alpha}\right), & \text{if } \lambda\gamma_{n-1}^{\alpha} > 0.1, \\ \gamma_n, & \text{otherwise.} \end{cases} \tag{20}$$

Even with the presented forcing-term choices and safeguards in place, oversolving may occur in the final Newton iteration. Indeed, suppose we want to solve Equation (3) up to a certain accuracy

$$\|\nabla f(x)\|_V \leq \delta, \tag{21}$$

and the current iterate x_n almost satisfies Equation (21). With a small value for γ_n, the final Newton iteration may reduce $\|\nabla f(x)\|_V$ far below the desired accuracy δ. To prevent this type of oversolving, the following safeguard

$$\gamma_n^{\text{final}} = \min(\gamma_{\max}, \max(\gamma_n^{\text{safe}}, 0.5\ \delta/\|\nabla f(x)\|_V)) \tag{22}$$

with $\gamma_{\max} \in [0, 1)$ is suggested in Section 6.3 in Kelley's book.[52]

3 | NEWTON AND QUASI-NEWTON METHODS IN FFT-BASED MICROMECHANICS

3.1 | Newton's method

We consider periodic homogenization problems[53] in the context of small-strain continuum mechanics. Let Y be a rectangular cell in $\mathbb{R}^d$ ($d = 2, 3$). The Hilbert space under consideration is

$$H^1_{\#}(Y; \mathbb{R}^d) = \left\{ u \in H^1(Y; \mathbb{R}^d) \mid u \text{ periodic}, \ \partial_n u \text{ antiperiodic on } \partial Y, \ \langle u \rangle_Y = 0 \right\},$$

where the mean of any integrable scalar or vector valued function q on Y is defined by

$$\langle q \rangle_Y = \frac{1}{|Y|} \int_Y q(x)\, dx,$$

together with the inner product induced by the quadratic form

$$\|u\|^2_{H^1_\#(Y;\mathbb{R}^d)} = \frac{1}{|Y|} \int_Y \|\nabla^s u\|^2\, dx,$$

where ∇^s denotes the symmetrized gradient and the quadratic form in the integrand corresponds to the Frobenian inner product on square matrices, $\|S\|^2 = \mathrm{tr}(S^T S)$.

Furthermore, let a (heterogeneous) strain energy potential

$$w : Y \times \mathrm{Sym}(d) \to \mathbb{R}, \quad (x, \varepsilon) \mapsto w(x, \varepsilon),$$

be given, measurable in Y and C^2 in $\mathrm{Sym}(d)$, where $\mathrm{Sym}(d)$ denotes the linear space of symmetric $d \times d$-matrices. Denote by $\sigma = \frac{\partial w}{\partial \varepsilon}$ the associated stress function, and by $\frac{\partial^2 w}{\partial \varepsilon^2}$ its Hessian. For prescribed strain E, we seek a minimizer of the function

$$H^1_\#(Y;\mathbb{R}^d) \ni u \mapsto f(u) = \langle w(\cdot, E + \nabla^s u) \rangle_Y. \tag{23}$$

To conform to the framework of the previous section, we compute the differential of f

$$Df(u) = -\mathrm{div}\ \sigma(\cdot, E + \nabla^s u),$$

and its gradient

$$\nabla f(u) = G\ \mathrm{div}\ \sigma(\cdot, E + \nabla^s u),$$

where G is the Green's operator $G = (\mathrm{div}\nabla^s)^{-1}$, which corresponds to the negative of the Riesz map on $H^1_\#(Y;\mathbb{R}^d)$.

In this context, the equation for the nth Newton increment $\xi_n \in H^1_\#(Y;\mathbb{R}^d)$, corresponding to Equation (9), is given by

$$G\mathrm{div}\ \left[\frac{\partial^2 w}{\partial \varepsilon^2}(\varepsilon_n) : \nabla^s \xi_n\right] = -G\mathrm{div}\ \sigma(\varepsilon_n), \tag{24}$$

where $\varepsilon_n = E + \nabla^s u_n$. For any $\beta^0 > 0$, Equation (24) is equivalent to the Lippmann-Schwinger equation

$$\Xi_n + \Gamma^0 : \left[\frac{\partial^2 w}{\partial \varepsilon^2}(\varepsilon_n) - \mathbb{C}^0\right] : \Xi_n = -\Gamma^0 : \sigma(\varepsilon_n), \tag{25}$$

where $\mathbb{C}^0 = \beta^0 \mathrm{Id}$, $\Gamma^0 = (\beta^0)^{-1} \nabla^s G \mathrm{div}$, via the identification $\Xi_n = \nabla^s \xi_n$. Note, if a strain-based iterative scheme is used to solve Equation (25), only the converged solution Ξ^* is compatible, in general, whereas this may be false for the iterates $\{\Xi_n\}$. This is the case, for instance, for polarization-based schemes as the Eyre-Milton method used by Kabel et al.[2] Typically, Equation (25) is solved using Krylov-subspace methods, such as CG or MINRES,[20-22] due to their excellent performance for linear problems. In addition, these schemes operate on compatible strain-fields, permitting a memory efficient implementation.[2] With these formulae at hand, we may formulate a damped Newton scheme, depending on Dong's version of the Wolfe conditions, Equations (16) and (7). The resulting algorithm is summarized in Algorithm 1.

Algorithm 1 Newton's method with backtracking by Dong (E, $\mathbb{C}^0$, $c_{1,0}$, c_2, maxiter)[4]

$\varepsilon \leftarrow E$
$\varepsilon \leftarrow \text{MSiterate}\ (\varepsilon, E, \mathbb{C}^0)$
repeat
  $\Xi \leftarrow -\left(\text{Id}+\Gamma^0 : \left[\frac{\partial^2 w}{\partial \varepsilon^2}(\varepsilon) - \mathbb{C}^0\right]\right)^{-1} : \Gamma^0 : \sigma(\varepsilon)$ ▷ Solving Equation (25), cf. Kabel et al[2]
  $\mu \leftarrow 0$
  $\nu \leftarrow +\infty$
  $\alpha \leftarrow 1$
  $k \leftarrow 0$
  while $k <$ maxiter **do**
    $k \leftarrow k + 1$
    $c_1 \leftarrow c_{1,0}(1 - (c_2)^k) - (c_2)^k$ ▷ Typical parameters for Equations (16) and (7): $c_{1,0} = 10^{-4}$, $c_2 = 0.9$, cf. Dong[4]
    if $\langle \Gamma^0 : \sigma(\varepsilon + \alpha\Xi), \Xi \rangle_{L^2} > c_1 \langle \Gamma^0 : \sigma(\varepsilon), \Xi \rangle_{L^2}$ **then**
      $\nu \leftarrow \alpha$
      $\alpha \leftarrow 0.5(\mu + \nu)$
    else if $\langle \Gamma^0 : \sigma(\varepsilon + \alpha\Xi), \Xi \rangle_{L^2} < c_2 \langle \Gamma^0 : \sigma(\varepsilon), \Xi \rangle_{L^2}$ **then**
      $\mu \leftarrow \alpha$
      $\alpha \leftarrow 2\mu$
    else
      break
    end if
  end while
  $\varepsilon \leftarrow \varepsilon + \alpha\Xi$
until Convergence ▷ Criterion (26)
return ε

MSiterate (ε, E, $\mathbb{C}^0$)
$\varepsilon \leftarrow \sigma(\varepsilon) - \mathbb{C}^0 : \varepsilon$
$\varepsilon \leftarrow \text{FFT}(\varepsilon)$
$\varepsilon \leftarrow -\Gamma^0 : \varepsilon, \quad \varepsilon(0) = E$
$\varepsilon \leftarrow \text{FFT}^{-1}(\varepsilon)$
return ε

The convergence criterion reads

$$\beta^0 \frac{\|\Gamma^0 : \sigma^k\|_{L^2}}{\|\langle \sigma^k \rangle_Y\|} \equiv \frac{\|\text{div}\,(\sigma^k)\|_{H^{-1}}}{\|\langle \sigma^k \rangle_Y\|} \leq \delta, \tag{26}$$

with a prescribed tolerance δ. This choice was introduced and discussed in Schneider et al.[18] Both, the convergence criterion (26) and the convergence behavior of the linear Krylov-subspace solver are independent of β^0, cf. Zeman et al.[20] As we start with a single iteration of the basic scheme, we use the associated reference material $\beta^0 = (\beta^+ + \beta^-)/2$ with the extremal eigenvalues β^+ and β^- of the material tangent evaluated over all voxels. For the parameters of the line-search procedure, we choose $c_{1,0} = 10^{-4}$ and $c_2 = 0.9$, cf. Dong.[4]

A few remarks on the practical implementation are in order.

1. The storage requirements for Newton-CG read: one current strain, and four strains for solving the linear system by CG. Furthermore, the symmetric material tangent needs to be stored. In three spatial dimensions, this corresponds to 21 scalar components for every voxel, the equivalent of 3.5 strain fields. In total, the storage requirements amount to 8.5 strain-like fields. Using the line-search procedure by Dong[4] involves storing another strain field, as gradient and Newton step have to be kept in memory separately. If affine-linear extrapolation is needed, an additional strain needs to be stored.

2. We have found out that storing the Hessian in single precision does not influence the performance of Newton's method significantly. In contrast, the current strain, and the vectors needed for CG need to be stored in double precision to avoid numerical problems (in particular, in connection to the FFT).
3. Similar to the previous comment, the last converged strain can be stored in single precision, as it solely serves as the initial condition. This remark holds true for other solution methods in FFT-based micromechanics, as well.
4. For finite-difference and finite-element discretizations,[54-57] both the CG method and the Newton update can be implemented on displacement instead of strain,[2,58] saving 50% of memory for the corresponding fields.
5. Combining all three previous memory optimizations, only nine displacement fields need to be stored. For a microstructure with 512^3 voxels, 27 GBs RAM are needed, not taking into account internal variables.

3.2 | Anderson acceleration

The BFGS method as outlined in Section 2.2 requires the Hessian B_n (or its inverse) to be kept in memory. Thereby, the algorithm cannot be directly applied in the context of FFT-based micromechanics, as the Hessian is usually not assembled in this setting due to memory limitations. To circumvent this problem, limited-memory Quasi-Newton methods were developed, which implicitly update the Hessian by storing a limited number m of recent iterates and gradients, with m commonly called the depth of the scheme.

One such algorithm is Anderson acceleration[26] which was recently applied by Shantraj et al[3] and Chen et al.[10,28] in the context of FFT-based micromechanics. A general discussion of the scheme and its implementation is found, for example, in Walker and Ni[59] or Kelley.[51] Eyert[60] and Fang and Saad[29] pointed out the relation of Anderson acceleration to Quasi-Newton schemes and identified it as a generalized form of Broyden's second method. Recently, Evans et al[31] provided a proof that Anderson acceleration improves the convergence rate of linearly converging fixed-point methods.

For an integer depth $m \geq 1$, Anderson acceleration requires the last $m+1$ iterates ε_k and gradients $g_k = \Gamma^0 : \sigma(\varepsilon_k)$ to be kept in memory, resulting in a memory footprint of $2m+2$ strain-like fields. The algorithm is outlined in Algorithm 2 for the convenience of the reader. Note that for the given algorithm Anderson acceleration is applied for every iteration. In contrast, Chen et al[10,28] only accelerate every third iteration and apply Moulinec-Suquet's basic scheme[6] otherwise.

Algorithm 2 Anderson acceleration (E, $\mathbb{C}^0$)

$\varepsilon_0 \leftarrow E$
$\varepsilon_1 \leftarrow \text{MSiterate}\,(\varepsilon_0, E, \mathbb{C}^0)$
$k \leftarrow 0$
repeat
  $k \leftarrow k+1$
  $m_k \leftarrow \min(m, k)$
  $g_k \leftarrow \sigma(\varepsilon_k)$
  $g_k \leftarrow \text{FFT}(g_k)$
  $g_k \leftarrow \Gamma^0 : g_k, \quad g_k(0) = 0$
  $g_k \leftarrow \text{FFT}^{-1}(g_k)$
  $(\alpha_0, \dots, \alpha_{m_k}) \leftarrow \min \| \sum_{j=0}^{m_k} \alpha_j g_{k-m_k+j} \|_{L^2} \quad \text{s.t.} \quad \sum_{j=0}^{m_k} \alpha_j = 1$ ▷ cf. Solving Equation (29)
  $\varepsilon_{k+1} = \sum_{j=0}^{m_k} \alpha_j (\varepsilon_{k-m_k+j} - g_{k-m_k+j})$
  Delete ε_{k-m_k}, g_{k-m_k}
until Convergence ▷ Criterion (26)
return ε_{k+1}

Determining the coefficients $\alpha = (\alpha_0, \dots, \alpha_{m_k})$ by solving the minimization problem

$$\min_{\alpha} \left\| \sum_{j=0}^{m_k} \alpha_j g_{k-m_k+j} \right\|_{L^2} \quad \text{s.t.} \quad \sum_{j=0}^{m_k} \alpha_j = 1, \tag{27}$$

is the key step in one iteration of the Anderson acceleration. To solve this problem, we reformulate Problem (27) in terms of the Lagrangian function

$$\sum_{i=0}^{m_k}\sum_{j=0}^{m_k}\frac{1}{2}\alpha_i\alpha_j\langle g_{k-m_k+i}, g_{k-m_k+j}\rangle_{L^2} + \lambda\left(\sum_{j=0}^{m_k}\alpha_j - 1\right) \to \min_{\alpha}\max_{\lambda}, \tag{28}$$

by squaring the objective function and introducing the Lagrangian multiplier λ. The associated KKT-conditions

$$\begin{aligned}
\sum_{j=0}^{m_k}\alpha_j\langle g_{k-m_k}, g_{k-m_k+j}\rangle_{L^2} + \lambda &= 0,\\
&\vdots\\
\sum_{j=0}^{m_k}\alpha_j\langle g_{k}, g_{k-m_k+j}\rangle_{L^2} + \lambda &= 0,\\
\sum_{j=0}^{m_k}\alpha_j - 1 &= 0,
\end{aligned} \tag{29}$$

constitute a system of $m_k + 2$ linear equations, which are solved for α and λ.

