



A dynamical view of nonlinear conjugate gradient methods with applications to FFT-based computational micromechanics

Matti Schneider¹

Received: 27 December 2019 / Accepted: 6 April 2020 / Published online: 4 May 2020
© The Author(s) 2020

Abstract

For fast Fourier transform (FFT)-based computational micromechanics, solvers need to be fast, memory-efficient, and independent of tedious parameter calibration. In this work, we investigate the benefits of nonlinear conjugate gradient (CG) methods in the context of FFT-based computational micromechanics. Traditionally, nonlinear CG methods require dedicated line-search procedures to be efficient, rendering them not competitive in the FFT-based context. We contribute to nonlinear CG methods devoid of line searches by exploiting similarities between nonlinear CG methods and accelerated gradient methods. More precisely, by letting the step-size go to zero, we exhibit the Fletcher–Reeves nonlinear CG as a dynamical system with state-dependent nonlinear damping. We show how to implement nonlinear CG methods for FFT-based computational micromechanics, and demonstrate by numerical experiments that the Fletcher–Reeves nonlinear CG represents a competitive, memory-efficient and parameter-choice free solution method for linear and nonlinear homogenization problems, which, in addition, decreases the residual monotonically.

Keywords Homogenization · Micromechanics · FFT-based method · Nonlinear conjugate gradient scheme

1 Introduction

FFT-based solvers, as pioneered by Moulinec–Suquet [1,2], enjoy great popularity for solving homogenization problems on complex microstructures. Their success rests essentially on three pillars. First, they profit from modern imaging techniques like micro-computed tomography which naturally produce volumetric data on a regular grid. Secondly, the current implementations of the fast Fourier transform [3] are extremely powerful, providing programmers of FFT-based methods a cutting edge. Last but not least, FFT-based methods naturally support matrix-free solvers also for strongly nonlinear mechanical problems, which can be decisive if memory occupancy is an issue. A recent overview of applications in micromechanics is given in Matouš et al. [4], including the use in FE-FFT schemes [5] and as precomputations for model-order reduction methods [6–8] and deep learning techniques [9,10].

Progress in FFT-based methods focussed on three principal directions. Firstly, the areas of applicability were extended, for instance to finite strain problems [11], crystal viscoplasticity [12] and damage [13]. Recently, also applications to interface decohesion [14], upscaling of micromechanics-based fatigue [15] and the homogenization of brittle fracture [16] were established.

Secondly, the disadvantages of the original discretization method of Moulinec–Suquet were focus of investigations. Fourier–Galerkin methods [17] and finite element [18–20], finite difference [21,22] as well as finite volume discretizations [23] were investigated, some of them playing a crucial role when treating media with imperfections such as cracks and pores. Also, recently, a discretization in terms of B-splines was introduced [24].

Last but not least, the basic (solution) scheme was superseded by more powerful numerical solution methods. Eyre–Milton [25] introduced a so-called polarization scheme, which was subsequently generalized by Michel–Moulinec–Suquet [26]. Monchiet–Bonnet [27] introduced a one-parameter family of polarization schemes and exhibited both the Eyre–Milton scheme and the augmented Lagrangian method of Michel–Moulinec–Suquet as particular instances, see also Moulinec–Silva [28]. Furthermore, they introduced

✉ Matti Schneider
matti.schneider@kit.edu

¹ Institute of Engineering Mechanics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

a hybrid method which is essentially the average of the Eyre–Milton and the augmented Lagrangian operators. Schneider–Wicht–Böhlke [29] interpreted polarization schemes from the viewpoint of global optimization and identified them as Douglas–Rachford schemes [30], providing convergence guarantees and optimal parameter selection for strongly convex problems.

Zeman et al. [31] and Brisard–Dormieux [18] introduced Krylov subspace solvers for linear homogenization problems. Zeman et al. [31] proposed using the (linear) conjugate gradient method with great success. As the conjugate gradient method was applied to a non-symmetric operator, its successful application came as a surprise at that time (and triggered further research, see below). Brisard–Dormieux used a reformulation of the corrector equation in terms of the Hashin–Shtrikman variational principle, guaranteeing a symmetric, but possibly indefinite formulation. They proposed using the conjugate gradient method or MINRES and SYMMLQ, respectively, see Paige–Saunders [32].

As an alternative to the basic scheme and polarization-type methods, Newton solvers were proposed by Lahellec et al. [11]. Initially, the basic scheme was used as the linear solver. Only much later, Gélébart–Mondon–Cancel [33] and Kabel–Böhlke–Schneider [34] proposed Newton–Krylov solvers, ie, to use Krylov subspace solvers for computing the Newton increment. For a thorough study on line-search methods suitable in the FFT context and strategies for choosing the forcing term, we refer to Wicht–Schneider–Böhlke [35].

Kabel–Böhlke–Schneider [34] observed that the basic scheme may be interpreted as a gradient descent scheme, and connected the reference material to (the inverse of) a step-size. By this observation, numerical solution methods from large-scale optimization could enter into FFT-based computational homogenization. The benefits of using fast gradient methods in the FFT-based context were first exploited in Schneider [36], see also Ernesti–Schneider–Böhlke [37] for an application of fast gradient methods to phase-field fracture. Using Quasi-Newton methods was pioneered by Shantraj et al [38], and further extended in Wicht–Böhlke–Schneider [35].

All of these solvers have their distinctive advantages. The basic scheme is most memory-efficient and extremely robust, but may become slow for highly contrasted problems. The polarization schemes fail to converge quickly for problems involving pores (see Schneider [39] for a counterexample) and rely upon a complexity-reduction trick to be efficient, see Schneider–Wicht–Böhlke [29]. Newton methods are fast and robust, but suffer from heavy memory demands. They are particularly useful if evaluating the nonlinear material law is very expensive compared to applying the tangent. The fast gradient methods combine fast conver-

gence with a low memory footprint. However, they suffer from a delicate parameter selection. More precisely, they require the strong convexity constant to be known to be efficient. However, determining the strong convexity constant for problems involving pores is extremely difficult, in general. Restarting techniques do not solve these problems. Indeed, for problems where the strong convexity constant is known, restarting schemes require about twice the iteration count as if the method was used with optimal parameter choice.

Thus, research was devoted to finding solution techniques that share the positive characteristics of the mentioned solution schemes without combining their disadvantages. Schneider [39] proposed the Barzilai–Borwein method as a general-purpose method for computational micromechanics. It may be interpreted as the basic scheme with adaptive time-stepping (or, equivalently, adaptive reference material). Thus, it has low memory-footprint, but is competitive in terms of convergence speed when compared to the solvers mentioned above. However, the Barzilai–Borwein method is an intrinsically non-monotone method, ie, the residual is not monotonically decreasing from one iteration to the next. This behavior is unfamiliar, and may limit the applicability for industrial applications. It may happen that after the lunch break the residual is two orders of magnitude higher than before the lunch break.

Thus, it is desirable to seek alternatives to the Barzilai–Borwein scheme with guaranteed monotonicity properties.

Contributions

Nonlinear conjugate gradient methods were introduced by Fletcher–Reeves [40] as a generalization of the linear conjugate gradient scheme [41]. Nonlinear conjugate gradient methods were among the first general-purpose solution schemes for unconstrained optimization problems, and have been studied thoroughly, see Nocedal–Wright [42] for a comprehensive textbook treatment. They are based upon an iteratively updated search direction, involving a coefficient (the “ β ”) to be chosen wisely and a step-size which is typically determined by line search. For the β -coefficient, a number of different variants were proposed, the most popular due to Fletcher–Reeves [40], Polak–Ribière–Polyak [43,44], Hestenes–Stiefel [41] and Dai–Yuan [45].

On a formal level, nonlinear conjugate gradient methods and the heavy-ball scheme [46], a very powerful fast gradient scheme, are identical, see Polyak’s book [47]. However, the parameters are chosen in a different way. Indeed, for the heavy-ball method, the step-size and the β -coefficient are typically chosen as constants depending on the properties of the function (the Lipschitz constant of the gradient of the objective, and the strong monotonicity constant, if available [44,48,49]). In contrast, nonlinear conjugate gradient

methods choose the β -coefficient depending on the current and the last iterate, in general, and determine the step-size by a line-search procedure. We refer to Sect. 2 for more details.

A naive application of nonlinear conjugate gradient methods in the FFT-based context does not lead to competitive schemes, essentially due to the intrinsic line-search. (Another difficulty arising in computational micromechanics is the fact that the condensed incremental energy is typically not computed - only gradient information, ie, the stress, is evaluated.) In the nonlinear conjugate gradient community, this problem is well-known. Indeed, Quasi-Newton methods of the Broyden class, see chapter 8 in Nocedal–Wright [42], may also be interpreted as generalizations of linear CG. However, they typically outperform nonlinear conjugate gradient methods because they asymptotically accept the proposed step-size—in contrast, the most efficient conjugate gradient methods are reported to require about two line-search steps per iteration to be efficient. Sun–Zhang [50] pioneered using conjugate gradient methods without line search and proved convergence under mild conditions. However, these step-sizes are too conservative for the resulting methods to be efficient. Indeed, under suitable conditions, they ensure a monotonic decrease of the objective value, which is known to be inefficient for fast gradient methods.

In this work, we connect nonlinear conjugate gradient methods to fast gradient methods by letting the step-size go to zero, trying to understand the resulting dynamical system. Indeed, the Fletcher–Reeves CG permits such a limiting procedure, giving rise to Newtonian dynamics with nonlinear damping. More precisely, the damping is determined by the speed of decrease of the residual (the norm of the gradient). In theory, this damping could become negative, inducing an instability of the dynamical system. In the numerical experiments, however, this situation did not arise frequently, whence we discarded including proper safeguarding strategies into the manuscript.

We propose to combine the Fletcher-Reeves nonlinear CG with the step-size of the basic scheme, and detail its implementation in the FFT-based context, see Sect. 3. The method may be implemented on three strain-like fields, which is only one more than the Barzilai–Borwein method. However, the resulting solver leads to a monotonic, and thus reliable, decrease of the residual. We demonstrate the usefulness of the proposed method as a general-purpose solver for FFT-based computational micromechanics in Sect. 4, comparing the method to a multitude of state-of-the-art numerical solvers for problems of industrial scale.