3.3 | Limited-memory BFGS

As another limited-memory Quasi-Newton scheme, we propose to apply of Nocedal's L-BFGS method, cf. Section 2.2, to FFT-based micromechanics. The L-BFGS method can be implemented with a memory-footprint of $2m + 4$ strain-like fields. More precisely, the last m differences of iterates $s_k = \varepsilon_{k+1} - \varepsilon_k$, differences of gradients $y_k = \Gamma^0 : \sigma(\varepsilon_{k+1}) - \Gamma^0 : \sigma(\varepsilon_k)$ and inner products $\rho_k = 1/\langle y_k, s_k\rangle_{L^2}$ have to be kept in memory. In addition, the current strain ε and gradient $\Gamma^0 : \sigma(\varepsilon)$ and the last strain ε_n and gradient $\Gamma^0 : \sigma(\varepsilon_n)$ need to be stored.

For evaluating the L-BFGS increment $\Xi = -H_n^m\nabla f(x_n)$, the two-loop recursion of Matthies and Strang[61] proves useful. A pseudo code is given in Algorithm 3, where we use the initial Hessian $H_n^0 = \frac{\langle s_{n-1}, y_{n-1}\rangle_{L^2}}{\langle y_{n-1}, y_{n-1}\rangle_{L^2}}\mathrm{Id}$, as suggested by Shanno and Puah[49] and Nocedal and Liu.[50] The algorithm takes the current gradient $\Gamma^0 : \sigma(\varepsilon_k)$ as input and overwrites it by the increment Ξ_k.

Algorithm 3 Two-loop recursion for the evaluation of $H_n^m q$ for given q[40,61]

for $k = m-1, m-2, \dots, 0$ **do**
  $a_k \leftarrow \rho_k\langle s_k, q\rangle_{L^2}$
  $q \leftarrow q - a_k y_k$
end for
$q \leftarrow \frac{\langle s_{n-1}, y_{n-1}\rangle_{L^2}}{\langle y_{n-1}, y_{n-1}\rangle_{L^2}} q$
for $k = 0, 1 \dots, m-1$ **do**
  $b_k \leftarrow \rho_k\langle y_k, q\rangle_{L^2}$
  $q \leftarrow q + (a_k - b_k)s_k$
end for
return q

The L-BFGS method is implemented analogously to Algorithm 1, where the two-loop recursion replaces the solution of the linear system (25) for obtaining Ξ.

3.4 | BFGS update of the material tangent

As an alternative to the limited-memory Quasi-Newton scheme, we propose using the BFGS update to approximate the local material tangent $\frac{\partial^2 w}{\partial \varepsilon^2}$ in Equation (25) instead of the global Hessian of f in Equation (23). In this context, the BFGS update reads

$$\mathbb{C}_{n+1}^{\mathrm{BFGS}} = \mathbb{C}_n^{\mathrm{BFGS}} + \frac{\Delta\sigma_n \otimes \Delta\sigma_n}{\Delta\sigma_n : \Delta\varepsilon_n} - \frac{(\mathbb{C}_n^{\mathrm{BFGS}} : \Delta\varepsilon_n) \otimes (\mathbb{C}_n^{\mathrm{BFGS}} : \Delta\varepsilon_n)}{\Delta\varepsilon_n : \mathbb{C}_n^{\mathrm{BFGS}} : \Delta\varepsilon_n}, \tag{30}$$

where

$$\Delta\varepsilon_n = \varepsilon_{n+1} - \varepsilon_n \quad \text{and} \quad \Delta\sigma_n = \sigma(\varepsilon_{n+1}) - \sigma(\varepsilon_n).$$

We found that the material's linear elastic stiffness serves as a decent initial guess for $\mathbb{C}_0^{\mathrm{BFGS}}$. Consequently, Algorithm 1 may be applied with $\mathbb{C}_n^{\mathrm{BFGS}}$ replacing $\frac{\partial^2 w}{\partial \varepsilon^2}(\varepsilon_n)$ in (25). Note that, in contrast to the limited-memory schemes in Sections 3.2 and 3.3, the linear system (25) still needs to be solved with an iterative solver. In comparison to the Newton-CG method, two additional strain-like fields need to be kept in memory to compute $\Delta\sigma_n$.

4 | NUMERICAL DEMONSTRATIONS

4.1 | General setup

The solution schemes were implemented in Python 2.7. Computationally expensive operations such as the application of Γ^0 and the evaluation of the material law were written as Cython extensions and parallelized using OpenMP. For the FFT, we relied on the FFTW library.[62] The computations ran on six threads on a desktop computer with 32 GB RAM and an Intel i7-8700K CPU with six cores and a clock rate of 3.7 GHz. An affine-linear extrapolation[6] was used as initial guess for the strain field in case of multiple load steps. For the convergence criterion, we use inequality (26)

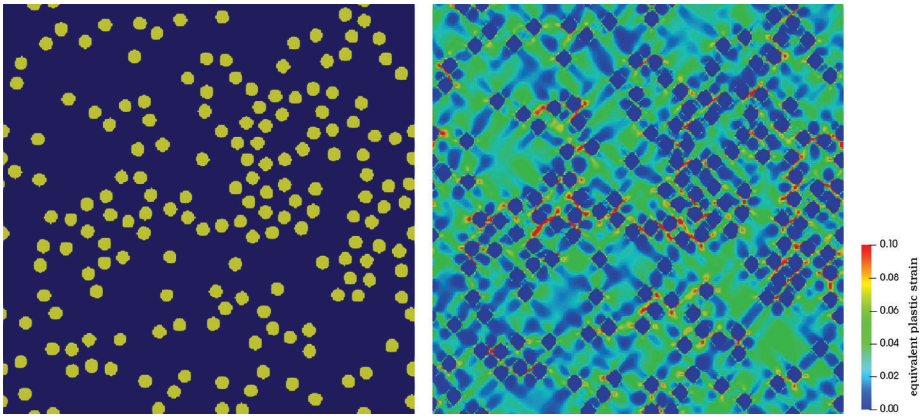
$$\beta^0 \frac{\|\Gamma^0 : \sigma^k\|_{L^2}}{\|\langle \sigma^k \rangle_Y\|} \leq \delta,$$

where β^0 is the scaling factor of the reference material $\mathbb{C}^0 = \beta^0 \mathrm{Id}$. As $\Gamma^0 = (\beta^0)^{-1} \nabla^s G \mathrm{div}$, this convergence criterion is actually independent of β^0. For this study, we use the reference material of the basic scheme $\beta^0 = (\beta^+ + \beta^-)/2$. The tolerance is set to $\delta = 10^{-5}$ in Section 4.2 and $\delta = 10^{-4}$ in Sections 4.3 and 4.4. Throughout, we utilize the staggered grid discretization.[55]

4.2 | Continuous glass-fiber reinforced polyamide

In the following, we investigate the performance of the L-BFGS method and Anderson acceleration as discussed in Sections 3.2 and 3.3 with respect to the chosen depth m. As microstructure we consider a polyamide matrix, reinforced by continuous glass fibers with a volume fraction of 15%, and a resolution of 256^2 pixels, cf. Figure 1. Using a two-dimensional structure enables investigating large values of the depth m, without memory becoming a limiting factor. Following Doghri et al,[63] we assume that the mechanical behavior of the polyamide matrix is governed by J_2-elastoplasticity, cf. Section 3.3 in Simo and Hughes[64]. For the sake of simplicity, the rate-dependent behavior of the material is neglected in this approach. A more involved material model, accounting for viscoelastic and viscoplastic effects was proposed, for example, by Krairi et al.[65] The relation between the yield stress σ_Y and the equivalent plastic strain $p = \int_0^t \sqrt{\frac{2}{3}} \|\dot{\varepsilon}_{\mathrm{p}}\| \, \mathrm{d}\hat{t}$ is modelled by a linear-exponential hardening function

$$\sigma_Y(p) = \sigma_0 + k_1 p + k_2(1 - \exp(-mp)),$$



(A) Microstructure (256^2 pixels)

(B) Equivalent plastic strain at 5% uniaxial extension

FIGURE 1 Continuous glass-fiber reinforced polyamide. (A) Microstructure (256^2 pixels) and (B) equivalent plastic strain at 5% uniaxial extension

where σ_0 denotes the initial yield strength, k_1 denotes the asymptotic hardening modulus, and $k_2 = \sigma_0 - \sigma_\infty$ denotes the difference between the initial and saturated yield strength for $k_1 = 0$. The prefactor in the exponential function is given by $m = \Theta/k_2$, where Θ denotes the initial hardening modulus. The glass fibers are modelled as linear elastic. The material parameters according to Doghri et al[63] are given in Table 1. We apply mixed boundary conditions,[66] corresponding to a uniaxial extension of 5% perpendicular to the fiber direction, in a single load step.

The L-BFGS scheme and Anderson acceleration are investigated for depths from 1 to 200. In addition, Moulinec-Suquet's basic scheme,[6] the basic scheme with Barzilai-Borwein step-size control,[19,25] the Newton-CG method and the BFGS-CG method are included as benchmarks. For the Newton-CG method and the BFGS-CG method, we use forcing-term choice 2 of Eisenstat and Walker Equation (18), cf. Section 4.3.1. The resulting iteration counts and the computational runtimes are given, depending on the depth, in Figure 2 and Table 2.

For Anderson acceleration, we observe that the required number of iterations drops significantly up to a depth of 5 and stagnates for depths larger than 50. Between the minimum depth of 1 and a depth of 200, that is, keeping all iterates in memory, the iteration count decreases by 85%. In contrast, the convergence behavior of L-BFGS is much less affected by the chosen depth. From the onset, it requires much fewer iterations than Anderson acceleration and exhibits a faster convergence behavior up to depths of 20. For depths larger than 5, the iteration counts of L-BFGS remain approximately constant with a decrease of about 20% compared with a depth of 1.

Considering the overall computational effort, depths around 2 to 5 appear to be optimal for both schemes. Taking more iterates into account increases the computational effort for each iteration, which offsets a further decrease in iteration counts. For this range of depths, L-BFGS and Anderson acceleration have memory footprints of 8 to 14 and 6 to 12 strain fields, respectively, compared with 8.5 for the Newton-CG method, 10.5 for the BFGS-CG method, 2 for the Barzilai-Borwein scheme, and 1 for the basic scheme.

With the optimal depth choice, L-BFGS is the faster of the two limited-memory schemes. However, it performs worse than the (Quasi-)Newton-Krylov methods and the Barzilai-Borwein scheme which exhibit similar runtimes. Even though L-BFGS converges in fewer iterations than the Barzilai-Borwein method, it is slower overall, due to the higher computational cost per iteration. In particular, the parallelization of the inner products in the two-loop recursion of Algorithm 3 was not effective, introducing a significant overhead, cf. Chen et al.[67] The basic scheme is the slowest of the investigated solvers, taking about an order of magnitude longer to converge. Whereas its computational cost per iteration is similar to the Barzilai-Borwein scheme, the required iteration count is significantly higher, due to the pronounced material contrast of the composite during plastification. In conclusion, we observe that the Barzilai-Borwein

TABLE 1 Glass-fiber reinforced polyamide: material parameters of fibers and matrix

Fibers	$E = 72$ GPa,	$\nu = 0.22$				
Matrix	$E = 2.1$ GPa,	$\nu = 0.3$,	$\sigma_Y = 29$ MPa,	$k_1 = 139$ MPa,	$k_2 = 32.7$ MPa,	$m = 319.4$

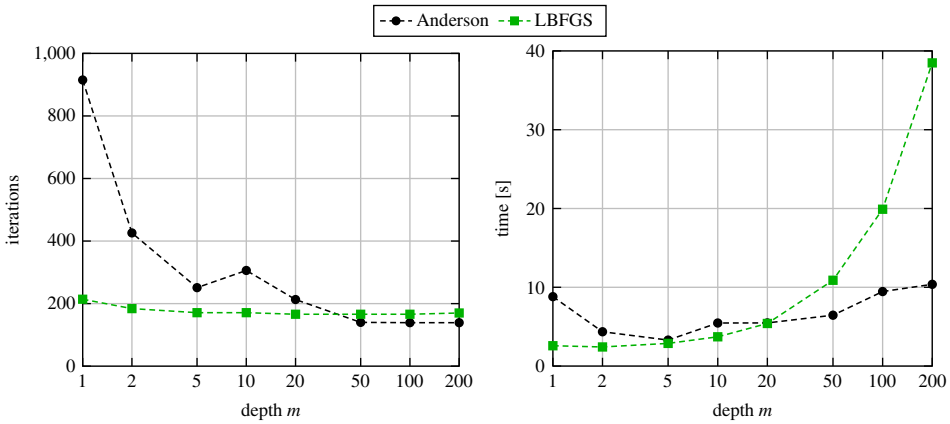


FIGURE 2 Continuous glass-fiber reinforced polyamide: iteration count (left) and computation time (right) with respect to the chosen depth [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 2 Continuous glass-fiber reinforced polyamide: iteration counts and computational runtime with respect to the depth used in the algorithm

	Depth	Iteration count	Computational time (s)
Anderson acceleration	1	915	8.7
	2	426	4.2
	5	251	3.2
	10	306	5.5
	20	213	5.5
	50	140	6.5
	100	139	9.5
	200	139	10.4
L-BFGS	1	214	2.4
	2	184	2.3
	5	171	2.8
	10	171	3.7
	20	166	5.4
	50	166	10.9
	100	166	19.9
	200	170	38.4
Newton CG	–	9 (N)	1.6
		233 (CG)	
BFGS CG	–	14 (N)	2.0
		281 (CG)	
Barzilai-Borwein scheme	–	229	1.7
Basic scheme	–	3897	27.2

Abbreviations: BFGS, Broyden-Fletcher-Goldfarb-Shanno; CG, conjugate gradient; N, Newton.

scheme outclasses the investigated limited-memory methods both in performance and memory footprint. Therefore, we do not include the latter algorithms in the remaining numerical examples. The performance comparison of the remaining algorithms is expanded in Sections 4.3.3 and 4.4.2 for more complex microstructures and material laws, respectively.

4.3 | Porous short glass-fiber reinforced polyamide

We consider a porous polyamide matrix with short glass-fiber reinforcements, cf. Figure 3, which is resolved by 256^3 voxels. The glass fibers are unidirectionally aligned in x-direction with a volume fraction of 15%. The volume fraction of the pores is 1%. The material models and parameters correspond to those in Section 4.2, cf. Table 1. The given example constitutes a challenging nonlinear test problem for the investigated micromechanical solvers. Due to the high stiffness of the glass fibers in comparison to the softer polymer matrix, the material contrast between the two phases is large. During plastification, the contrast increases even further as the minimum eigenvalue of the polyamides tangential stiffness approaches 0, owing to the exponential hardening law. In combination with the unidirectional short fiber structure, this results in strong localization of the strain fields around the fibers, cf. Figure 3. Last but not least, due to the presence of pores, the material contrast of the overall microstructure is infinite.

First, we investigate the different forcing-term choices from Section 2.4 in the FFT-based setting to identify a suitable general-purpose strategy for the Newton-CG and BFGS-CG method. Next, we compare the performance of the solvers with the given forcing-term choice for studying the material behavior under uniaxial extension.

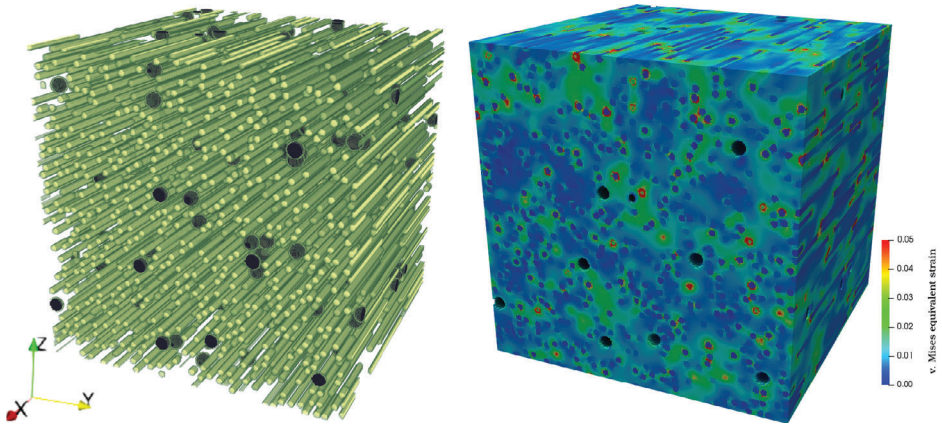
4.3.1 | Influence of the forcing term on convergence and runtime

In their study on forcing-term strategies, Eisenstat and Walker[39] considered numerical examples with up to 10^4 degrees of freedom. In the context of FFT-based micromechanics, much larger problem sizes are commonly considered, as it takes high voxel counts to finely discretize complex microstructures. Thus, we are interested whether the results of Eisenstat and Walker carry over to the FFT-based setting for our current example with $6 \times 256^3 \approx 10^8$ degrees of freedom. Furthermore, we investigate how the BFGS-CG scheme is affected by the different forcing-term strategies in comparison to the Newton-CG scheme. The following choices are considered:

1. Choice 1 corresponds to the first adaptive strategy of Eisenstat and Walker[39] (17)

$$\gamma_n = \left| \frac{\|\Gamma^0 : \sigma(\varepsilon_n)\|_{L^2} - \left\| \left(\mathrm{Id} + \Gamma^0 : \left[\frac{\partial^2 w}{\partial \varepsilon^2}(\varepsilon_n) - \mathbb{C}^0 \right] \right) : \Xi_{n-1} + \Gamma^0 : \sigma(\varepsilon_{n-1}) \right\|_{L^2}}{\|\Gamma^0 : \sigma(\varepsilon_{n-1})\|_{L^2}} \right|,$$

with the associated safequard (19) and Kelley's safeguard against oversolving (22) in place. For this choice, the forcing term is proportional to the disagreement between the gradient and its linear approximation. Thus, γ_n decreases in the vicinity of the solution, and the linear system is solved with increasing accuracy. We start with



(A) Microstructure (256^3 voxels)

(B) Von Mises equivalent strain at 1% uniaxial extension (J_2-elastoplasticity)

FIGURE 3 Porous glass-fiber reinforced polyamide. (A) Microstructure (256^3 voxels) and (B) Von Mises equivalent strain at 1% uniaxial extension (J_2-elastoplasticity)

a high value, that is, low accuracy, of $\gamma_0 = \gamma_{\max} = 0.75$, which also serves as the upper bound for the forcing term.

2. Choice 2 corresponds to the second forcing-term strategy (18) by Eisenstat and Walker[39]

$$\gamma_n = \lambda \left(\frac{\|\Gamma^0 : \sigma(\varepsilon_n)\|_{L^2}}{\|\Gamma^0 : \sigma(\varepsilon_{n-1})\|_{L^2}} \right)^{\alpha},$$

with safeguards (20) and (22) preventing oversolving. Like choice 1, this represents an adaptive strategy. In this case, the ratio of recent residuals serves as a measure of the convergence rate. The latter is expected to decrease close to the solution, leading to smaller values of γ_n. For the algorithmic parameters, we chose $\lambda = 1$ and $\alpha = \frac{1+\sqrt{5}}{2}$, resulting in a convergence behavior similar to choice 1. The initial value and upper bound for the forcing term are set to $\gamma_0 = \gamma_{\max} = 0.75$.

3. Choice 3 is given by $\gamma_n = 0.1$, that is, the forcing term is set to a constant value, corresponding to a modest accuracy for solving the linear system. Kelley[51] suggests this choice as a simple forcing-term strategy which works well in practice.
4. Choice 4 sets the forcing term to a low constant value of $\gamma_n = 5 \times 10^{-5}$, corresponding to a high accuracy. The accuracy is chosen so that the Newton-CG scheme converges in one step for the linear elastic case.

The boundary conditions for the problem correspond to uniaxial extension up to 1% tensile strain in fiber direction, parallel to the x-axis. The load is applied in a single step.

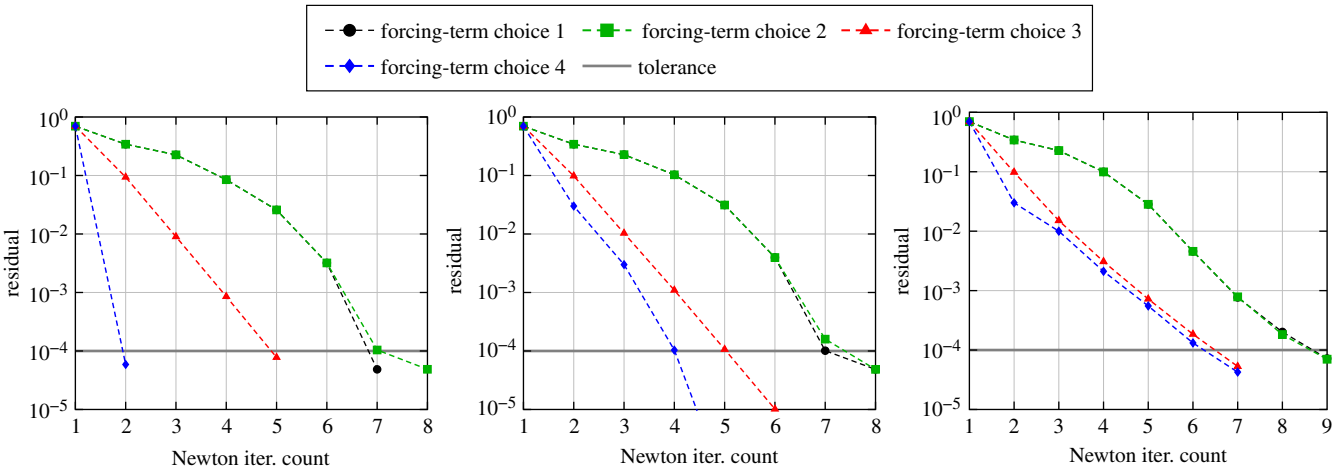
Two scenarios are considered. In the first case, the polyamide matrix is assumed to behave in a purely elastic way, resulting in a linear problem. For this example, the Newton-CG scheme and the BFGS-CG scheme are equivalent. In particular, this allows us to investigate the characteristic convergence behavior of the adaptive forcing-term choices 1 and 2 and the modest accuracy choice 3. Furthermore, we are interested how the computational runtimes of choices 1 to 3 compare with that of choice 4, which is expected to converge in a single Newton step.

In the second case, the matrix behavior is governed by J_2-elastoplasticity, constituting a nonlinear problem. For the Newton-CG scheme, we compare the convergence behavior of the high accuracy choice 4 to the other options and evaluate whether quadratic convergence can be reached. Furthermore, we discuss how the convergence behavior for the different strategies changes when the approximated tangent stiffness of the BFGS-CG scheme is used. We conclude the investigation by evaluating the computational performance of the forcing-term choices for both solvers and evaluate whether a strategy of choice can be identified.

To evaluate the impact of the different forcing-term choices, the residual is plotted as a function of the number of Newton iterations in Figure 4, and as a function of the computation time in Figure 5. The final iteration counts and computation times are listed in Table 3.

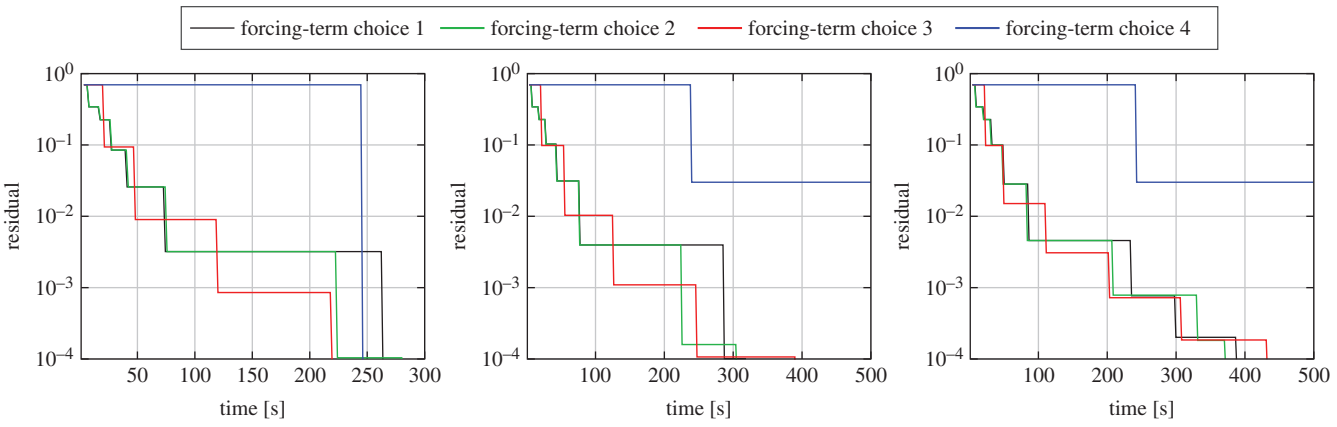
First, we take a look at the linear elastic case. As expected, the Newton scheme converges in a single step for the high accuracy choice 4. Choice 3 requires five iterations and converges at a linear rate. For choices 1 and 2, the convergence behavior is similar. Both start with a low accuracy and a comparatively slow convergence rate. As the residual becomes smaller, the value of γ_n decreases and the linear system is solved to higher accuracy. Consequently, the convergence rate increases for the last iterations. For the linear elastic case, we observe that the overall number of iterations, that is, the sum of CG and Newton iterations, is similar for all forcing-term strategies, cf. Table 3. The computational effort of solving the linear system to high accuracy is comparable to taking a larger number of Newton steps with modest accuracy. Hence, despite the differences in Newton iteration counts, the different forcing-term choices exhibit similar computation times, cf. Figure 5. Notably, choice 4 is not the fastest even, though it led to convergence in a single step. The remaining difference in runtimes between the choices is explained by the wasted computational effort of solving to a smaller residual than required. Fortuitously, the final residual for choice 3 is the closest to the chosen tolerance, leading to the lowest computation time.