2 The dynamics of nonlinear conjugate gradients

2.1 From linear to nonlinear conjugate gradient methods

Let V be a Hilbert space with inner product $\langle \cdot, \cdot \rangle$, and let $A \in L(V)$ be a bounded linear operator, which we assume self-adjoint and uniformly positive, ie, $\langle Au, v \rangle = \langle Av, u \rangle$ holds for all $u, v \in V$ and there is $\mu > 0$, s.t. $\langle Au, u \rangle \geq \mu \|u\|^2$ holds for all $u \in V$, where $\|u\| = \sqrt{\langle u, u \rangle}$ denotes the induced norm. For prescribed $b \in V$, we are concerned with minimizing the quadratic objective

$$V \ni x \mapsto f(x) = \frac{1}{2} \langle x, Ax \rangle - \langle b, x \rangle, \tag{2.1}$$

whose unique minimizer x^* solves the linear equation $Ax^* = b$, ie, $x^* = A^{-1}b$.

An ordered set of vectors (d_0, d_1, d_2, \dots) is called A -conjugate (or simply conjugate if A is fixed), if $d_i \neq 0$ for all i and $\langle d_i, Ad_j \rangle = 0$ for all $i \neq j$. Put differently, A -conjugacy is equivalent to orthogonality w.r.t. the A -weighted inner product $(u, v) \mapsto \langle u, Av \rangle$ on V .

For a collection (d_0, d_1, d_2, \dots) of A -conjugate vectors and an initial point $x_0 \in V$, the conjugate direction method (Sect. 5.1 in Nocedal–Wright [42]) iteratively updates

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, 2, \dots, \tag{2.2}$$

where the step size α_k is chosen by exact line search

$$\alpha = \operatorname{argmin}_{\alpha \in \mathbb{R}} f(x_k + \alpha d_k), \tag{2.3}$$

which can be computed explicitly for the quadratic objective (2.1) as

$$\alpha_k = -\frac{\langle g_k, d_k \rangle}{\langle d_k, Ad_k \rangle}, \quad \text{where } g_k = \nabla f(x_k) \equiv Ax_k - b. \tag{2.4}$$

If V has finite dimension n , any n conjugate directions form a basis of V , and the conjugate direction method converges to x^* after at most n iterations. More generally, if the collection $\{d_0, d_1, d_2, \dots\}$ forms a complete basis of the Hilbert space V , the conjugate direction method converges.

The conjugate gradient method of Hestenes–Stiefel [41] is a conjugate direction method where the conjugate directions are generated by applying the Gram–Schmidt procedure to the negative gradients

$$d_k = -g_k + \sum_{i=0}^{k-1} \frac{\langle g_k, Ad_i \rangle}{\langle d_i, Ad_i \rangle} d_i, \quad g_k = \nabla f(x_k) \equiv Ax_k - b.$$

The latter formula may be simplified, taking into account the update (2.2) and the conjugacy to arrive at

$$d_k = -g_k + \beta_{k-1}d_{k-1}, \quad \beta_{k-1} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \quad k = 1, 2, \dots \tag{2.5}$$

We formulate this version of the linear conjugate gradient algorithm for later reference

$$\begin{aligned} g_k &= Ax_k - b \\ \beta_{k-1} &= \begin{cases} 0, & k = 0, \\ \frac{\|g_k\|^2}{\|g_{k-1}\|^2} & k > 0. \end{cases} \\ d_k &= -g_k + \beta_{k-1}d_{k-1} \\ \alpha_k &= -\frac{\langle g_k, d_k \rangle}{\langle d_k, Ad_k \rangle} \\ x_{k+1} &= x_k + \alpha_k d_k \end{aligned} \tag{2.6}$$

and formally defining $d_{-1} = 0$. Fletcher–Reeves [40] extended this algorithm to general unconstrained nonlinear optimization problems

$$f(x) \rightarrow \min_{x \in V} \tag{2.7}$$

in the form

$$\begin{aligned} d_k &= -\nabla f(x_k) + \beta_{k-1}d_{k-1}, \\ x_{k+1} &= x_k + \alpha_k d_k, \end{aligned} \tag{2.8}$$

where α_k is determined by a line search procedure (2.3), and

$$\beta_k^{FR} = \frac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2}, \tag{2.9}$$

where the superscript FR stands for Fletcher–Reeves. Subsequently, to account for inaccurate line-search procedures, a variety of other formulae for β_k were introduced, the most popular are the following.

1. The formula introduced, independently, by Polak–Ribière [43] and Polyak [44],

$$\beta_k^{PRP} = \frac{\langle g_{k+1}, g_{k+1} - g_k \rangle}{\|g_k\|^2}, \quad g_i = \nabla f(x_i). \tag{2.10}$$

Due to a counter-example exhibiting divergence even for exact line search [51], Gilbert–Nocedal [52] introduced a modification

$$\beta_k^{PRP+} = \max \left\{ \beta_k^{PRP}, 0 \right\}. \tag{2.11}$$

2. The formula

$$\beta_k^{HS} = \frac{\langle g_{k+1}, g_{k+1} - g_k \rangle}{\langle d_k, g_{k+1} - g_k \rangle} \tag{2.12}$$

which is the form of β_k in the original work of Hestenes–Stiefel [41]. This variant ensures that the vectors d_k and d_{k-1} are A_k -conjugate for the averaged Hessian

$$A_k = \int_0^1 \nabla^2 f(x_k + s\alpha_k d_k) ds$$

between successive iterates.

3. Dai–Yuan [45] introduced the formula

$$\beta_k^{DY} = \frac{\|g_{k+1}\|^2}{\langle d_k, g_{k+1} - g_k \rangle}, \tag{2.13}$$

which ensures that d_k is always a descent direction, ie, $\langle d_k, g_k \rangle < 0$ holds, provided f is convex and any α_k [53] or for general f if α_k satisfies the Wolfe conditions [45].

The different formulae for β_k coincide for quadratic objective (2.1) and exact line search (2.4). For line search, typically either the strong Wolfe conditions

$$\begin{aligned} f(x_k) - f(x_k + \alpha_k d_k) &\geq -\delta \alpha_k \langle g_k, d_k \rangle, \\ |\langle \nabla f(x_k + \alpha_k d_k), d_k \rangle| &\leq -\sigma \langle g_k, d_k \rangle, \end{aligned} \tag{2.14}$$

or the Wolfe conditions

$$\begin{aligned} f(x_k) - f(x_k + \alpha_k d_k) &\geq -\delta \alpha_k \langle g_k, d_k \rangle, \\ \langle \nabla f(x_k + \alpha_k d_k), d_k \rangle &\geq \sigma \langle g_k, d_k \rangle, \end{aligned} \tag{2.15}$$

involving parameters $0 < \delta < \sigma < 1$, are enforced. (See Sect. 3.1 in Nocedal–Wright [42] for details.) For general optimization problems and proper line-search conditions, convergence theorems may be established, see Dai–Yuan [45] and Hager–Zhang [54] for an overview. However, the results are typically of the form

$$\lim_{k \rightarrow \infty} \|g_k\| = 0 \quad \text{or} \quad \liminf_{k \rightarrow \infty} \|g_k\| = 0,$$

ie, no explicit convergence rate may be deduced, in general.

The biggest limitation of the nonlinear conjugate gradient methods is the line search involved in the update (2.8). This becomes apparent even for the quadratic case (2.6). In the presented form, two applications of A (ie, gradient evaluations of f) are required for the linear case. Of course, introducing $z_k = Ad_k$ and noticing $g_{k+1} = g_k + \alpha_k z_k$, the standard implementation of linear CG (Sect. 5.1 in Nocedal–Wright [42]) may be obtained, which only requires one application of A per iteration. However, if we apply the nonlinear CG method to a quadratic function using the exact line search

again requires two applications of A per iteration. Thus, if we assume that the effort of applying A dominates the computational expense per iteration, the nonlinear CG is about a factor of two slower than linear CG applied to quadratic problems (2.1).

For general nonlinear optimization problems, state-of-the-art implementations of the nonlinear conjugate gradient methods require two gradient evaluations per line search (see the introduction of Dai–Kou [55]). In contrast, (limited memory) Quasi-Newton methods [56–60] asymptotically accept the suggested step size. Thus, they often outperform conjugate gradient methods in practice.

For these reasons, research effort has been invested in nonlinear conjugate gradient methods avoiding line search at all. Sun–Zhang [50] have shown that coming close to the upper bound in the Armijo condition [the first inequality in the Wolfe conditions (2.15)] leads to convergence for the different variants of β_k . Also, combinations of nonlinear CG and the Barzilai–Borwein method have been explored [55, 61].

In general, the Armijo condition ensures a monotonic decrease of the function value (provided d_k is a descent direction). We will see that this is not optimal.

2.2 The heavy-ball method versus nonlinear conjugate gradients

For minimizing general (differentiable) objective functions (2.7), the heavy-ball method [46] uses the iterative scheme

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) + \gamma_k (x_k - x_{k-1}), \tag{2.16}$$

involving step-sizes α_k and momentum coefficients γ_k . The heavy-ball scheme (2.16) may be motivated, see Helmke–Moore [62], as a time-discretized version of the linearly damped Newtonian dynamics¹

$$\ddot{x} + b \dot{x} = -\nabla f(x) \tag{2.17}$$

for a curve $x : [0, \infty) \rightarrow V$ and damping parameter $b \geq 0$. A backward Euler discretization of the dynamics (2.17) with time-step size $h > 0$ reads

$$\frac{x_{k+1} - 2x_k + x_{k-1}}{h^2} + b \frac{x_k - x_{k-1}}{h} = -\nabla f(x_k),$$

which may be rearranged into the scheme

$$x_{k+1} = x_k - h^2 \nabla f(x_k) + (1 - bh)(x_k - x_{k-1}), \tag{2.18}$$

which has the form (2.16) with $\alpha_k \equiv h^2$ and $\beta_k \equiv 1 - bh$.

¹ Formally, the “mass” in front of the acceleration is set to unity.

Initially, Polyak investigated the local convergence of the scheme (2.16) (for strongly convex objective functions) by Lyapunov’s first method. However, the determined optimal parameters failed to be stable for general strongly convex objectives with Lipschitz gradient, see Lessard–Recht–Packard [49] and Ghadimi–Feysmahdavian–Johansson [48] for counterexamples.

A global convergence analysis was carried out based on Lyapunov’s second method, ie, in terms of suitable Lyapunov functions, see the pioneering work Zavriev–Kostyuk [63]. More precisely, in the continuous setting, the total energy of the dynamical system (2.17)

$$V(x, p) = f(x) + \frac{1}{2} \|p\|^2 \quad \text{with } p = \dot{x} \tag{2.19}$$

may be used as a Lyapunov function, ie,

$$\frac{d}{dt}[V(x, \dot{x})] = -b \|\dot{x}\|^2$$

holds. Put differently, if $b \geq 0$, the dynamics is stable. For $b > 0$, the function V decreases strictly along the trajectories. Time-discretized variants of the Lyapunov function (2.19) were used [48,63,64] to establish global convergence of the heavy-ball method for particular choices of α_k and γ_k , and different classes of objective functions f .