Next, we consider the nonlinear case solved by the Newton-CG scheme. For choices 1 to 3, the convergence behavior is similar to the linear elastic case. Choice 4, however, requires five iterations and does not converge much faster than choice 3, even though a much higher accuracy is used. Note that for the current example, the Newton-CG scheme with forcing-term choice 4 does not exhibit a quadratic convergence rate within the chosen tolerance. For a preliminary computation on the small microstructure of Section 4.2, we could confirm a quadratic convergence rate for the Newton-CG method using very low tolerances $\delta = 10^{-8}$ and $\gamma_n = 10^{-9}$ and thereby validate our implementation. However, the computational effort wasted by oversolving was even more excessive for such a setup. With respect to computation time, choices 1 and 2 are the fastest for the current example, converging after just over 300 seconds. Choice 3 takes roughly 30%



(A) Linear elastic matrix behavior: Newton-CG solver (B) J_2-elastoplastic matrix behavior: Newton-CG solver (left) and BFGS-CG solver (right)

FIGURE 4 Porous glass-fiber reinforced polyamide: residual vs number of Newton iterations. (A) Linear elastic matrix behavior: Newton-CG solver and (B) J_2-elastoplastic matrix behavior: Newton-CG solver (left) and BFGS-CG solver (right). BFGS, Broyden-Fletcher-Goldfarb-Shanno; CG, conjugate gradient [Colour figure can be viewed at wileyonlinelibrary.com]



(A) Linear elastic matrix behavior: Newton-CG solver (B) J_2-elastoplastic matrix behavior: Newton-CG solver (left) and BFGS-CG solver (right)

FIGURE 5 Porous glass-fiber reinforced polyamide: residual vs computation time. (A) Linear elastic matrix behavior: Newton-CG solver and (B) J_2-elastoplastic matrix behavior: Newton-CG solver (left) and BFGS-CG solver (right). BFGS, Broyden-Fletcher-Goldfarb-Shanno; CG, conjugate gradient

longer. Taking a look at the overall runtime of choice 4 reveals the computational cost of oversolving. For this example, the advantage of Kelley's safeguard (22) becomes apparent. For all forcing-term strategies, we arrive at a residual slightly above the desired accuracy in the second to last iteration. For the adaptive choices 1 and 2, safeguard (22) is active and, consequently, the linear system is solved to low accuracy in short time. In case of the constant choices 3 and 4, where the safeguard is not used, we arrive at residuals much smaller than the desired accuracy, wasting computational effort.

To conclude the investigation, we take a look at the BFGS-CG scheme. For this solver, choices 3 and 4 lead to roughly the same linear rate of convergence. After few initial steps with a low accuracy, an identical convergence rate is approached for choices 1 and 2, as well. Apparently, higher accuracy than for choice 3 does not improve the convergence rate for the BFGS tangent approximation (30). With respect to the overall runtime, choices 1 and 2 are fastest, with choice 3 being only marginally slower. Choice 4 is the slowest option by far.

To summarize, we observe that for nonlinear material behavior, the forcing-term choices 1 and 2 by Eisenstat and Walker lead to the shortest runtime. However, choice 3 with a constant forcing term of $\gamma_n = 0.1$ is not much slower and serves as an easy-to-implement alternative. Based on the performance of choice 4, we come to the same conclusion as

TABLE 3 Porous glass-fiber reinforced polyamide: iteration counts and computation times for different forcing-term choices

Linear elastic matrix					
		Choice 1	**Choice 2**	**Choice 3**	**Choice 4**
Computation time (s)		264.0	281.8	219.6	246.3
Newton iteration count		7	8	5	2
CG iteration count		126	132	107	119
Matrix governed by J_2 plasticity					
		Choice 1	**Choice 2**	**Choice 3**	**Choice 4**
Newton-CG	Computation time (s)	321.0	306.1	391.2	1524.0
	Newton iteration count	8	8	6	5
	CG iteration count	154	147	193	770
BFGS-CG	Computation time (s)	389.7	373.5	434.3	2109.9
	Newton iteration count	9	9	7	7
	CG iteration count	179	174	207	1053

Abbreviations: BFGS, Broyden-Fletcher-Goldfarb-Shanno; CG, conjugate gradient.

Knoll and Keyes[43]: Aiming for a high (possibly quadratic) convergence rate by solving the linear system to high accuracy is inefficient with respect to the overall runtime of the scheme. These conclusions hold both for Newton-CG and BFGS-CG. Comparing the two solution schemes, we find that for the fastest forcing-term choice 2 the BFGS-CG scheme is only about 22% slower than the Newton-CG method, even though we applied a large nonlinear load step. For the material laws considered in this example, we conclude that the BFGS update leads to a decent approximation of the tangent stiffness in a limited number of iterations.

4.3.2 | Discussion of the effective elastoplastic material properties

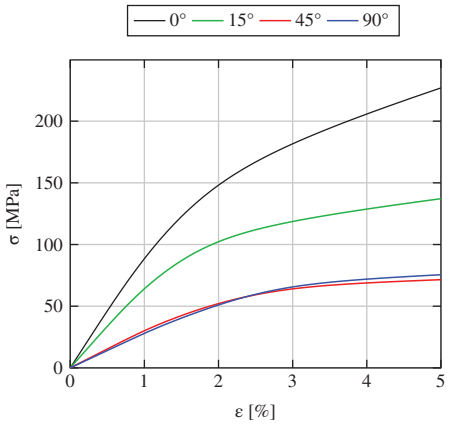
From a material-science viewpoint, the effective elastoplastic behavior of the composite material is of interest. In particular, this includes characterizing the anisotropy of the stress-strain relation in the elastic regime and the shape of the yield-boundary. To this end, we simulate uniaxial tensile tests in various directions relative to the fiber direction, that is, the x-axis. To be specific, the loading is applied at 0°, 15°, 45°, and 90° relative to the x-axis in the xz- and xy-plane and at 0°, 45°, and 90° relative to the y-axis in the yz-plane. The tensile tests are performed up to 5% strain in load direction and subdivided into 50 load steps to obtain finely resolved stress-strain curves. This gives us the opportunity to evaluate the performance of the investigated solvers for a relevant practical application.

This paragraph focuses on the characterization of the material behavior, based on the results of the simulations. The convergence behavior and runtimes of the solution schemes are subsequently discussed in Section 4.3.3. The linear elastic behavior of the composite is characterized by the effective stiffness tensor $\overline{\mathbb{C}}$ relating effective stress $\overline{\sigma} = \langle \sigma \rangle$ and effective strain $\overline{\varepsilon} = \langle \varepsilon \rangle$ by Hooke's law

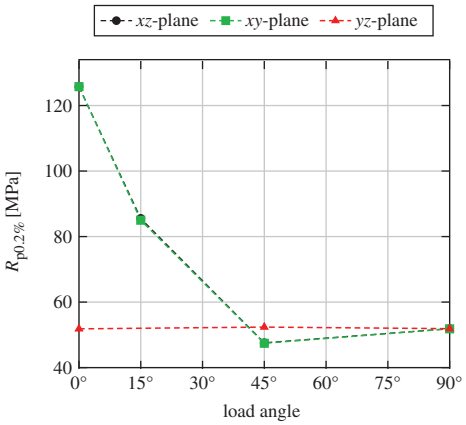
$$\overline{\sigma} = \overline{\mathbb{C}} : \overline{\varepsilon}.$$

Using the elastic parameters in Table 1, the effective stiffness of the composite material, given in Voigt's notation, reads

$$\overline{\mathbb{C}} = \begin{bmatrix} 10.1 & 1.42 & 1.41 & 0.01 & 0.0 & 0.01 \\ 1.42 & 3.49 & 1.45 & 0.03 & 0.0 & 0.0 \\ 1.41 & 1.45 & 3.48 & 0.02 & 0.0 & 0.0 \\ 0.01 & 0.03 & 0.02 & 1.04 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.11 & 0.02 \\ 0.01 & 0.0 & 0.0 & 0.0 & 0.02 & 1.11 \end{bmatrix} \text{GPa},$$



(A) Stress-strain curves for varying load angles in the *xz*-plane



(B) Offset yield strength $R_{p0.2\%}$ at varying load angles in the *xz*-, *xy*- and *yz*-plane

FIGURE 6 Elastoplastic behavior of the porous glass-fiber reinforced polyamide. The load angles are measured relative to the *x*-axis (fiber direction) in the *xz* and *xy*-plane and relative to the *y*-axis in the *yz*-plane. (A) Stress-strain curves for varying load angles in the *xz*-plane and (B) offset yield strength $R_p0.2\%$ at varying load angles in the *xz*-, *xy*-, and *yz*-plane

up to three significant digits, and was identified through six linear elastic computations. $\overline{\mathbb{C}}$ may be well approximated by a transversely isotropic stiffness tensor with engineering constants $E_L = 9.29$ GPa, $E_T = 2.81$ GPa, $\nu_{TT} = 0.38$, $\nu_{LT} = 0.29$, and $G_{LT} = 1.11$ GPa, with a relative error below 1%. As a measure of the elastic anisotropy, we consider $\mathbb{C}^{\text{aniso}}$ defined as

$$\mathbb{C}^{\text{aniso}} = \overline{\mathbb{C}} - \mathbb{C}^{\text{iso}} \quad \text{with} \quad \mathbb{C}^{\text{iso}} = (\overline{\mathbb{C}} :: \mathbb{P}_1)\mathbb{P}_1 + \frac{1}{5}(\overline{\mathbb{C}} :: \mathbb{P}_2)\mathbb{P}_2,$$

where $\mathbb{P}_1$ and $\mathbb{P}_2$ denote the projectors onto the spherical and deviatoric $d \times d$ matrices, respectively. The symbol :: denotes the quadruple tensor contraction, that is, $a = \mathbb{B} :: \mathbb{C}$ is equivalent to $a = B_{ijkl}C_{ijkl}$ in index-notation, using the summation convention. For the given material, $\|\mathbb{C}^{\text{aniso}}\|/\|\overline{\mathbb{C}}\| = 47\%$ in Frobenian norm, signifying strong elastic anisotropy.

The stress-strain curves for the simulated uniaxial tensile tests in the *xz*-plane are shown in Figure 6A. We observe that, up to an angle of 45°, the stiffness decreases and the onset of plastic behavior shifts to lower stresses and higher strains. Between 45° and 90° offset of fiber to load direction, the observed behavior stays roughly identical. A common measure to quantify the onset of plasticity is the offset yield point $R_{p0.2\%}$, as the actual yield stress is difficult to determine for smooth stress-strain diagrams. The offset yield point $R_{p0.2\%}$ is defined as the stress where the component of the effective plastic strain $\bar{\varepsilon}_p = \bar{\varepsilon} - \overline{\mathbb{C}}^{-1} : \bar{\sigma}$ in load direction reaches 0.2%. The results with respect to the load angle are shown in Figure 6B. Due to the isotropic behavior in the *yz*-plane perpendicular to the fiber direction, as well as the similarity of the curves in the *xz*- and *xy*-plane, the boundary of the effective yield surface is approximately transversely isotropic. The yield strength in fiber direction is highest and decreases in a roughly linear way up to a relative angle of 45°. Between 45° and 90°, it stays approximately constant. Even though the yield strength perpendicular to the fiber direction is a factor 2.5 lower than in fiber direction, it is still 1.6 times higher than for the unreinforced matrix material, cf. Table 1.

4.3.3 | Performance comparison for uniaxial extension

Due to the transversely isotropic material behavior, we restrict the performance comparison of the solution schemes to the computations in the *xz*-plane. Figure 7 shows the computation time, the total number of iterations, and the number of gradient evaluations for each load step. For the Newton-CG and BFGS-CG solvers, the total number of iterates denotes the sum of CG and outer iterations, whereas only the latter are counted for the number of gradient evaluations. For the basic scheme and the Barzilai-Borwein scheme, the gradient is evaluated in each iteration, leading to identical counts for both values.

Qualitatively, the resulting plots for the computations at varying load angles are roughly similar. As the affine-linear extrapolation takes effect, the iteration counts and runtimes significantly decrease from the first to the second iteration. For the computations with relative load angles of 45° and 90°, the second load step is still linear elastic and the solution schemes converge within a single iteration. Subsequently, the iteration counts increase at the onset of plastification and

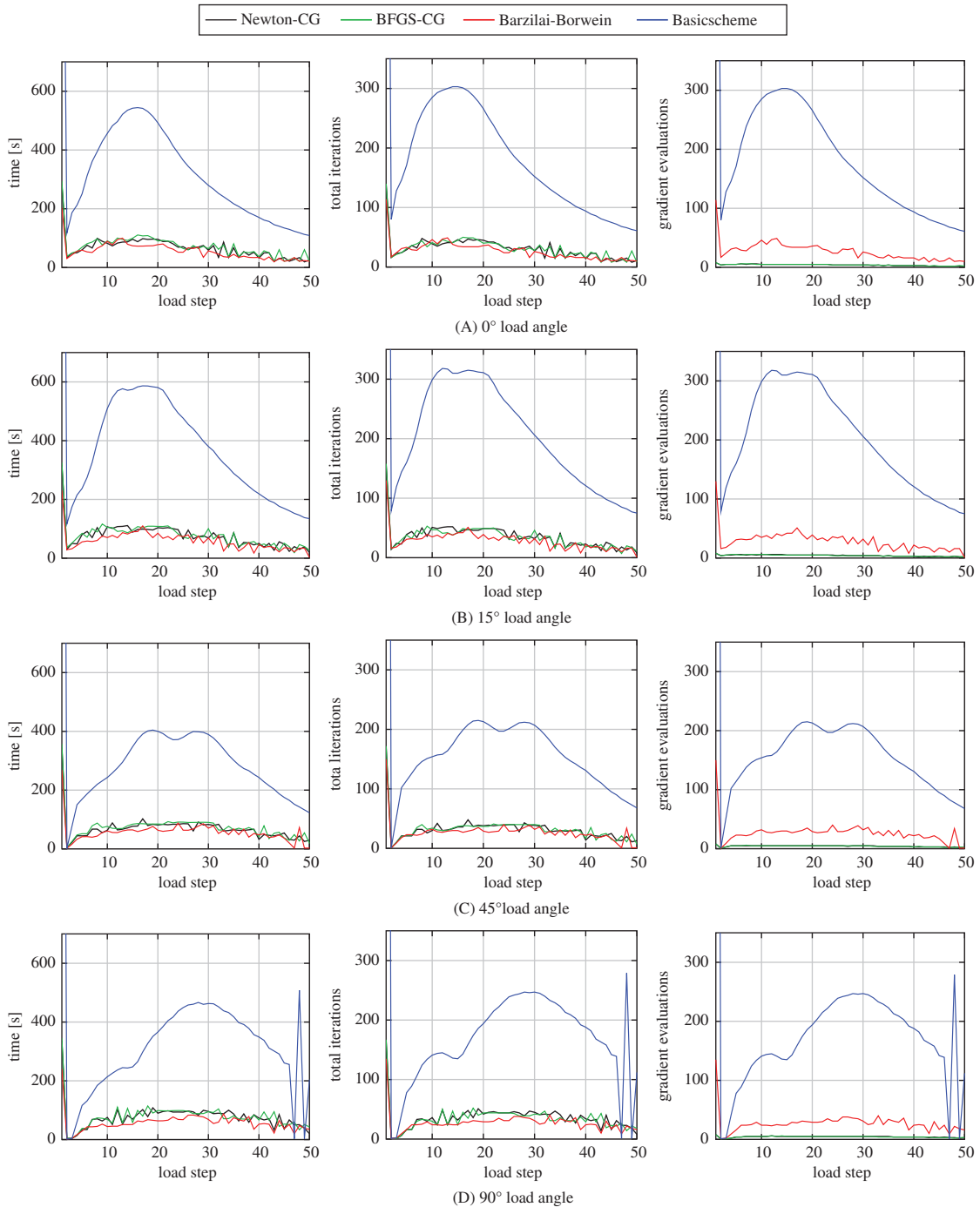


FIGURE 7 Porous glass-fiber reinforced polyamide: performance comparison of the solution schemes for uniaxial extension at various load angles relative to the *x*-direction in the *xz*-plane. (A) 0° load angle, (B) 15° load angle, (C) 45° load angle, and (D) 90° load angle

decrease again after the material is fully plastified. Taking a closer look at the BFGS-CG method, we notice that its performance closely matches that of the Newton-CG method. This observation holds for both the overall performance, cf. Table 4, as well as for the iteration count and runtime within each load step, cf. Figure 7. The tangent stiffness tensor for J_2-elastoplasticity is merely a rank-one update of the elastic stiffness tensor, cf. Section 3.3.2 in Simo and Hughes.[64] As the BFGS-CG method is initialized with the elastic stiffness, the analytic tangent is well approximated within a few BFGS-updates.

		0°	15°	45°	90°
Newton-CG	Mean Newton iteration count	4.1	4.3	4.3	4.4
	Mean CG iteration count	28.9	31.7	27.6	31.3
	Mean computation time (s)	72.3	76.0	69.3	76.1
BFGS-CG	Mean Newton iteration count	4.1	4.2	4.3	4.3
	Mean CG iteration count	28.0	30.1	28.0	30.5
	Mean computation time (s)	72.3	78.1	73.2	78.3
Barzilai-Borwein	Mean iteration count	27.2	28.7	27.0	27.4
	Mean computation time (s)	55.6	59.3	56.4	57.4
Basic scheme	Mean iteration count	199.8	236.6	201.4	210.9
	Mean computation time (s)	341.1	411.6	347.7	367.8

TABLE 4 Porous glass-fiber reinforced polyamide: mean computation times and iteration counts for uniaxial extension at various load angles in the *xz*-plane

Abbreviations: BFGS, Broyden-Fletcher-Goldfarb-Shanno; CG, conjugate gradient.

	Mean computation time per application (ms)
Material law	653.0
Tangent	315.9
FFT	893.7
Γ^0 operator	147.9

TABLE 5 Porous glass-fiber reinforced polyamide: computation time per application of the most expensive operations for loading in *x*-direction and solved by the Newton-CG method

Abbreviation: CG, conjugate gradient.

Evaluating the material law of J_2-elastoplasticity is comparatively cheap, cf. Simo and Hughes.[64] More precisely, the computation time spent on evaluating $\varepsilon \mapsto \sigma(\varepsilon)$ for all voxels is roughly of the same order of magnitude as the computation time for the application of Γ^0 and the associated FFTs for typical cell sizes and resolutions. Usually, these are the most expensive steps in an FFT-based solution algorithm. In Table 5, the average computation time per application of these operations is given for the 0° load case solved by the Newton-CG method. For the given problem, we observe that evaluating the material law is slightly faster than applying forward and backward FFT, and about twice as expensive as applying the tangent $\Xi \mapsto \frac{\partial^2 w}{\partial \varepsilon^2}(\varepsilon_n) : \Xi$, that is, a linear elastic material. The results for the other load cases and solution schemes are roughly similar. Note that the tangent operator is only applied when using the Newton-CG and BFGS-CG method. As a consequence, the computational cost of a gradient evaluation is similar to a CG iteration and the runtimes of all solvers are roughly proportional to their total iteration count, cf. Figure 7. Thus, even though the Newton-CG and BFGS-CG method require much less evaluations of the material law, the Barzilai-Borwein scheme converges faster. The basic scheme is slower than the other investigated algorithms by a factor of 5 to 8. Due to the affine-linear extrapolation, the difference in performance is not as pronounced as for our previous example in Section 4.2.

4.4 | Directionally solidified NiAl-Cr(Mo) alloy

Due to its high melting point and corrosion-resistance, nickel-aluminum-chrome eutectics with minor additions of molybdenum (NiAl-Cr(Mo) alloys) are a promising class of structural high-temperature materials. The material behavior of the components in this alloy is governed by single-crystal elastoviscoplasticity. Compared with the material laws of Section 4.3, linear elasticity and J_2-elastoplasticity, evaluating the material law of a single-crystal elastoviscoplasticity model is considerably more expensive and tends to dominate the overall computation time.[68] Thus, NiAl-Cr(Mo) alloys represent a valuable benchmark for the investigated solution schemes. It is expected that the number of required gradient evaluations is more indicative of the overall performance in this case. This fact favors the use of (Quasi-)Newton-Krylov methods, as the solution of the linear system is less relevant for the runtime.

After a directional solidification process, NiAl-Cr(Mo) develops a cellular structure with NiAl and Cr(Mo) lamellae parallel to the growth direction.[69] Similar microstructures are observed for other intermetallics, for example,

titanium-aluminides[70] or iron-aluminides.[71,72] To investigate mechanical behavior of a lamellar NiAl-Cr(Mo) alloy, a cellular microstructure with 512 grains was generated using the Voronoi tessellation routine of the software Neper.[73] Based on findings by Whittenberger et al[74] and Raj and Locci[75] for moderate solidification rates, an aspect ratio of 4 along the growth direction parallel to the y-axis was chosen for the grains. The microstructure is shown in Figure 8, resolved by 64^3 voxels.

Notice that we do not resolve the lamellar structure for each grain as this would require an excessively high voxel count. Instead, we homogenize a two-phase laminate for each voxel using the algorithm presented in Kabel et al.[76] The orientation of the grains was chosen so that the normal direction of the laminate interface is uniformly distributed in the xz-plane, that is, perpendicular to the growth direction. Cline and Walter[69] investigated the crystallographic relationship in the laminate and showed that all planes and directions of NiAl and Cr(Mo) are parallel. The laminate interface is parallel to the $(11\bar{2})$ plane and the growth direction is parallel to the $\langle 111 \rangle$ direction.

For the two phases of the laminate, the material behavior is governed by a single-crystal elastoviscoplastic model. The infinitesimal strain is additively decomposed

$$\varepsilon = \varepsilon_\mathrm{e} + \varepsilon_\mathrm{p},$$

into elastic ε_e and plastic ε_p parts. The stress-strain relationship follows Hooke's law

$$\sigma = \mathbb{C} : \varepsilon_\mathrm{e} = \mathbb{C} : (\varepsilon - \varepsilon_\mathrm{p}),$$

for the elastic strains. For single-crystal elastoviscoplasticity, the plastic strain is composed of simple shear deformations of the individual crystallographic slip systems. The evolution of the plastic strain is governed by

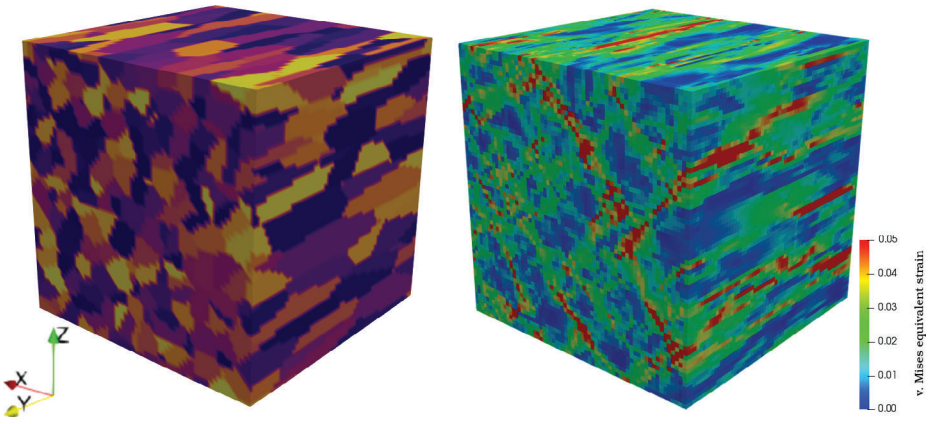
$$\dot{\varepsilon}_\mathrm{p} = \sum_{\alpha=1}^{N} \dot{\gamma}_\alpha d_\alpha \otimes^s n_\alpha,$$

where $\dot{\gamma}_\alpha$, d_α, and n_α denote the slip rate, slip direction, and slip plane normal for the αth of N slip systems, respectively, cf. Bishop.[77] The operator $\otimes^\mathrm{s}$ denotes the symmetrized dyadic product. For the flow rule of the slip rate, we chose the power-law formulation of Hutchinson[78]

$$\dot{\gamma}_\alpha = \dot{\gamma}_0 \, \mathrm{sgn}(\tau_\alpha) \left| \frac{\tau_\alpha}{\tau^\mathrm{F}} \right|^m, \quad \text{with} \quad \tau_\alpha = \sigma : (d_\alpha \otimes^s n_\alpha),$$

and reference slip-rate $\dot{\gamma}_0$, yield stress τ^F, and stress exponent m. For the reinforcing Cr(Mo) phase, the yield stress τ^F is modeled following Albiez.[79]

$$\tau^F = \frac{\tau_\infty}{d\sqrt{\rho}+1} \quad \text{with} \quad \rho = \rho_\mathrm{s}\left[1 - \exp\left(-\frac{1}{2}k_2\gamma\right)\left(1 - \sqrt{\frac{\rho_0}{\rho_\mathrm{s}}}\right)\right]^2, \tag{31}$$



(A) Grain structure (64^3 voxels)

(B) Von Mises equivalent strain after 100 seconds of creep loading at 200 MPa in z-direction