Siegel [65] used Lyapunov-function arguments to show that, for μ -strongly convex functions f , choosing $b = 2\sqrt{\mu}$ leads to the estimate

$$f(x(t)) - f(x^*) \leq 2e^{-\sqrt{\mu}t} (f(x_0) - f(x^*)), \tag{2.20}$$

where x^* denotes the minimizer of f .

It has been long known that nonlinear conjugate gradient methods (2.8) and heavy-ball schemes (2.16) coincide as numerical schemes. Indeed, for the conjugate gradient update (2.8)

$$d_k = -\nabla f(x_k) + \beta_{k-1} d_{k-1}, \quad x_{k+1} = x_k + \alpha_k d_k,$$

combining these two equations yields

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) + \frac{\alpha_k \beta_{k-1}}{\alpha_{k-1}} (x_k - x_{k-1}),$$

ie, it is of heavy-ball form (2.16) with $\gamma_k = \frac{\alpha_k \beta_{k-1}}{\alpha_{k-1}}$.

Although they formally coincide, the way the numerical parameters are chosen is different. Also, both the respective interpretations and the convergence theories differ. For the heavy-ball method, the parameters α_k and γ_k are chosen independently of k , but depending on the strong convexity constant of the objective function and the Lipschitz constant of its gradient. More precisely, the Lipschitz constant L and

the strong monotonicity constant μ of the gradient of f are required. Then, the convergence of the heavy-ball method is deduced by Lyapunov’s first or second method, deducing local [46,49] or global [48,64] convergence, respectively. The convergence results are typically very strong, involving a q -linear convergence towards the minimum. Getting hands on the strong monotonicity constant μ is difficult, in general, and for homogenization problems of Sect. 3, in particular. This constitutes the biggest limitation of the heavy-ball method.

In contrast, for nonlinear conjugate gradient methods, the parameters α_k and β_k are chosen adaptively. The different formulae for β_k measure the local geometry of the function, and the step-size α_k is chosen by a line-search procedure. Notice that the bounds entering the Wolfe conditions (2.15) and (2.14) also depend on local information about the function f . Convergence of the conjugate gradient method can be typically established for a large class of objective functions (see, for instance, Dai [66]), not necessarily restricted to convex objective functions. However, the convergence results are typically rather weak, for instance $\|\nabla f(x_{k_n})\| \rightarrow 0$ along a subsequence. As already mentioned in the previous section, the biggest limitation of nonlinear conjugate gradient methods is their reliance upon line-search procedures.

2.3 A dynamical perspective on conjugate gradient methods

We have seen that heavy-ball methods work well for fixed step-size, but the momentum parameter needs tweaking. Fixing the step-size $\alpha_k \equiv \alpha$ in the nonlinear conjugate gradient update (2.8) leads to $\gamma_k = \beta_k$ in the heavy-ball scheme (2.16) and, thus, to the method

$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta_k(x_k - x_{k-1}).$$

Reversing the arguments leading to Eq. (2.18) yields, for $\alpha = h^2$

$$\frac{x_{k+1} - 2x_k + x_{k-1}}{h^2} + \frac{1 - \beta_k}{h} \frac{x_k - x_{k-1}}{h} = -\nabla f(x_k).$$

Thus, provided the limit $\frac{1-\beta_k}{h}$ as $h \rightarrow 0$ exists, we get the corresponding dynamical system

$$\ddot{x} + b(x, \dot{x}) \dot{x} = -\nabla f(x) \tag{2.21}$$

which is formally similar to Eq. (2.17), but with a state-dependent damping. However, not all formulae for β_k of Sect. 2.1 give rise to a dynamical system. (For instance, the Polak–Ribière–Polyak formula does not satisfy $\beta_k^{\text{PRP}} \rightarrow 1$ as $h \rightarrow 0$.) For the Fletcher–Reeves CG, $\beta_k^{\text{FR}} = \frac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2}$, we get

$$\frac{1 - \beta_k^{\text{FR}}}{h} = -\frac{1}{h} \frac{\|\nabla f(x_{k+1}) - \|\nabla f(x_k)\|^2}{\|\nabla f(x_k)\|^2} \rightarrow -\frac{d}{dt} \left[\frac{\|\nabla f(x)\|^2}{\|\nabla f(x)\|^2} \right] \text{ as } h \rightarrow 0,$$

ie, the “damping” term in Eq. (2.21) becomes

$$b(x, \dot{x}) = -2 \frac{\langle \nabla f(x), \nabla^2 f(x) \dot{x} \rangle}{\|\nabla f(x)\|^2}. \tag{2.22}$$

Thus, we arrive at the dynamical system

$$\ddot{x} - 2 \frac{\langle \nabla f(x), \nabla^2 f(x) \dot{x} \rangle}{\|\nabla f(x)\|^2} \dot{x} = -\nabla f(x) \tag{2.23}$$

with initial conditions $x(0) = x_0$ and $\dot{x}(0) = -\nabla f(x_0)$. Several remarks are in order.

1. The Fletcher–Reeves dynamical system is similar to the dynamical system associated to Nesterov’s fast gradient method [67,68], as developed by Su–Boyd–Candes [69],

$$\ddot{x} + \frac{3}{t} \dot{x} = -\nabla f(x), \quad t \geq 0, \tag{2.24}$$

where $\frac{3}{t}$ serves as a time-dependent damping. For convex f , a $1/t^2$ -convergence rate of the objective along a trajectory of the ODE (2.24) may be established using Lyapunov-type arguments.

To highlight the similarity with the Fletcher–Reeves–ODE (2.23), we rewrite the latter in the form

$$\ddot{x} + \frac{d}{dt} \left[-\ln \|\nabla f(x)\|^2 \right] \dot{x} = -\nabla f(x), \tag{2.25}$$

whereas we may cast (2.24) in the form

$$\ddot{x} + \frac{d}{dt} \left[-\ln \frac{1}{t^3} \right] \dot{x} = -\nabla f(x), \quad t > 0. \tag{2.26}$$

Thus, for both Eqs. (2.25) and (2.26), a logarithmic barrier function, see Sect. 4.2 in Nesterov [70], determines the damping coefficient, leading to a damping when approaching the critical set and $t = +\infty$, respectively.

Also, for both ODEs (2.25) and (2.26), the trajectories are invariant w.r.t. a homogeneous rescaling of time.

2. The stability analysis of the dynamical system (2.18) using the Lyapunov function V from Eq. (2.19) may be repeated without change, leading to

$$\frac{d}{dt} [V(x, \dot{x})] = -b(x, \dot{x}) \|\dot{x}\|^2. \tag{2.27}$$

We see that the dynamical system is stable provided b is non-negative. In turn, b from Eq. (2.22) is non-negative

if and only if

$$2\langle \nabla f(x), \nabla^2 f(x)\dot{x} \rangle \equiv \frac{d}{dt} \left[\|\nabla f(x)\|^2 \right] \leq 0$$

holds. Put differently, as long as the residual $\|\nabla f(x)\|^2$ is non-increasing, the Fletcher–Reeves dynamical system (2.23) is stable. Clearly, due to the initial condition $(x(0), \dot{x}(0)) = (x_0, -\nabla f(x_0))$ with $\nabla f(x_0) \neq 0$, the residual is decreasing for convex functions f and small times $t > 0$. Indeed,

$$b(x_0, -\nabla f(x_0)) = 2 \frac{\langle \nabla f(x_0), \nabla^2 f(x_0)\nabla f(x_0) \rangle}{\|\nabla f(x_0)\|^2}$$

holds. Thus, $b(x_0, -\nabla f(x_0))$ is non-negative if $\nabla^2 f(x_0)$ is positive semi-definite (which is true if f is convex).

In our numerical experiments, b remained positive (except for very few iterates). However, we did not find a mathematical proof for this fact. Various safeguarding strategies (preventing b from becoming negative) never led to an improvement of the convergence behavior, whence we discarded discussing them from the manuscript.

3. In view of Siegel’s convergence result (2.20), fast convergence is to be expected if $b \approx 2\sqrt{\mu}$. b may be considered as a feedback controller for the dynamical system (2.21). Its analysis goes beyond this manuscript.
4. For some other nonlinear conjugate gradient methods, corresponding dynamical systems may be derived as well. For instance, the Dai–Yuan method [45] leads to the dynamical system

$$\ddot{x} + \frac{d}{dt} [-\ln -\langle \nabla f(x), \dot{x} \rangle] \dot{x} = -\nabla f(x).$$

However, the latter ODE is degenerate, and we did not investigate it further.

5. The linear conjugate gradient method was interpreted as a dynamical system with feedback control in Bhaya-Kaszukiewicz [71]. Their treatment is different from ours, as they consider discrete dynamical systems and discrete controls, and does not directly generalize to nonlinear CG.

3 Application to micromechanical problems and implementation in the FFT framework

In a small-strain setting, suppose a heterogeneous, possibly nonlinear elastic energy $w : Y \times \text{Sym}(d) \rightarrow \mathbb{R}$ is given, where $Y \subseteq \mathbb{R}^d$ denotes a rectangular cell in d dimensions and $\text{Sym}(d)$ stands for the linear space of symmetric $d \times d$ -matrices. Typically, w is differentiable in the second variable,

and has jumps in the first, the spatial, variable. The latter framework also encompasses inelastic problems, as w may stand for the condensed incremental potential corresponding to a fixed time step, see Ortiz–Stainier [72] and Miehe [73].

For prescribed macroscopic $E \in \text{Sym}(d)$, we seek a periodic displacement field $u : Y \rightarrow \mathbb{R}^d$ solving the balance of linear momentum on the microscale

$$\text{div } \sigma(E + \nabla^s u) = 0, \tag{3.1}$$

where ∇^s denotes the symmetrized gradient, div stands for the divergence, and σ is the stress operator derived from the potential w ,

$$\sigma(\varepsilon) = \frac{\partial w}{\partial \varepsilon}(\cdot, \varepsilon) \quad \text{for } \varepsilon : Y \rightarrow \text{Sym}(d).$$

After solving Eq. (3.1) for u , the effective stress is computed by volume averaging

$$\Sigma = \langle \sigma(E + \nabla^s u) \rangle_Y \equiv \frac{1}{|Y|} \int_Y \sigma(E + \nabla^s u) \, dx.$$

To apply nonlinear conjugate gradient methods, we recast the balance of linear momentum (3.1) as an optimization problem (2.7). We set $V = H^1_{\#}(Y; \mathbb{R}^d)$, the first order L^2 -Sobolev space of periodic vector fields with vanishing mean, and

$$f(u) = \langle w(E + \nabla^s u) \rangle_Y, \quad u \in H^1_{\#}(Y; \mathbb{R}^d), \tag{3.2}$$

which is well-defined under technical conditions, see Schneider [74]. We notice that the first order optimality condition of the functional (3.2) is precisely the balance of linear momentum (3.1).

For the Hilbert space $V = H^1_{\#}(Y; \mathbb{R}^d)$, we fix the inner product

$$(u, v) \mapsto \langle u, v \rangle = \frac{1}{|Y|} \int_Y \nabla^s u : \nabla^s v \, dx \equiv \langle \nabla^s u : \nabla^s v \rangle_Y, \tag{3.3}$$

where $:$ denotes the double contraction of tensor indices. Recall that, for a continuously differentiable function f on a Hilbert space, the differential $Df(u)$, for $u \in V$, is an element of the (continuous) dual Hilbert space, ie, $Df(u) \in V'$. The gradient of f at u is defined implicitly via

$$\langle \nabla f(u), v \rangle = Df(u)[v] \quad \text{for all } v \in V.$$

Thus, $\nabla f(u) \in V$, but is dependent on the chosen inner product on V . (In contrast, for two equivalent inner products

on a fixed Hilbert space, the corresponding differentials are identical.) More explicitly, the gradient may be written as

$$\nabla f(u) = RDf(u),$$

where $R : V' \rightarrow V$ denotes the Riesz isomorphism, the inverse of the mapping $V \rightarrow V', u \mapsto \langle u, \cdot \rangle$.

In the homogenization context (3.2), we have

$$\langle Df(u), v \rangle = \langle \sigma(E + \nabla^s u) : \nabla^s v \rangle_Y,$$

which we may write more succinctly in the form

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

in terms of the “natural” pairing

$$V' \times V \rightarrow \mathbb{R}, \quad (g, u) \mapsto \frac{1}{|Y|} \int_Y g \cdot u \, dx.$$

The Riesz map for the inner product (3.3), in turn, becomes $R = -G$ in terms of the Green’s function $G = (\text{div } \nabla^s)^{-1}$, which is well-defined on $H^1_\#(Y; \mathbb{R}^d)$ (due to the mean-value constraint).

With these identifications, for given initial vector u_0 and $d_{-1} = 0$, the nonlinear conjugate gradient update (2.8) becomes

$$\begin{aligned} d_k &= -G \text{div } \sigma(E + \nabla^s u_k) + \beta_{k-1} d_{k-1}, \\ u_{k+1} &= u_k + \alpha_k d_k. \end{aligned}$$

For FFT-based methods, it is convenient to introduce the total strain $\varepsilon_k = E + \nabla^s u_k$ and $D_k = \nabla^s d_k$, and to rewrite the latter update in the form

$$\begin{aligned} D_k &= -\Gamma : \sigma(\varepsilon_k) + \beta_{k-1} D_{k-1}, \\ \varepsilon_{k+1} &= \varepsilon_k + \alpha_k D_k, \end{aligned} \tag{3.4}$$

using the operator $\Gamma = \nabla^s G \text{div}$, an orthogonal projector on $L^2(Y; \text{Sym}(d))$ (see Milton [75]). Upon choosing $\beta_k \equiv 0$ in Eq. (3.4), we recover the basic scheme of Moulinec-Suquet

$$\varepsilon_{k+1} = \varepsilon_k - \alpha_k \Gamma : \sigma(\varepsilon_k)$$

with variable step-size, confirming the observations in Schneider [74]. Specializing to the Fletcher–Reeves CG (2.9), we may write

$$\begin{aligned} G_k &= \Gamma : \sigma(\varepsilon_k) \\ \beta_{k-1} &= \frac{\|G_k\|^2}{\|G_{k-1}\|^2} \\ D_k &= -G_k + \beta_{k-1} D_{k-1}, \\ \varepsilon_{k+1} &= \varepsilon_k + \alpha_k D_k. \end{aligned} \tag{3.5}$$

Discretizing the balance of linear momentum (3.1), see Moulinec–Suquet [1,2], Willot [22] or Schneider–Ospald–Kabel [21], leads to discretized systems of equations (with corresponding discretized Γ -operators) that may be solved using nonlinear CG (3.5). We summarized the implementation in algorithm 1. Several remarks are in order.

Algorithm 1 Fletcher–Reeves nonlinear CG (maxit, tol)

```

1: Determine initial guess  $\varepsilon$  and initial reference material  $C^0$ 
2: residual  $\leftarrow 1$ 
3:  $k \leftarrow 0$ 
4:  $\gamma \leftarrow 1$ 
5:  $D \leftarrow 0$ 
6: while  $k < \text{maxit}$  and residual  $> \text{tol}$  do
7:    $k \leftarrow k + 1$ 
8:    $G \leftarrow \sigma(\varepsilon)$ 
9:    $\langle \sigma \rangle \leftarrow \langle G \rangle_Y$ 
10:   $\hat{G} \leftarrow \text{FFT}(G)$ 
11:   $\hat{G}(\xi) \leftarrow \begin{cases} E, & \xi = 0 \\ \Gamma^0(\xi) \hat{G}(\xi), & \text{else} \end{cases}$ 
12:   $G \leftarrow \text{FFT}^{-1}(\hat{G})$ 
13:   $\gamma_{old} \leftarrow \gamma$ 
14:   $\gamma \leftarrow \|G\|^2$ 
15:   $\beta \leftarrow \gamma / \gamma_{old}$ 
16:   $D \leftarrow -G + \beta D$ 
17:   $\varepsilon \leftarrow \varepsilon + \alpha D$ 
18:  residual  $\leftarrow \sqrt{\gamma} / \|\langle \sigma \rangle\|$ 
19: end while
20: return  $\varepsilon$ , residual

```

1. To complete the description of the Algorithm 1, a step-size strategy needs to be prescribed. We use the step-size of the basic scheme (see Schneider [74] for the nonlinear convex case), ie, if c_- and c_+ denote the smallest and largest eigenvalue of the material tangents $\frac{\partial \sigma}{\partial \varepsilon}$, evaluated over all microscopic points and admissible strains, we set $\alpha = \frac{2}{c_- + c_+}$.
2. In the presented form, the algorithm can be implemented on three strain-like fields (ε, G, D) . In contrast to the Barzilai–Borwein scheme [39] and the fast gradient schemes [74], an additional field is necessary for evaluating β_k .
3. The other nonlinear conjugate gradient methods (2.10), (2.12) and (2.13) may be implemented in a similar way. However, they require storing (at least) an additional vector, as $D_k - D_{k-1}$ enters the formulae for β_k .
4. If the memory footprint is decisive, reduced-memory implementations may be used, see for instance Grimm–Strehle–Kabel [76].
5. Incorporating mixed boundary conditions is straightforward, see Kabel–Flieger–Schneider [77].

4 Numerical investigations

4.1 Setup and materials

The nonlinear conjugate gradient method, cf Alg. 1, was integrated into an in-house FFT-based computational micromechanics code (written in Python with Cython extensions). For all but the MMC example, we use the staggered-grid discretization [21]. For the MMC example, in consistence with Schneider–Wicht–Böhlke [29], we rely upon the original spectral discretization of Moulinec–Suquet [1,2,78].

As the convergence criterion, we use

$$\frac{\|\Gamma : \sigma(\varepsilon^k)\|}{\|(\sigma(\varepsilon^k))_Y\|} \leq \text{tol} \quad (4.1)$$

for prescribed tolerance tol , see Schneider–Wicht–Böhlke [29], Sect. 5. (Notice $\Sigma \equiv 0$ in (5.7) for strain-driven uniaxial extension.) For the algorithm 1, the quantity (4.1) is computed in line 18. We set $\text{tol} = 10^{-5}$ for all examples in this section, unless explicitly mentioned otherwise.

Affine extrapolation [2] is used to generate a good initial guess for the subsequent load step. The linear elastic computations were carried out on a Laptop with 16 GB RAM and a dual core Intel i7 CPU. For the inelastic computations, a desktop computer with a 6-core Intel i7 CPU and 32GB RAM was utilized.

All used material parameters are collected in Table 1. The meaning of the inelastic parameters is specified in the proper subsection.

4.2 A sandcore microstructure

As our first example, we consider a microstructure of inorganically bound sand for casting applications, which was already investigated in Schneider [39]. It consists of 64 sand grains with 58.58% volume fraction, held together by 1.28% binder, cf Fig. 1a. The geometry was generated by the algorithm described in the work of Schneider et al. [84] and resolved by 256^3 voxels. The material parameters used for the grains and the binder are given in Table 1.

We apply uniaxial extension in x -direction (for 5% axial strain), see Fig. 1b, where the norm of the strain field is shown. For more details about digital sand-core physics, we refer to the work of Etemeyer et al [85]. We compare several solution methods for this porous example. First, we compare the different conjugate gradient methods, cf Fig. 2a. The linear conjugate gradient method [31,41] serves as our reference. We investigate the performance of the nonlinear conjugate gradient methods of Fletcher–Reeves (2.9), Dai–Yuan (2.13), Hestenes–Stiefel (2.12) and Polyak–Ribière–Polyak (2.10) for constant step-size (chosen according to the basic scheme). If exact line search was used,

all these conjugate gradient methods would lead to identical iterates.

From Fig. 2a we see that the Fletcher–Reeves and Dai–Yuan methods require about 50% more iterations than linear CG. The Dai–Yuan method is only slightly slower than Fletcher–Reeves, but exhibits a distinct zig-zagging from one iteration to the next. In contrast, the decrease of the residual is monotonic for the Fletcher–Reeves method.

The two other investigated nonlinear conjugate gradient methods, Hestenes–Stiefel and Polyak–Ribière–Polyak are much slower for this scenario. Actually, their convergence behavior is very close to the basic scheme in Fig. 2b. This is no coincidence, as we observe $\beta_k^{\text{HS}} \rightarrow 0$ and $\beta_k^{\text{PRP}} \rightarrow 0$ as $k \rightarrow \infty$.

In addition to the conjugate gradient method, other popular FFT-based solvers were applied to the problem at hand, see Fig. 2b. The Broyden–Fletcher–Goldfarb–Shanno algorithm [56–59] may be considered as another generalization of the linear CG method, as it reduces to the linear CG method when applied to quadratic problems with exact line search. Here, we investigate the limited-memory version of BFGS, as introduced by Nocedal [60], see Wicht–Schneider–Böhlke [35] for details in the FFT-based context. For a depth of 4, L-BFGS exhibits a convergence behavior that almost coincides with Fletcher–Reeves CG, however at a much higher computational cost per iteration.