FIGURE 8 Directionally solidified NiAl-Cr(Mo). (A) Grain structure (64^3 voxels) and (B) Von Mises equivalent strain after 100 seconds of creep loading at 200 MPa in z-direction

TABLE 6 Directionally solidified NiAl-Cr(Mo): material parameters of Cr(Mo) lamellae and NiAl matrix[80]

	Cr(Mo)		NiAl	
Volume fraction	$c_{\mathrm{NiAl}} = 0.54$		$c_{\mathrm{Cr(Mo)}} = 0.46$	
Elastic moduli	$C_{11} = 350.0$ GPa	$C_{12} = 67.8$ GPa	$C_{11} = 182$ GPa	$C_{12} = 120$ GPa
	$C_{44} = 100.8$ GPa		$C_{44} = 85.4$ GPa	
Flow rule	$\dot{\gamma}_0 = 0.4\ \mathrm{s}^{-1}$	$m = 4.6$	$\dot{\gamma}_0 = 10^{-3}\ \mathrm{s}^{-1}$	$m = 5.75$
Hardening	$\tau_\infty = 3256.7$ MPa	$d = 0.409\ \mu$m	$\tau_0^F = 37.25$ MPa	
	$\rho_0 = 10^3\ \mathrm{mm}^{-2}$	$\rho_\mathrm{s} = 2.9 \times 10^7\ \mathrm{mm}^{-2}$		
	$k_2 = 13$			
Slip systems	$\{110\}\langle 111\rangle$		$\{001\}\langle 100\rangle$	
	$\{112\}\langle 111\rangle$		$\{011\}\langle 100\rangle$	
	$\{123\}\langle 111\rangle$		$\{011\}\langle 110\rangle$	

and maximum yield stress τ_∞, characteristic length d, recovery constant k_2, dislocation density ρ with its initial value ρ_0 and its saturation value ρ_s. NiAl is assumed to behave perfectly plastic, that is, $\tau^F = \tau_0^F$. The material parameters and volume fractions for NiAl-31Cr-3Mo are taken from Albiez et al,[80] cf. Table 6.

Note that the single-crystal plasticity model with Hutchinson's flow rule is not a generalized standard material[81] and has a nonsymmetric tangent stiffness. As the tangent stiffness of the phases enters the homogenized tangent stiffness of the laminate, cf. Glüge and Kalisch,[82] this would usually prohibit using the CG method for solving Equation (25). However, we found in another study[83] that using the Newton-CG method and only the considering the symmetric part of the tangent stiffness yielded decent results. Hence, we use the symmetrized tangent stiffness of the single phases for the solution of the laminate and the computation of its tangent.

4.4.1 | Discussion of the effective creep behavior

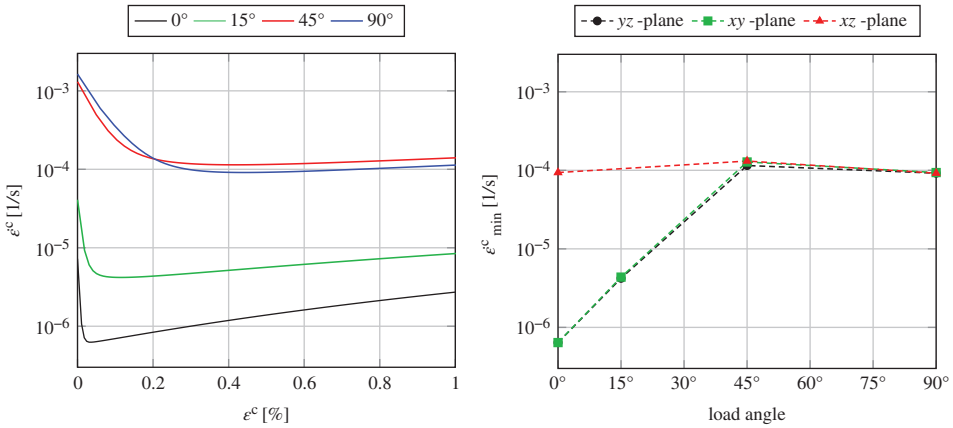
For high-temperature structural materials, the creep behavior, that is, the deformation of the material subjected to a constant stress load, is an important mechanical characteristic. To investigate the anisotropic creep behavior of the NiAl-Cr(Mo) microstructure, we simulate creep tests in various directions relative to the growth direction of the material, that is, the y-axis. More specifically, we apply boundary conditions corresponding to uniaxial compression with a magnitude of 200 MPa at 0°, 15°, 45°, and 90° relative to the y-axis in the yz- and xy-plane and at 0°, 45°, and 90° relative to the x-axis in the xz-plane. The load is applied in 1 second and a single load step and, afterward held constant for 50 load steps for a specified creep time. The creep times for each angle are listed in Table 7 and were chosen to obtain a fine resolution of the creep rate in time. Note that, due to the prescribed softening behavior Equation (31), an excessively coarse resolution of the load steps over time leads to divergence of the solution schemes for this material. Simulating such a creep loading is a challenging problem for the investigated solution schemes, as a load transfer from the softer NiAl to the more creep resistant Cr(Mo) occurs as a viscous effect after the initial loading, cf. Albiez et al.[79,80] Thus, the loading in the single phases is nonmonotone, especially in the first few load steps after the initial loading.

In the following, we discuss the creep behavior observed in the simulations. The performance of the solution schemes for this example is compared in Section 4.4.2. For the characterization of the creep behavior, the creep rate $\dot{\varepsilon}^\mathrm{c}$, that is, the strain component in load direction measured after the initial loading, and its minimum value $\dot{\varepsilon}^\mathrm{c}_\mathrm{min}$ are of interest.

In Figure 9A, the creep curves for the simulations in the yz-plane are shown. The curve for the load in growth direction agrees well with the computational and experimental results reported by Albiez et al.[80] Up to a load angle of 45°,

TABLE 7 Directionally solidified NiAl-Cr(Mo): creep times with respect to load angle for all simulated creep experiments

	0°	15°	45°	90°
yz-plane	10 000 s	2000 s	100 s	100 s
xy-plane	10 000 s	2000 s	100 s	100 s
xz-plane	100 s	–	100 s	100 s



(A) Creep rate vs. creep strain for varying load angles in the *yz*-plane

(B) Minimum creep rates for varying load angles in the *yz*-, *xy*- and *xz*-plane

FIGURE 9 Effective creep behavior of directionally solidified NiAl-Cr(Mo) at different load angles for an applied load of 200 MPa. The load angles are given with respect to the *y*-axis (growth direction) in the *yz*- and *xy*-plane and with respect to the *x*-axis in the *xz*-plane. (A) Creep rate vs creep strain for varying load angles in the *yz*-plane and (B) minimum creep rates for varying load angles in the *yz*-, *xy*-, and *xz*-plane

we observe an increase in the overall creep rate and a less pronounced softening behavior, that is, an increase of the creep rate at increasing strains. This signifies that, in case of aligned load and growth direction, a large amount of stress is carried by the creep resistant Cr(Mo) lamellae which in turn activates their softening behavior. Figure 9B shows the minimum creep rate for all computations as a function of the load angle. The good agreement of the results in the *yz*- and *xy*-plane as well as the approximately isotropic behavior in the *xz*-plane indicate a transversely isotropic effective creep behavior for NiAl-Cr(Mo). We observe that with increasing angle relative to the growth direction the logarithm of the minimum creep rate increases linearly up to an angle of 45° and subsequently stagnates. The difference between the highest and lowest value for $\dot{\varepsilon}^c_{min}$ is slightly over two orders of magnitude. This represents an improvement in robustness compared with the similar directionally solidified molybdenum-reinforced nickel-aluminum alloys (NiAl-Mo) which form unidirectionally aligned fiber structures instead of laminates. For NiAl-Mo, FFT-based computations predicted a decrease in creep strength by roughly four orders of magnitude down to the level of pure NiAl in case of off-axis loading.[83] Similarly, Seemüller et al[84] experimentally observed a considerable increase in creep rate for NiAl-Mo with a high content of misaligned fibers. Thus, we conclude that the cellular laminate structure of NiAl-Cr(Mo) leads to a weaker anisotropy and a larger robustness against misaligned loading compared to fibrous materials with a similar composition.

4.4.2 | Performance comparison for creep loading

In analogy to Section 4.3.3, we take a closer look at the runtimes, total iteration counts and gradient evaluations of the solvers for each load step, cf. Figure 10. Due to the material's transversely isotropic behavior, we restrict the discussion to the computations in the *yz*-plane. During the first few load steps of the creep computations, we observe high iteration counts and runtimes, due to the initial load application and the subsequent load transfer. This behavior is less pronounced for the case where growth direction and loading direction are parallel. As the normal direction of the laminates are distributed in the *xz*-plane, all laminate planes are parallel to the *y*-direction. Thus, the resultant fields are less heterogenous for a loading in this direction, leading to lower computational costs. As the fields stabilize and the affine-linear extrapolation takes effect, computation times and the required number of material evaluations decrease to a lower level, roughly between load step 5 and 15. For the 0° load angle and 15° load angle computations, the computation time per material evaluation increases with the creep time, due to the softening of the material. In the former case, the required number of iterations increases as well, as the softening is more pronounced and leads to a higher internal material contrast.

In contrast to our previous example in Section 4.3.3, solving the two-phase laminate and evaluating the single-crystal elastoviscoplastic material laws dominates the overall computation time, cf. Table 8. This holds true for all solvers and load cases. Hence, we observe that the runtime is approximately proportional to the number of gradient evaluations.

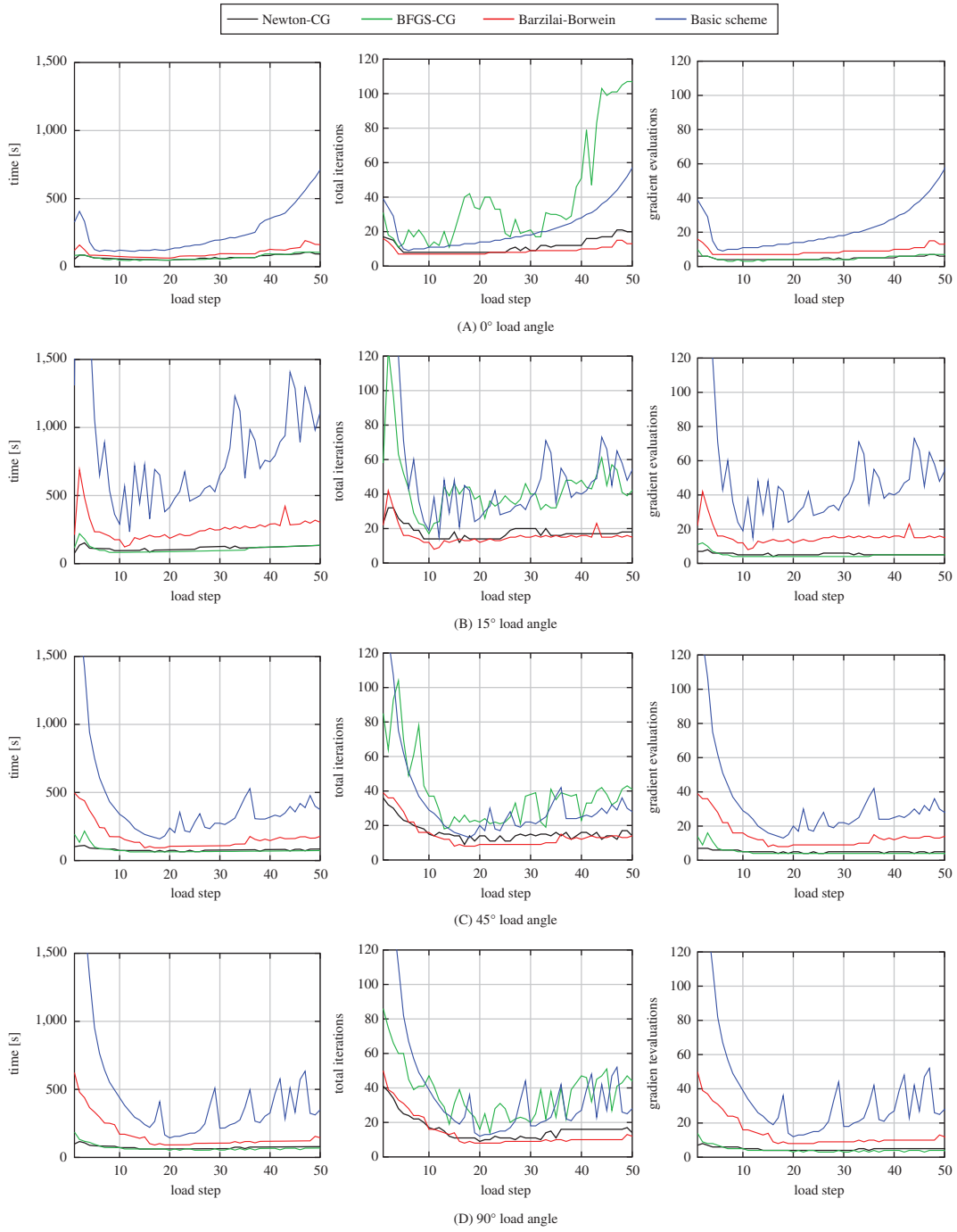


FIGURE 10 Directionally solidified NiAl-Cr(Mo): performance comparison of the solution schemes for creep loading at various load angles relative to the y-direction in the yz-plane. (A) 0° load angle, (B) 15° load angle, (C) 45° load angle, and (D) 90° load angle