The Barzilai–Borwein method [86] is also investigated, as it serves as a reference solver in FFT-based computational micromechanics [39]. The Barzilai–Borwein method exhibits a strongly non-monotonic convergence behavior, as can also be seen in Fig. 2b. However, the method can be shown to exhibit local r -linear convergence [87] when applied to uniformly convex optimization problems. As we are only interested in the smallest k , s.t. the convergence criterion (4.1) is fulfilled, the non-monotonicity may be non-critical. Concerning the number of iterations to terminate, the Barzilai–Borwein scheme is almost identical to L-BFGS(4) and Fletcher–Reeves CG. The Anderson-accelerated basic scheme was pioneered by Shantraj et al. [38] and Chen et al. [88]. It may be interpreted as a Quasi-Newton method of multi-secant type [89]. Applied to linear problems it is closely related to GMRES, see Toth–Kelley [90]. For low accuracy (above 10^{-3}), Anderson with depth 4 outperforms all other methods. This is to be expected, as the (generalized) minimal residual method represents the Krylov subspace solver with minimal residual [91]. (Recall that linear CG is the Krylov subspace solver with minimal “energy”.) However, for higher accuracy, the method slows down considerably. This complies with theoretical and numerical results [90], which show that an Anderson-accelerated gradient method does not improve the convergence rate, only the constant, see Li–Li [92].

Table 1 Material parameters used in this article, supplemented by their source

Material	E in GPa	ν	Other parameters	
Quartz sand grains [79,80]	66.9	0.25		
Quartz glass binder [81]	71.7	0.17		
E-glass fibers [82]	72	0.22		
Polymer matrix [82]	3.0	0.35	$\sigma_Y = 20$ MPa $k_2 = 15$ MPa	$k_1 = 1$ MPa $m = 150$
Ceramic particles [83]	400	0.2		
Aluminium matrix [83]	75	0.3	$\sigma_Y = 75$ MPa $m = 0.3895$	$k = 416$ MPa

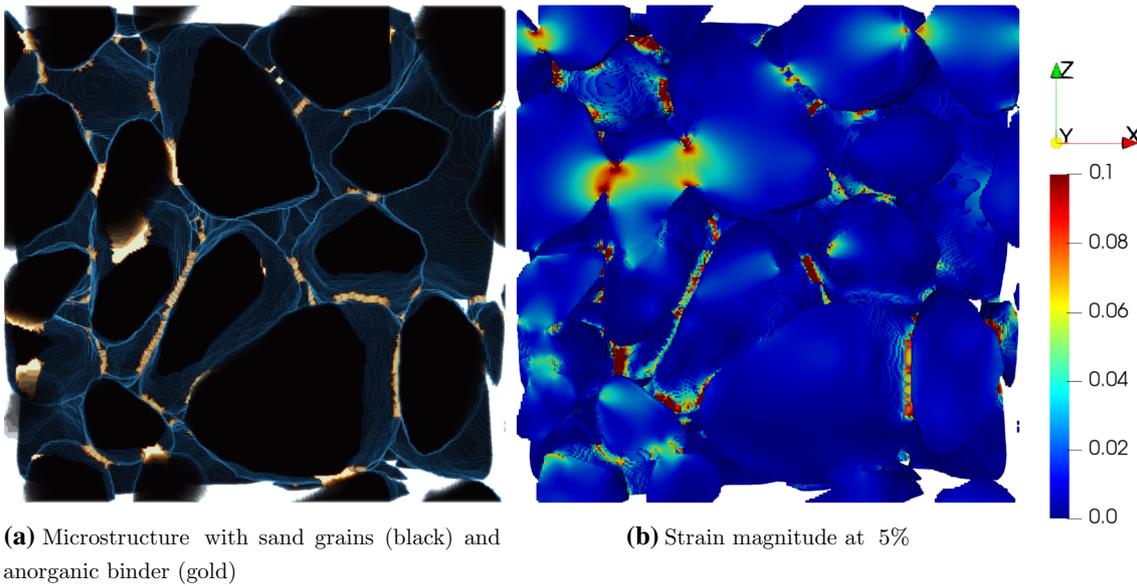


Fig. 1 Sandcore microstructure and local solution field

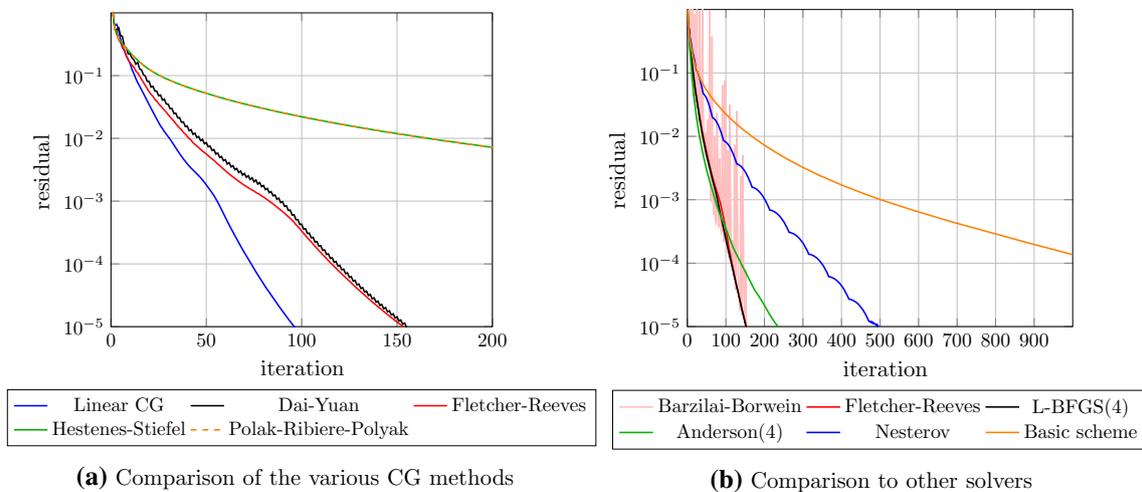


Fig. 2 Comparison of different solvers for the sandcore microstructure, cf Fig. 1

For comparison we have also included Nesterov's method with speed restart, as described in Schneider [74]. It represents the fastest accelerated gradient method for this kind of problem, compare Schneider [39], Sect. 3.2. Nesterov's method is optimal in the sense that its convergence rate is identical to the worst-case convergence rate for optimizing uniformly convex functions with Lipschitz gradient [70]. However, the constant in front of the convergence estimate is about 4 times as high as the worst-case bound (which might also be too pessimistic). Furthermore, speed restarting is always slower than if Nesterov's method is applied for explicitly known convexity constant. For the problem at hand, Nesterov's method requires about 5 times as many iterations as linear CG. Last but not least, we report on the basic scheme which fails to converge to the prescribed tolerance within 1000 iterations.

4.3 A planar short-fiber reinforced composite

We investigate a short-fiber reinforced microstructure and the associated effective mechanical response. We consider short glass fibers with a length of $275\ \mu\text{m}$ and a diameter of $10\ \mu\text{m}$, reinforcing a polymer by 18% volume fraction. The second-order fiber-orientation tensor reads $A = \text{diag}(0.45, 0.45, 0.1)$. The periodic microstructure, cf Fig. 3a, has a size of $1024\ \mu\text{m} \times 1024\ \mu\text{m} \times 128\ \mu\text{m}$ and was generated by the sequential addition and migration algorithm [93]. The resulting microstructure contains 1119 cylindrical fibers and was discretized by $512 \times 512 \times 64$ voxels.

For the elastic parameters of E-glass and the polymer, cf Table 1, we solve a linear homogenization problem for prescribed uniaxial extension of 5% strain up to a specified tolerance of 10^{-10} . Compared to the example of the previous section, the example at hand is simpler as the constituents are finitely contrasted. (The difficulties emerge for the nonlinear behavior, though). We compare the performance of the various nonlinear CG methods to linear CG in Fig. 4a. Both the Hestenes–Stiefel and the Polak–Ribière–Polyak method were not competitive, whence they are not included in the figure. The linear conjugate gradient method converges linearly, and quickly so. The Dai–Yuan and the Fletcher–Reeves CG are almost indistinguishable, converging also linearly, but with a slower rate than linear CG. Both require about 50% more iterations than linear CG. In contrast to the sandcore microstructure, the Dai–Yuan CG does not exhibit a zig-zagging convergence behavior. This observation may indicate that the Dai–Yuan method is more sensitive to the step-size than the Fletcher–Reeves CG. Indeed, due to the finite contrast, the step-size of the basic scheme (which we also use for the nonlinear CGs) is smaller than for the infinite-contrast case.

The numerical performance of a variety of other solvers is shown in Fig. 4b. We see that the Fletcher–Reeves method

lags behind three other methods in terms of convergence rate: the L-BFGS(4) method, the heavy-ball scheme and the Barzilai–Borwein solver all lead to approximately identical convergence rate. The heavy-ball scheme, however, starts to become unstable for a residual of 10^{-9} . This is certainly a problem of the implementation, see Ernesti–Schneider–Böhlke [37], and can be cured easily, but at the expense of storing an additional field. The fluctuations inherent to the Barzilai–Borwein method are smaller than for the infinite-contrast case of the previous section. Similar to the sandcore microstructure, the Anderson-accelerated basic scheme is superior to all other schemes up to a tolerance of 10^{-3} , but become dramatically slower for higher accuracy. The basic scheme is not competitive.

After this warm-up, we turn our attention to the inelastic behavior. More precisely, following Wu et al. [82], we furnish the matrix with a J_2 -elastoplastic material model and linear-exponential hardening

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

We apply uniaxial extension in x -direction up to 5% strain, distributed into 50 equal-sized time steps. The stress-strain curves for matrix, fiber and the composite are shown in Fig. 5a. The local strain fields corresponding to 1%, 2%, 3%, 4% and 5% are shown in Fig. 3. Up to 1% applied strain, cf Fig. 3b, the local strain field exhibits peaks only at fiber tips and at spots where fibers are closely packed. For higher levels of strain, large portions of the matrix get plastified, as is reflected in the high strain levels. For the inelastic problem at hand, we compare the basic scheme, Fletcher–Reeves CG, the Newton-CG method, L-BFGS(4) and the Barzilai–Borwein scheme, both in terms of iteration count and runtime, cf Fig. 6. Please keep in mind that the runtimes were recorded for the computer with 6 cores, see Sect. 4.1. For the forcing term in Newton-CG, we choose forcing-term choice 2 with $\gamma_0 = 0.75$, see Wicht-Schneider-Böhlke [35].

Taking a look at the iteration count, cf Fig. 6a, we see that the iterations required for the basic scheme increase monotonically up to about 2% applied strain. For higher levels of loading, the iteration count decreases again. For the other solvers, the iterations counts are much smaller than for the basic scheme. Their respective iteration counts are on a similar level, with the Newton-CG lagging behind slightly for strain levels exceeding 3.5%.

A look at the runtime per step, cf Fig. 6b, enables us to further examine the performance of the solvers. As for the iteration counts, the basic scheme leads to much higher runtimes. The other solvers however, operate on different levels. The limited-memory BFGS method clearly lags behind in terms of performance. Fletcher–Reeves CG and Newton-CG exhibit more or less the same runtimes, with Newton-CG having advantages for lower strain-levels and Fletcher–Reeves

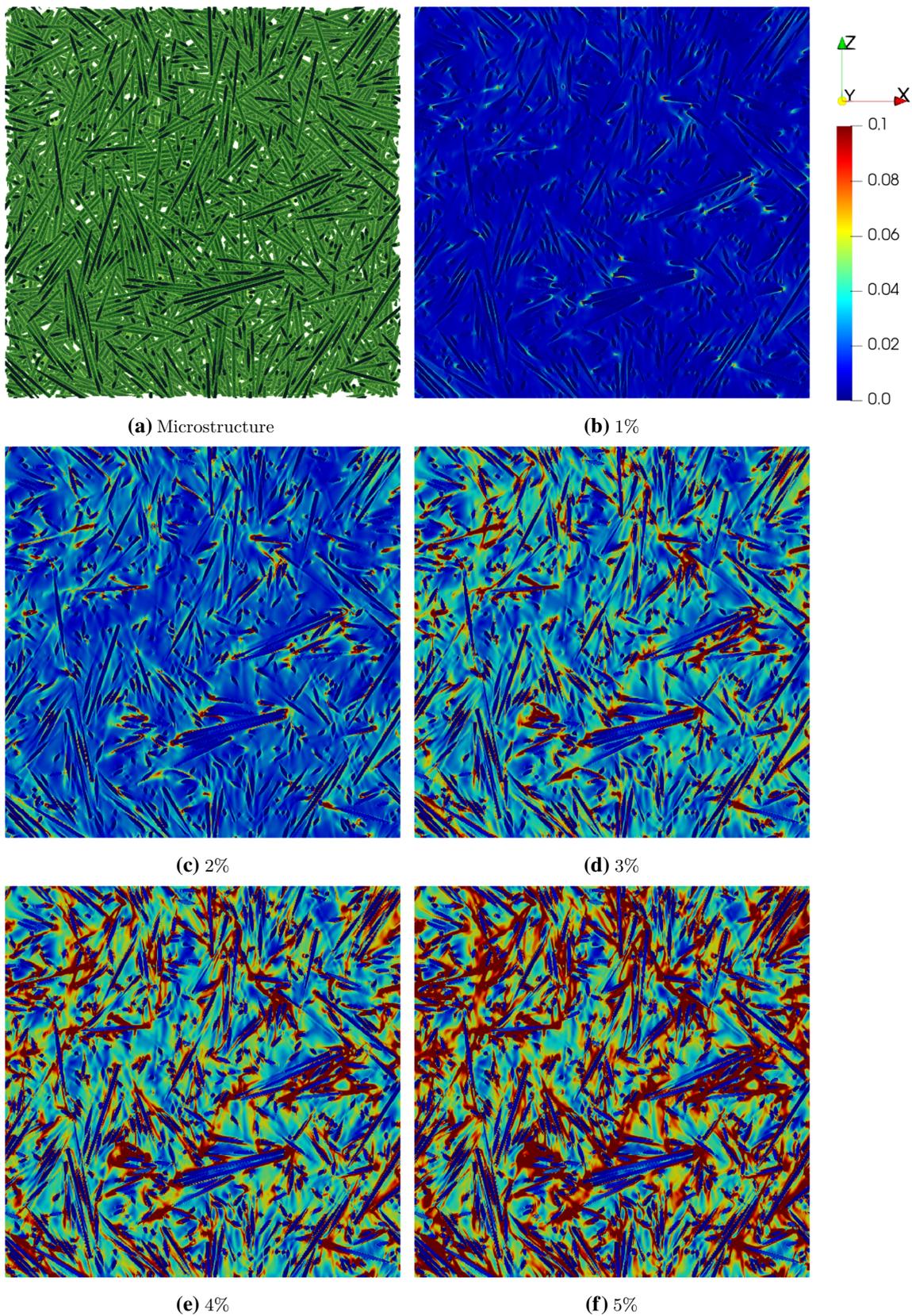


Fig. 3 Fiber-reinforced microstructure and norm of the strain field upon different levels of monotonic uniaxial loading

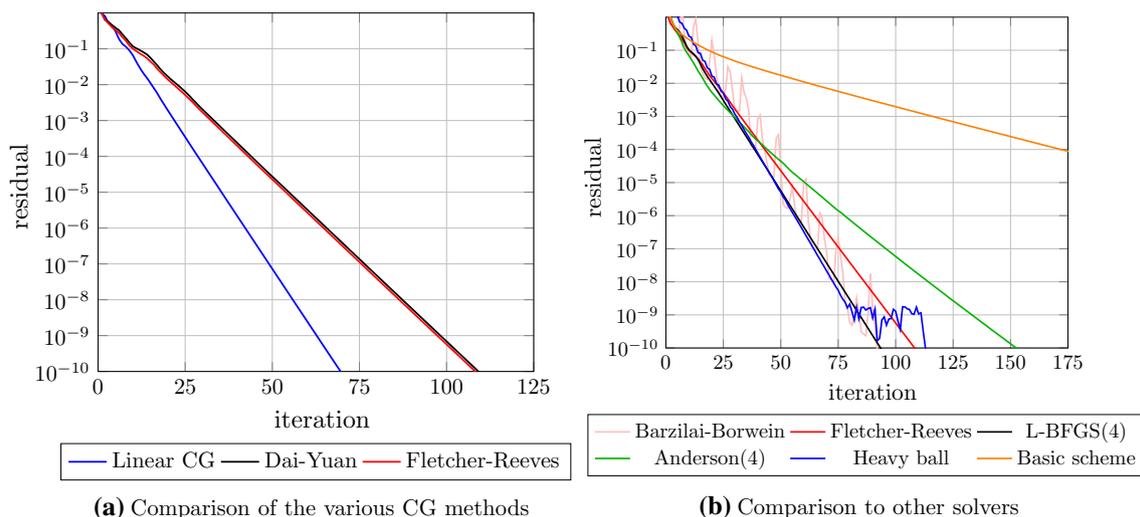


Fig. 4 Convergence behavior of different solvers for homogenizing the fiber-reinforced microstructure, Fig. 3a

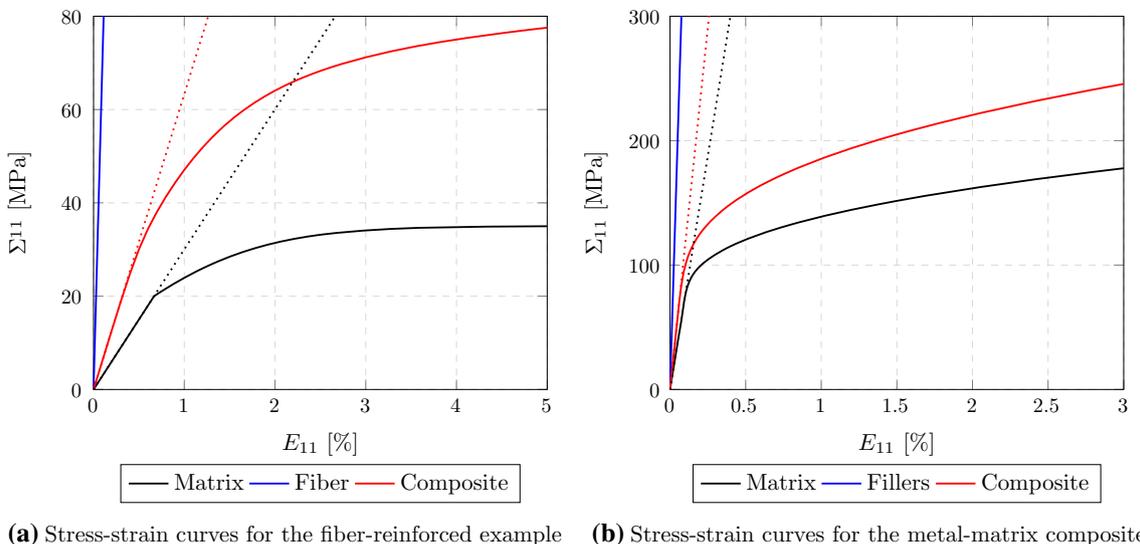


Fig. 5 Stress–strain diagrams for the inelastic examples in this paper. The elastic behavior is indicated by dots

CG being faster later on. The Barzilai–Borwein scheme appears to be faster, probably because of the lower computational cost per step. Indeed, the Barzilai–Borwein scheme is very close to the basic scheme in terms of computational cost per step, whereas both Newton-CG and Fletcher–Reeves CG are similar in terms of these costs, as long as the non-linear material law is comparably expensive as applying the material tangent.

Note that we have also run the heavy-ball method for this problem. However, the scheme did not converge within 1000 iterations for the fourth load step, whence we consider the method clearly inferior.

To quantify our observations, both the average iteration count and the average runtime per step are collected in Table 2. The basic scheme takes more than ten times the

iteration count that is required for the second worst solver considered. Thus, for the problem at hand, the basic scheme is clearly not competitive. Notice, however, that the basic scheme features the lowest run-time per iteration.