We take a closer look at the convergence behavior of the BFGS-CG method. Roughly up to the fifth load step, the BFGS-CG method requires a higher number of Newton iterations than the Newton-CG method. In comparison to the example in Section 4.3.3, it takes more BFGS update iterations to achieve a good approximation of the tangent stiffness. First, this can be traced back to the difference in loading. Whereas the first load steps of the uniaxial extension in Section 4.3.3 were in the linear elastic regime, the creep loading is rapidly applied in the first load step, immediately leading to nonlinear material behavior. Second, the tangent stiffness for the single-crystalline phases and the resulting homogenized tangent stiffness of the laminate is more complex than the one of J_2-elastoplasticity. Thus, with the linear elastic stiffness

TABLE 8 Directionally solidified NiAl-Cr(Mo): computation time per application of the most expensive operations for the case of loading in *y*-direction solved by the Newton-CG method

	Mean computation time per application (ms)
Material law	9944.0
Tangent	5.0
FFT	6.7
Γ^0 operator	2.4

Abbreviations: CG, conjugate gradient; FFT, fast Fourier transform.

as starting point, more BFGS updates are necessary to approximate the material's tangent stiffness. After the slower initial load steps, BFGS-CG and Newton-CG exhibit similar runtimes and Newton iteration counts. In fact, the BFGS-CG method even converges in slightly fewer Newton iterations than the Newton-CG method for some load steps. This may be due to a combination of two factors. First, we use the symmetrized tangent of the single phases to compute the tangent of the laminate. Second, we do not achieve the highest possible convergence rate for Newton-CG, using the forcing-term choice 2, cf. Section 4.3.1. We further note that the BFGS-CG method requires more CG iterations than Newton-CG, cf. Table 9. This indicates that the BFGS tangent approximation exhibits a higher internal material contrast than the analytic tangent for this example. Comparing the mean computation times per load step, we see that this does not negatively impact the method's overall performance. In conclusion, BFGS-CG and Newton-CG exhibit very similar computation times with BFGS-CG being even slightly faster for the 15° to 90° load angle computations.

For the Barzilai-Borwein method, we note that the total number of iterations is similar to the Newton-CG method for all computations. However, as the material law is evaluated for every iteration of the Barzilai-Borwein scheme, the resulting computation times are 1.5 to 2.5 times higher than for Newton-CG and BFGS-CG.

The basic scheme is the most time-consuming algorithm, taking about 4 to 10 times longer to converge than the inexact (Quasi-)Newton methods. Note that for all load cases except the 0° loading, the iteration counts of the basic scheme fluctuate significantly between load steps, even after the strain field stabilizes and the creep rate reaches its minimum value. This unexpected effect is a result of our choice of reference material $\beta^0 = (\beta^+ + \beta^-)/2$, which is only theoretically justified for materials whose tangent has a lower bound. For our given material, this cannot be assured globally, due to the prescribed softening behavior. However, convergence of the basic scheme to a critical point can be shown for materials with only an upper bound on the tangent if the reference material is chosen as $\beta^0 = \beta^+$, cf. Section 1.2.3 in Nesterov's book.[85] We compared the two choices for β^0 for the 15° load case where the fluctuations were most pronounced, cf. Figure 11. For the conservative choice $\beta^0 = \beta^+$, iteration counts and runtimes develop smoothly. However, the mean iteration count and computation time per load step are about 30% higher for this choice. Hence, the results for $\beta^0 = (\beta^+ + \beta^-)/2$ were included in the performance comparison of the different solution schemes.

TABLE 9 Directionally solidified NiAl-Cr(Mo): mean computation times and iteration counts for creep loading at various angles in the *yz*-plane

		0°	15°	45°	90°
Newton-CG	Mean Newton iteration count	4.8	5.4	5.1	4.8
	Mean CG iteration count	6.8	12.3	10.9	11.2
	Mean computation time (s)	68.4	114.2	78.5	73.8
BFGS-CG	Mean Newton iteration count	4.9	4.9	4.9	4.3
	Mean CG iteration count	35.1	38.2	33.2	32.8
	Mean computation time (s)	68.9	107.2	77.6	69.4
Barzilai-Borwein	Mean iteration count	9.1	15.7	14.9	14.0
	Mean computation time (s)	100.6	258.7	166.4	161.2
Basic scheme	Mean iteration count	22.7	55.5	46.0	57.8
	Mean computation time (s)	259.6	939.8	611.7	793.2

Abbreviations: BFGS, Broyden-Fletcher-Goldfarb-Shanno; CG, conjugate gradient.

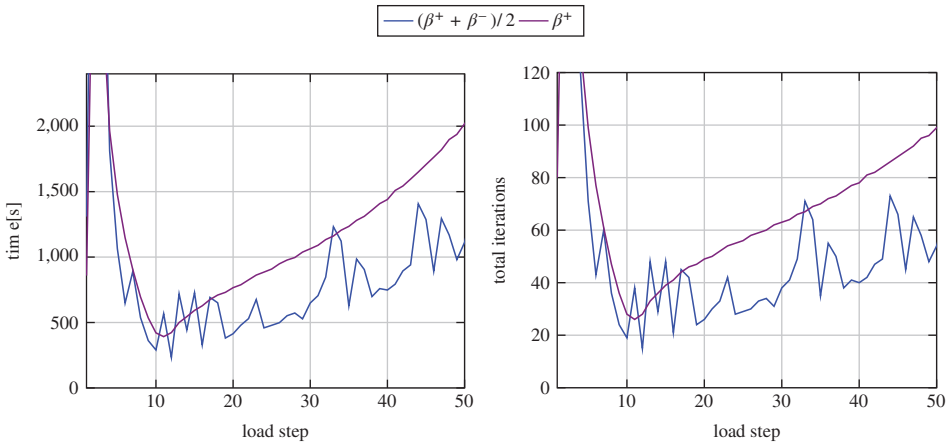


FIGURE 11 Directionally solidified NiAl-Cr(Mo): performance comparison of the two reference-material choices for the basic scheme for the 15° load case

5 | SUMMARY AND CONCLUSION

Quasi-Newton methods, such as Anderson acceleration[3,10,28] and the Barzilai-Borwein method,[19] have attracted considerable attention for FFT-based micromechanics. In contrast to the classical Newton method, these schemes do not require computing the Hessian. In addition, they generally outperform gradient-descent methods which share this property.[24] For the current article, this motivated us to exploit the most popular Quasi-Newton algorithm, the BFGS method, in the context of FFT-based micromechanics. First, we proposed an implementation of Nocedal's L-BFGS algorithm.[40] Although, this scheme proved to be faster than the similar Anderson acceleration, pioneered by Shantraj et al,[3] L-BFGS performed worse than the Barzilai-Borwein method which is nonmonotonic but has a smaller memory-footprint. This can be traced back to the comparatively high computational cost per iteration of L-BFGS, due to the many inner product evaluations in the classical two-loop algorithm, cf. Algorithm 3. It may be possible to reduce this computational overhead, using the more involved L-BFGS implementation proposed by Chen et al,[67] where the computation of all inner products can be parallelized more effectively. However, for material laws which can be cheaply evaluated, the Barzilai-Borwein scheme currently represents the general-purpose method of choice

For computationally expensive material laws, such as single-crystal plasticity, it has been shown that Newton-CG is more efficient, due to the lower number of gradient (and thus material law) evaluations.[83] This led us to our second use of the BFGS update for approximating the material tangent stiffness in the Newton-CG scheme. With the resulting BFGS-CG method, we arrived at a scheme which was competitive in performance to the classical Newton-CG method, in particular for multistep loads. Although it can not be measured in performance benchmarks, time spent programming is as much of a resource as time spent on computations. Thus, the main advantage of the BFGS-CG scheme is that it enables the tangent-free implementation of complex and computationally demanding material laws while still being fast enough to permit their efficient computational homogenization. The results of the performance comparison between the investigated solution schemes are summarized in Table 10.

As a side product of our investigation of (Quasi-)Newton methods, we found a globalization strategy suitable for FFT-based micromechanics in the line-search algorithm of Dong.[4] Another aspect of major importance for the overall performance of these schemes was the choice of the forcing term. Among the various strategies tested in our numerical experiments, consistently solving the linear system to a high accuracy was by far the slowest option. Whereas this increased the overall computation time by factors of 5 to 7 compared with the other choices, the resulting convergence rate with respect to the required Newton iterations was barely improved within the given tolerance. The best overall performance was achieved by forcing-term choice 2 of Eisenstat and Walker and its associated safeguards.[39,52] However, similar performance was observed for a constant moderate forcing term of 0.1. Thus, the choice between these two options can be seen as a matter of preference, that is, choosing optimal performance vs ease of implementation.

As demonstrated in our numerical experiments, both Newton-CG and BFGS-CG can handle nonlinear materials with infinite contrast. Consequently, they are among the most widely applicable algorithms currently available in the FFT-based context. However, the robust handling of materials with negative tangent eigenvalues, for example, in case of damage or strain-softening, is an open topic for further research. Dai[86] demonstrated that the BFGS method does not

TABLE 10 Summary of the performance comparison between the investigated solution schemes

Solution scheme	Memory footprint (strain-like fields)	Summary and remarks
Basic scheme[5,6]	1	• Gradient descent method[2] • Lowest memory requirements • Slowest among the studied solvers
Anderson acceleration[3,26]	$2m + 2$	• Limited-memory Quasi-Newton method[29] • Optimal depth m between 2 and 5 • Accelerates the basic scheme but slower than the remaining algorithms
L-BFGS[40]	$2m + 4$	• Limited-memory Quasi-Newton method • Optimal depth m between 2 and 5 • Outperformed by the more memory-efficient Barzilai-Borwein method
Barzilai-Borwein[19,25]	2	• Gradient descent with step size based on Quasi-Newton methods • Nonmonotonic convergence behavior • Fastest choice for inexpensive material laws
Newton-CG[2,42]	8.5	• Inexact Newton method • Highest efficiency in combination with Eisenstat and Walker's forcing-term choice 2[39] • Requires computing the material tangent • Fastest choice for expensive material laws
BFGS-CG	10.5	• Inexact Quasi-Newton method • Uses the BFGS update to approximate the material tangent • Matches performance of Newton-CG for small load steps, slightly slower otherwise

Abbreviations: BFGS, Broyden-Fletcher-Goldfarb-Shanno; CG, conjugate gradient.

converge for general functions in four or higher dimensions. Damped versions of the BFGS update formula are available, cf. Procedure 18.2 in Nocedal and Wright,[24] which stabilize the convergence behavior of the linear solver. Still, this may result in overall divergence if the disagreement between the tangent and its approximation becomes too large. It remains to be investigated, if a suitable approach such as the arc-length method as used for conventional finite-element computations[87] can be adapted for FFT-based micromechanics.

ACKNOWLEDGEMENTS

D.W. and T.B. gratefully acknowledge the partial financial support by the German Research Foundation (DFG) under the project "Lamellar Fe-Al in situ composite materials" (BO 1466/12-2) [Project name has been added]. M.S. acknowledges partial financial support by the German Research Foundation (DFG) within the International Research Training Group "Integrated engineering of continuous-discontinuous long fiber reinforced polymer structures" (GRK 2078). We thank the anonymous reviewers for their constructive feedback and I. Sprenger for fruitful discussions on directionally solidified eutectics.