In terms of runtime per iteration, the Barzilai–Borwein method and Newton-CG are comparable. The Barzilai–Borwein method requires fewer iterations than Newton-CG. This is caused by the delicate choice of the forcing term (the tolerance to which the linear system needs to be solved) in the Newton method, see Wicht-Schneider-Böhlke [35] for a discussion. The Fletcher–Reeves CG has a similar iteration count as the Barzilai–Borwein method, but features a higher runtime per iteration. All in all, it is 20% slower than the Barzilai–Borwein method. Still, it is slightly faster than Newton-CG, but is characterized by a much smaller memory-

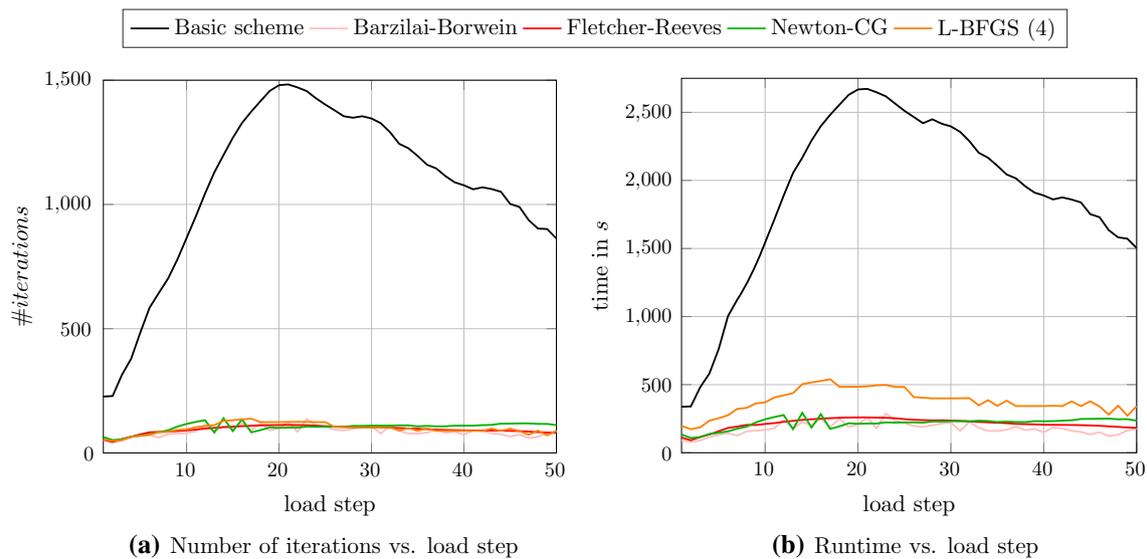


Fig. 6 Iteration count and runtime per load step for the fiber-reinforced composite with exponential-linear hardening

Table 2 Average iteration count and total runtime for 50 load steps and the inelastic fiber-reinforced problem

	Mean iterations	Runtime in s	Runtime per iteration in s	Performance overhead
Basic scheme	1071.70	94975.57	88.62	10.67
Barzilai–Borwein	86.50	8901.37	102.91	1.00
Fletcher–Reeves CG	93.96	10697.62	113.85	1.20
L-BFGS (4)	98.08	18969.10	193.40	2.13
Newton-CG	103.36*	10895.86	105.40	1.22

Notice that for the Newton-CG method we sum linear and nonlinear iterations, and mark it by a *. The performance overhead is defined as the runtime divided by the smallest runtime

footprint than Newton-CG. Last but not least, we mention L-BFGS, which has comparable iteration count, but is not competitive in terms of runtime. Indeed, the runtime per iteration is more than twice as high as for the basic scheme.

4.4 A metal-matrix composite

As our last inelastic example, we consider a metal-matrix composite (MMC). The microstructure containing spherical fillers is shown in Fig. 7a. The ceramic fillers are modeled elastically, whereas the aluminium matrix is governed by J_2 -elastoplasticity with power-law hardening

$$\sigma_0(p) = \sigma_Y + k p^m.$$

The material parameters are listed in Table 1 following Segurado et al [83]. The composite was subjected to uniaxial extension up to 3% strain, decomposed into 225 equidistant time-steps. The stress-strain curves are shown in Fig. 5b. The accumulated plastic strain fields for increasing levels of applied strain are shown in Fig. 7b–d. Due to the strong nonlinearity induced by power-law hardening,

this setup represents a challenging benchmark for FFT-based solution methods, and has been considered before in Schneider–Wicht–Böhlke [29] and Schneider [39]. To ensure compatibility to the results in the mentioned papers, the spectral discretization of Moulinec–Suquet [1,2] was used.

The iteration count per load step for the basic scheme, the Barzilai–Borwein method and Fletcher–Reeves CG is shown on the left of Fig. 8. We see that the basic scheme performs quite well for this example, and the Fletcher–Reeves CG is on the same level as Barzilai–Borwein.

We refrained from including too many solvers into the latter diagram, as many solvers operate on a similar level for this example (or fail admirably, rendering the diagram less expressive). Thus, we included a table documenting the mean iteration counts for various solvers and increasing resolution. The solvers are grouped into

1. Gradient methods (basic scheme, Barzilai–Borwein),
2. Polarization schemes (Eyre–Milton scheme [25], the augmented Lagrangian method of Michel–Moulinec–Suquet [26] and the Monchiet–Bonnet solver [27]),

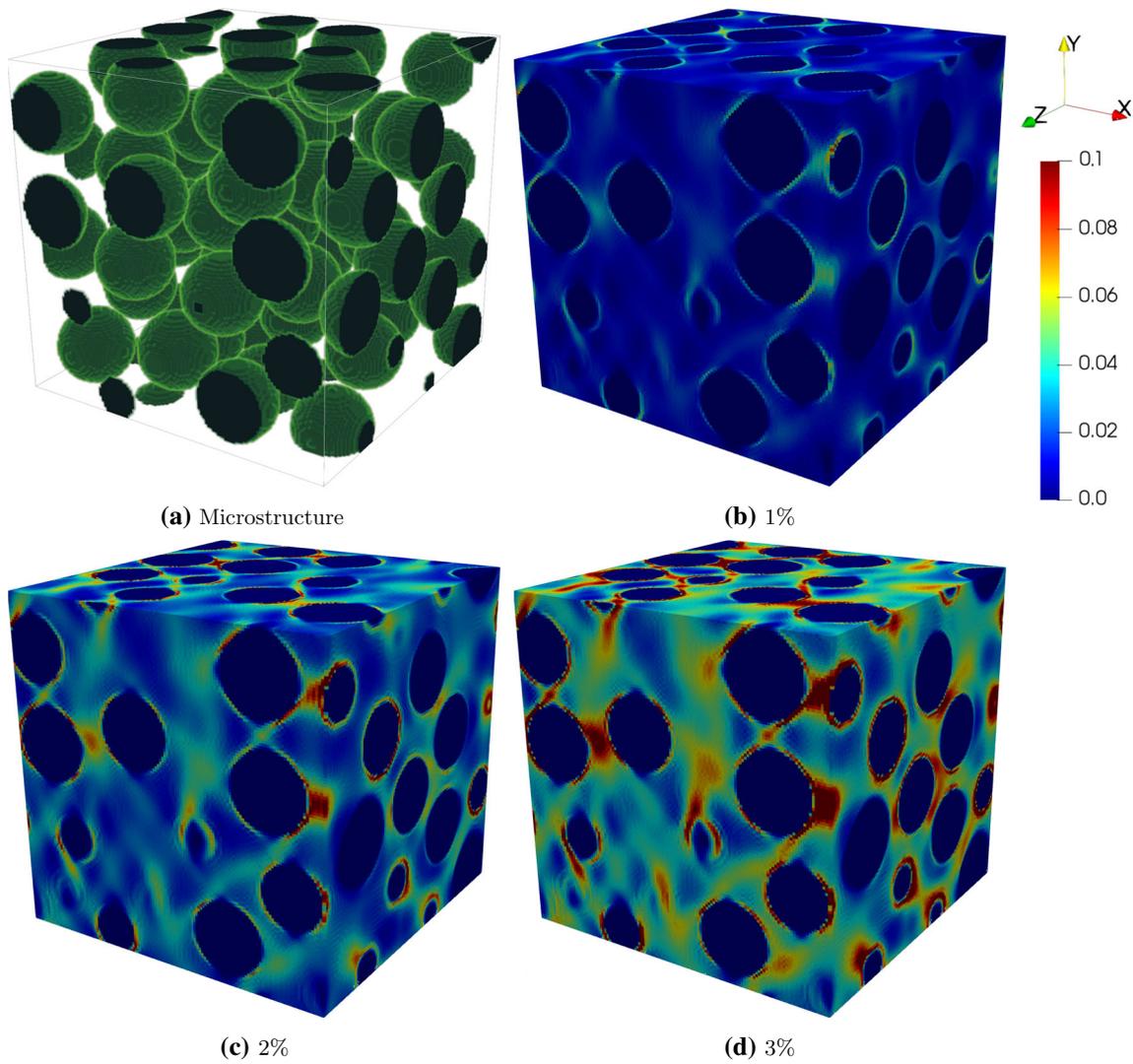
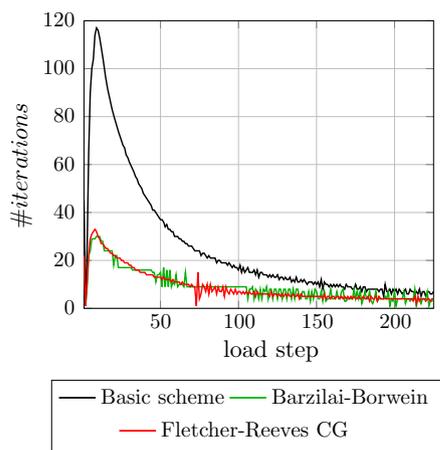


Fig. 7 Microstructure and accumulated plastic strain for increasing applied axial strain for the metal-matrix composite (MMC)



	16 ³	32 ³	64 ³	128 ³
Basic Scheme	24.62	23.41	24.8	25.98
Barzilai-Borwein	9.95	9.42	9.40	9.36
Eyre-Milton	7.69	7.4	7.31	8.74
Monchiet-Bonnet	7.0	7.08	6.51	6.83
Augmented Lagrangian	10.94	10.37	9.47	9.84
Nesterov	24.8	19.62	16.4	17.16
Heavy-ball method	29.15	35.72	33.79	34.73
Fletcher-Reeves CG	9.91	9.24	9.11	9.05

Fig. 8 Iteration per load step (128³ voxels, left) and average iteration count for varying resolution (right)

Table 3 Summary of FFT-based solvers for nonlinear and inelastic computational homogenization

Solution scheme	Memory footprint (strain-like fields)	Summary and remarks
Basic scheme [1,2]	1	<ul style="list-style-type: none"> • Gradient descent method [34] • Lowest memory requirements • Robust, but slow
Barzilai–Borwein [39,86]	2	<ul style="list-style-type: none"> • Gradient descent with adaptive step-size • Non-monotonic convergence behavior • Fastest choice for inexpensive material laws
Nesterov’s and heavy-ball method [37,74]	2	<ul style="list-style-type: none"> • Fast gradient methods • Parameters depending of convexity constants • Competitive if the parameters are known • Restarting strategies not competitive
Our FFT-based Fletcher–Reeves CG (2.9)	3	<ul style="list-style-type: none"> • Fast gradient method with adaptive momentum • Monotonic decrease of the residual • Devoid of tedious parameter selection
Anderson-accelerated basic scheme [38,94]	$2m + 2$	<ul style="list-style-type: none"> • Multi-secant Quasi-Newton method [89] • Accelerates convergent fixed-point iterations • Optimal depth m between 2 and 5 robust, but not exceedingly fast
Newton-CG [34,95]	8.5	<ul style="list-style-type: none"> • Inexact Newton method • Requires computing the material tangent • Fastest choice for expensive material laws

3. Fast gradient methods (Nesterov’s method with optimal step-size [49], the heavy-ball method with optimal step-size [49], Fletcher–Reeves CG (2.9)).