ORCID

Matti Schneider https://orcid.org/0000-0001-7017-3618
Thomas Böhlke https://orcid.org/0000-0001-6884-0530

REFERENCES

1. 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.
2. Kabel M, Böhlke T, Schneider M. Efficient fixed point and Newton–Krylov solvers for FFT-based homogenization of elasticity at large deformations. *Comput Mech*. 2014;54(6):1497-1514.
3. Shantraj P, Eisenlohr P, Diehl M, Roters F. Numerically robust spectral methods for crystal plasticity simulations of heterogeneous materials. *Int J Plast*. 2015;66:31-45.
4. Dong Y. Step lengths in BFGS method for monotone gradients. *Comput Math Appl*. 2010;60(3):563-571.
5. 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.
6. 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.
7. Lebensohn RA, Kanjarla AK, Eisenlohr P. An elasto-viscoplastic formulation based on fast Fourier transforms for the prediction of micromechanical fields in polycrystalline materials. *Int J Plast*. 2012;32-33:59-69.
8. Eisenlohr P, Diehl M, Lebensohn RA, Roters F. A spectral method solution to crystal elasto-viscoplasticity at finite strains. *Int J Plast*. 2013;46:37-53.
9. Boeff M, Gutknecht F, Engels PS, Ma A, Hartmaier A. Formulation of nonlocal damage models based on spectral methods for application to complex microstructures. *Eng Fract Mech*. 2015;147:373-387.
10. Chen Y, Vasiukov D, Gélébart L, Park CH. A FFT solver for variational phase-field modeling of brittle fracture. *Comput Methods Appl Mech Eng*. 2019;349:167-190.
11. Lucarini S, Segurado J. On the accuracy of spectral solvers for micromechanics based fatigue modeling. *Comput Mech*. 2019;63:365-382.
12. Vidyasagar A, Tan WL, Kochmann DM. Predicting the effective response of bulk polycrystalline ferroelectric ceramics via improved spectral phase field methods. *J Mech Phys Solids*. 2017;106:133-151.
13. Kochmann J, Wulfinghoff S, Ehle L, Mayer J, Svendsen B, Reese S. Efficient and accurate two-scale FE-FFT-based prediction of the effective material behavior of elasto-viscoplastic polycrystals. *Comput Mech*. 2017;61:1-14.
14. Göküzüm FS, Nguyen LTK, Keip M-A. A multiscale FE-FFT framework for electro-active materials at finite strains. *Comput Mech*. 2019;64:63-84.
15. 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.
16. Michel JC, Moulinec H, Suquet P. A computational scheme for linear and non-linear composites with arbitrary phase contrast. *Int J Numer Methods Eng*. 2001;52:139-160.
17. Monchiet V, Bonnet G. A polarization-based FFT iterative scheme for computing the effective properties of elastic composites with arbitrary contrast. *Int J Numer Methods Eng*. 2012;89:1419-1436.
18. Schneider M, Wicht D, Böhlke T. On polarization-based schemes for the FFT-based computational homogenization of inelastic materials. *Comput Mech*. 2019;64(4):1073-1095.
19. Schneider M. On the Barzilai-Borwein basic scheme in FFT-based computational homogenization. *Int J Numer Methods Eng*. 2019;118:482-494.
20. Zeman J, Vondřejc J, Novak J, Marek I. Accelerating a FFT-based solver for numerical homogenization of periodic media by conjugate gradients. *J Comput Phys*. 2010;229(21):8065-8071.
21. Brisard S, Dormieux L. FFT-based methods for the mechanics of composites: a general variational framework. *Comput Mater Sci*. 2010;49(3):663-671.
22. Brisard S, Dormieux L. Combining Galerkin approximation techniques with the principle of Hashin and Shtrikman to derive a new FFT-based numerical method for the homogenization of composites. *Comput Methods Appl Mech Eng*. 2012;217-220:197-212.
23. Ma R, Truster TJ. FFT-based homogenization of hypoelastic plasticity at finite strains. *Comput Methods Appl Mech Eng*. 2019;349:499-521.
24. Nocedal J, Wright SJ. *Numerical Optimization*. New York, NY: Springer; 1999.
25. Barzilai J, Borwein JM. Two-point step size gradient methods. *IMA J Numer Anal*. 1988;8:141-148.
26. Anderson DG. Iterative procedures for nonlinear integral equations. *J ACM*. 1965;12(4):547-560.
27. Roters F, Diehl M, Shanthraj P, et al. DAMASK – The Düsseldorf Advanced Material Simulation Kit for modeling multi-physics crystal plasticity, thermal, and damage phenomena from the single crystal up to the component scale. *Comput Mater Sci*. 2018;158:420-478.
28. 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.
29. Fang H-R, Saad Y. Two classes of multisecant methods for nonlinear acceleration. *Numer Linear Algebra Appl*. 2009;16:197-221.
30. Broyden CG. A class of methods for solving nonlinear simultaneous equations. *Math Comput*. 1965;19:577-593.
31. Evans C, Pollock S, Rebholz LG, Xiao M. A proof that Anderson acceleration improves the convergence rate in linearly converging fixed point methods (but not in those converging quadratically). 2018:1–22. arXiv:1810.08455.
32. Pollock S, Rebholz LG. Anderson acceleration for contractive and noncontractive operators. 2019:1–28. arXiv 1909.04638.
33. Broyden CG. The convergence of a class of double rank minimization algorithms: 2. The new algorithm. *J Math Anal Appl*. 1970;6:222-231.
34. Fletcher R. A new approach to variable metric algorithms. *Comput J*. 1970;13:317-322.
35. Goldfarb D. A family of variable metric methods derived by variational means. *Math Comput*. 1970;24:23-26.

36. Shanno DF. Conditioning of quasi-Newton methods for function minimization. *Math Comput*. 1970;24:647-650.
37. Wolfe P. Convergence conditions for ascent methods. *SIAM Rev*. 1969;11(2):226-235.
38. Lahellec N, Suquet P. On the effective behavior of nonlinear inelastic composites: I. Incremental variatonal principles. *J Mech Phys Solids*. 2007;55:1932-1963.
39. Eisenstat SC, Walker HF. Choosing the forcing terms in an inexact Newton method. *SIAM J Sci Comput*. 1996;17(1):16-32.
40. Nocedal J. Updating quasi-Newton matrices with limited storage. *Math Comput*. 1980;35(151):773-782.
41. Kantorovich LV. On Newton's method for functional equations. *Dokl Akad Nauk SSSR*. 1948;59:1237-1240.
42. Dembo RS, Eisenstat SC, Steihaug T. Inexact Newton methods. *SIAM J Numer Anal*. 1982;19:400-408.
43. Knoll DA, Keyes DE. Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *J Comput Phys*. 2004;193:357-397.
44. Griewank A, Walther A. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. 2nd ed. Philadelphia, PA: SIAM Publications; 2008.
45. Powell MJD. Some global convergence properties of a variable metric algorithm for minimization without exact line search. In: Cottle RW, Lemke CE, eds. *Nonlinear Programming, Vol IX of SIAM-AMS Proceedings*. Philadelphia, PA: SIAM Publications; 1976;53-72.
46. Turner PR, Huntley E. Variable metric methods in Hilbert space with applications to control problems. *J Optim Theory Appl*. 1976;19(3):381-400.
47. Griewank A. The local convergence of Broyden-like methods on Lipschitzian problems in Hilbert spaces. *SIAM J Numer Anal*. 1987;24(3):684-705.
48. Dennis JE, Moré JJ. Quasi-Newton methods, motivation and theory. *SIAM Rev*. 1977;19:46-89.
49. Shanno DF, Puah KH. Matrix conditioning and nonlinear optimization. *Math Program*. 1978;14:149-160.
50. Liu DC, Nocedal J. On the limited memory BFGS method for large scale optimization. *Math Program*. 1989;45:503-528.
51. Kelley CT. Numerical methods for nonlinear equations. *Acta Numerica*. 2018;27:207-287.
52. Kelley CT. *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics. Philadelphia, PA: SIAM Publications; 1995.
53. Bakhvalov NS, Panasenko G. *Homogenisation: Averaging Processes in Periodic Media: Mathematical Problems in the Mechanics of Composite Materials*. Mathematics and its Applications. Dordrecht, the Netherlands: Springer; 1989.
54. Willot F. Fourier-based schemes for computing the mechanical response of composites with accurate local fields. *Comptes Rendus Mécanique*. 2015;343(3):232-245.
55. Schneider M, Ospald F, Kabel M. Computational homogenization of elasticity on a staggered grid. *Int J Numer Methods Eng*. 2016;105(9):693-720.
56. Schneider M, Merkert D, Kabel M. FFT-based homogenization for microstructures discretized by linear hexahedral elements. *Int J Numer Methods Eng*. 2017;109(10):1461-1489.
57. Leuschner M, Fritzen F. Fourier-Accelerated Nodal Solvers (FANS) for homogenization problems. *Comput Mech*. 2018;32(3):359-392.
58. Grimm-Strele H, Kabel M. Runtime optimization of a memory efficient CG solver for FFT-based homogenization: implementation details and scaling results for linear elasticity. *Comput Mech*. 2019;64(5):1339-1345.
59. Walker HW, Ni P. Anderson acceleration for fixed-point iterations. *SIAM J Numer Anal*. 2011;49(4):1715-1735.
60. Eyert V. A comparative study on methods for convergence acceleration of iterative vector sequences. *J Comput Phys*. 1996;124(2):271-285.
61. Matthies H, Strang G. The solution of nonlinear finite element equations. *Int J Numer Methods Eng*. 1979;14:1613-1626.
62. Frigo M, Johnson SG. The design and implementation of FFTW3. *Proc IEEE*. 2005;93:216-231.
63. Doghri I, Brassart L, Adam L, Gérard J-S. A second-moment incremental formulation for the mean-field homogenization of elasto-plastic composites. *Int J Plast*. 2011;27:352-371.
64. Simo JC, Hughes TJR. *Computational Inelasticity*. New York, NY: Springer; 1998.
65. Krairi A, Doghri I, Schalnat J, Robert G, Van Paepegem W. Thermo-mechanical coupling of a viscoelastic-viscoplastic model for thermoplastic polymers: thermodynamical derivation and experimental assessment. *Int J Plast*. 2019;115:154-177.
66. Kabel M, Fliegener S, Schneider M. Mixed boundary conditions for FFT-based homogenization at finite strains. *Comput Mech*. 2016;57(2):193-210.
67. Chen W, Wang Z, Zhou J. Large-scale L-BFGS using MapReduce. *Adv Neural Inf Proces Syst*. 2014;27:1332-1340.
68. Eghtesad A, Barrett TJ, Germaschewski K, Lebensohn RA, McCabe RJ, Knezevic M. OpenMP and MPI implementations of an elasto-viscoplastic fast Fourier transform-based micromechanical solver for fast crystal plasticity modeling. *Adv Eng Softw*. 2018;126:46-60.
69. Cline HE, Walter JL. The effect of alloy additions on the rod-plate transition in the eutectic NiAl-Cr. *Metall Trans A*. 1970;1(10):2907-2917.
70. Huang S-C, Hall EL. Plastic deformation and fracture of binary TiAl-base alloys. *Metall Trans A*. 1991;22(2):427-439.
71. Scherf A, Kauffmann A, Kauffmann-Weiss S, et al. Orientation relationship of eutectoid FeAl and $FeAl_2$. *J Appl Crystallogr*. 2016;49:442-449.
72. Schmitt A, Kumar KS, Kauffmann A, Li X, Stein F, Heilmaier M. Creep of binary Fe-Al alloys with ultrafine lamellar microstructures. *Intermetallics*. 2017;90:180-187.
73. Quey R, Dawson PR, Barbe F. Large-scale 3D random polycrystals for the finite element method: generation, meshing and remeshing. *Comput Methods Appl Mech Eng*. 2011;200(17-20):1729-1745.
74. Whittenberger JD, Raj SV, Locci IE. Effect of microstructure solidified on creep in directionally NiA1-31Cr-3Mo. Technical Report NASA/TM-2001-211306, NASA Glenn Research Center, 2001.
75. Raj SV, Locci IE. Microstructural characterization of a directionally-solidified Ni-33 (at.%) Al-31Cr-3Mo eutectic alloy as a function of withdrawal rate. *Intermetallics*. 2001;9(3):217-227.

76. Kabel M, Fink A, Schneider M. The composite voxel technique for inelastic problems. *Comput Methods Appl Mech Eng*. 2017;322:396-418.
77. Bishop JFW. A theoretical investigation of the plastic deformation of crystals by glide. *Philos Mag*. 1953;44:51-64.
78. Hutchinson JW. Bounds and self-consistent estimates for creep of polycrystalline materials. *Proc R Soc Lond A*. 1976;348:101-127.
79. Albiez J, Sprenger I, Seemüller C, Weygand D, Heilmaier M, Böhlke T. Physically motivated model for creep of directionally solidified eutectics evaluated for the intermetallic NiAl–9Mo. *Acta Mater*. 2016;110:377-385.
80. Albiez J, Sprenger I, Weygand D, Heilmaier M, Böhlke T. Validation of the applicability of a creep model for directionally solidified eutectics with a lamellar microstructure. *PAMM*. 2016;16:297-298.
81. Steinmann P, Stein E. On the numerical treatment and analysis of finite deformation ductile single crystal plasticity. *Comput Methods Appl Mech Eng*. 1996;129:235-254.
82. Glüge R, Kalisch J. The effective stiffness and stress concentrations of a multi-layer laminate. *Compos Struct*. 2014;111:580-586.
83. Wicht D, Schneider M, Böhlke T. An efficient solution scheme for small-strain crystal-elasto-viscoplasticity in a dual framework. *Comput Methods Appl Mech Eng*. 2020;358:112611.
84. Seemüller C, Heilmaier M, Haenschke T, Bei H, Dlouhý A, George EP. Influence of fiber alignment on creep in directionally solidified NiAl-10Mo in-situ composites. *Intermetallics*. 2013;35:110-115.
85. Nesterov Y. *Introductory Lectures on Convex Optimization: A Basic Course*. Mathematics and its Applications. Boston, MA: Kluwer Academic Publishers; 2004.
86. Dai YH. A perfect example for the BFGS method. *Math Program*. 2013;138(1–2):501-530.
87. Wriggers P. *Nonlinear Finite Element Methods*. Berlin, Germany: Springer; 2008.

How to cite this article: Wicht D, Schneider M, Böhlke T. On Quasi-Newton methods in fast Fourier transform-based micromechanics. *Int J Numer Methods Eng*. 2020;121:1665–1694. https://doi.org/10.1002/nme.6283