We notice that all schemes are stable w.r.t. mesh refinement except for Nesterov’s method and the heavy-ball scheme. This is a result of the lack of uniform convexity of the condensed energy functional governing von-Mises elastoplasticity with power-law hardening. Recall that the performances of Nesterov’s method and the heavy-ball scheme crucially rely upon knowing the strong convexity constant in question. We follow the approach of determining the strong convexity constant in terms of the eigenvalues of the tangent stiffness, for every iteration. As the mesh gets refined, the strains at the fiber tips increase, leading to a lowering of the convexity constant. A similar, but less severe trend is observed for the Eyre-Milton scheme, where the average iteration count increases for 128^3 voxels. For the polarization schemes, the strong convexity constant is required as well. However, the two damped polarization solvers, the augmented Lagrangian and the Monchiet-Bonnet scheme suffer less from mesh

Concerning the overall performance, the basic scheme is worst, requiring about 150% more iterations than the Barzilai–Borwein scheme. In turn, the polarization schemes outperform the Barzilai–Borwein scheme, or operate on roughly the same level. Recall from Schneider [39] that polar-

ization schemes fail to perform well for infinitely contrasted media, but exhibit very good performance for finitely contrasted media, as long as the complexity-reduction trick [29] is available.

From the category of fast gradient schemes, both Nesterov’s scheme and the heavy-ball solver fail to be competitive. The heavy-ball method becomes unstable, probably rooted in the lack of global convergence of the scheme. Nesterov’s method is globally convergent (and faster than the basic scheme). However, it requires about twice as many iterations as the other competitive solvers. Fletcher–Reeves CG consistently outperforms the Barzilai–Borwein scheme in terms of required iterations, but lags behind the two better polarization schemes.

5 Conclusion

This work was devoted to studying nonlinear conjugate gradient methods applied in the context of FFT-based micromechanics. We showed that the Fletcher–Reeves nonlinear conjugate gradient method is capable of filling the role as general-purpose solver for FFT-based computational micromechanics. Its monotonicity properties, which we showed by computational examples, set it apart from the Barzilai–Borwein method. In other regards, the performance

is similar. Of course, the price to pay for monotonicity is a slight increase in both memory footprint and computational effort. In this regard, we may insert the Fletcher–Reeves CG into Table 3, which compares the corresponding merits and downsides of the competitive solvers for FFT-based computational micromechanics.

Our focus has been on providing a useful and useable method, and did not dwell upon theoretical investigations. Exploring the Fletcher–Reeves dynamical system (2.23) appears alluding, in particular in terms of stability and basins of attraction. Also, the corresponding time-discrete system should be investigated.

Acknowledgements Open Access funding provided by Projekt DEAL. This work was supported by the German Research Foundation (DFG) within the International Research Training Group “Integrated engineering of continuous-discontinuous long fiber reinforced polymer structures” (GRK 2078). The author acknowledges stimulating discussions with H. Andrä, D. Wicht and S. Gajek.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Moulinec H, Suquet P (1994) A fast numerical method for computing the linear and nonlinear mechanical properties of composites. *C R Acad Sci Sér II* 318(11):1417–1423
- Moulinec H, Suquet P (1998) A numerical method for computing the overall response of nonlinear composites with complex microstructure. *Comput Methods Appl Mech Eng* 157:69–94
- Frigo M, Johnson SG (2005) The design and implementation of FFTW3. *Proc IEEE* 93:216–231
- Matouš K, Geers MGD, Kouznetsova VG, Gillman A (2017) A review of predictive nonlinear theories for multiscale modeling of heterogeneous materials. *J Comput Phys* 330:192–220
- Spahn J, Andrä H, Kabel M, Müller R (2014) A multiscale approach for modeling progressive damage of composite materials using fast Fourier transforms. *Comput Methods Appl Mech Eng* 268:871–883
- Michel JC, Suquet P (2003) Nonuniform transformation field analysis. *Int J Solids Struct* 40:6937–6955
- Fritzen F, Leuschner M (2013) Reduced basis hybrid computational homogenization based on a mixed incremental formulation. *Comput Methods Appl Mech Eng* 260:143–154
- Köbler J, Schneider M, Ospald F, Andrä H, Müller R (2018) Fiber orientation interpolation for the multiscale analysis of short fiber reinforced composite parts. *Comput Mech* 61(6):729–750
- Liu Z, Wu CT, Koishi M (2019) A deep material network for multiscale topology learning and accelerated nonlinear modeling of heterogeneous materials. *Comput Methods Appl Mech Eng* 345:1138–1168
- Liu Z, Wu C (2019) Exploring the 3D architectures of deep material network in data-driven multiscale mechanics. *J Mech Phys Solids* 127:20–46
- Lahelec N, Michel JC, Moulinec H, Suquet P (2003) Analysis of inhomogeneous materials at large strains using fast Fourier transforms. In: IUTAM symposium on comp. mech. of solid materials at large strains (C. Miehe, ed.), vol. 108 of solid mechanics and its applications, pp 247–258. Springer, Netherlands
- Eisenlohr P, Diehl M, Lebensohn RA, Roters F (2013) A spectral method solution to crystal elasto-viscoplasticity at finite strains. *Int J Plast* 46:37–53
- Biner SB, Hu SY (2009) Simulation of damage evolution in composites: a phase-field model. *Acta Mater* 57:2088–2097
- Sharma L, Peerlings RHJ, Shanthraj P, Roters F, Geers MGD (2020) An FFT-based spectral solver for interface decohesion modelling using a gradient damage approach. *Comput Mech* 65:925–939
- Lucarini S, Segurado J (2020) An upscaling approach for micromechanics based fatigue: from RVEs to specimens and component life prediction. *Int J Fract* 223:93–108
- Schneider M (2020) An FFT-based method for computing weighted minimal surfaces in microstructures with applications to the computational homogenization of brittle fracture. *Int J Numer Methods Eng* 121:1367–1387
- Vondřejc J, Zeman J, Marek I (2015) Guaranteed upper–lower bounds on homogenized properties by FFT-based Galerkin method. *Comput Methods Appl Mech Eng* 297:258–291
- Brisard S, Dormieux L (2010) FFT-based methods for the mechanics of composites: a general variational framework. *Comput Mater Sci* 49(3):663–671
- Brisard S, Dormieux L (2012) 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* 217–220:197–212
- Schneider M, Merkert D, Kabel M (2017) FFT-based homogenization for microstructures discretized by linear hexahedral elements. *Int J Numer Methods Eng* 109(10):1461–1489
- Schneider M, Ospald F, Kabel M (2016) Computational homogenization of elasticity on a staggered grid. *Int J Numer Methods Eng* 105(9):693–720
- Willot F (2015) Fourier-based schemes for computing the mechanical response of composites with accurate local fields. *C R Méc* 343(3):232–245
- Dorn C, Schneider M (2019) Lippmann–Schwinger solvers for the explicit jump discretization for thermal computational homogenization problems. *Int J Numer Methods Eng* 118(11):631–653
- Tu F, Jiao Y, Zhou X, Cheng Y, Tan F (2019) The implementation of B-splines to Hashin and Shtrikman variational principle based FFT method for the homogenization of composite. *Int J Solids Struct* 191–192:133–145
- Eyre DJ, Milton GW (1999) A fast numerical scheme for computing the response of composites using grid refinement. *Eur Phys J Appl Phys* 6(1):41–47
- Michel JC, Moulinec H, Suquet P (2001) A computational scheme for linear and non-linear composites with arbitrary phase contrast. *Int J Numer Methods Eng* 52:139–160
- Monchiet V, Bonnet G (2012) A polarization-based FFT iterative scheme for computing the effective properties of elastic composites with arbitrary contrast. *Int J Numer Methods Eng* 89:1419–1436
- Moulinec H, Silva F (2014) Comparison of three accelerated FFT-based schemes for computing the mechanical response of composite materials. *Int J Numer Methods Eng* 97:960–985
- Schneider M, Wicht D, Böhlke T (2019) On polarization-based schemes for the FFT-based computational homogenization of inelastic materials. *Comput Mech* 64(4):1073–1095

30. Douglas J, Rachford HH (1956) On the numerical solution of heat conduction problems in two and three space variables. *Trans Am Math Soc* 82:421–439
31. Zeman J, Vondřejc J, Novák J, Marek I (2010) Accelerating a FFT-based solver for numerical homogenization of periodic media by conjugate gradients. *J Comput Phys* 229(21):8065–8071
32. Paige CC, Saunders MA (1975) Solution of sparse indefinite systems of linear equations. *SIAM J Numer Anal* 12(4):617–629
33. Gélébart L, Mondon-Cancel R (2013) Non-linear extension of FFT-based methods accelerated by conjugate gradients to evaluate the mechanical behavior of composite materials. *Comput Mater Sci* 77:430–439
34. Kabel M, Böhlke T, Schneider M (2014) Efficient fixed point and Newton–Krylov solvers for FFT-based homogenization of elasticity at large deformations. *Comput Mech* 54(6):1497–1514
35. Wicht D, Schneider M, Böhlke T (2020) On Quasi-Newton methods in FFT-based micromechanics. *Int J Numer Methods Eng* 121:1665–1694
36. Schneider M (2017) An FFT-based fast gradient method for elastic and inelastic unit cell homogenization problems. *Comput Methods Appl Mech Eng* 315:846–866
37. Ernesti F, Schneider M, Böhlke T (2020) Fast implicit solvers for phase-field fracture problems on heterogeneous microstructures. *Comput Methods Appl Mech Eng* 363:112793
38. Shantraj P, Eisenlohr P, Diehl M, Roters F (2015) Numerically robust spectral methods for crystal plasticity simulations of heterogeneous materials. *Int J Plast* 66:31–45
39. Schneider M (2019) On the Barzilai–Borwein basic scheme in FFT-based computational homogenization. *Int J Numer Methods Eng* 118:482–494
40. Fletcher R, Reeves C (1964) Function minimization by conjugate gradients. *Comput J* 7:149–154
41. Hestenes M, Stiefel E (1952) Methods of conjugate gradients for solving linear systems. *J Res Nat Bur Standards* 49:409–436
42. Nocedal J, Wright SJ (1999) *Numerical optimization*. Springer, Berlin
43. Polak E, Ribière G (1969) Note sur la convergence de directions conjuguées. *Rev Fr Inf Rech Opert* 3e Ann 16:35–43
44. Polyak BT (1969) The conjugate gradient method in extremum problems. *USSR Comput Math Math Phys* 9:94–112
45. Dai YH, Yuan Y (1999) A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J Optim* 10(1):177–182
46. Polyak BT (1964) Some methods of speeding up the convergence of iteration methods. *USSR Comput Math Math Phys* 4(5):1–17
47. Polyak BT (1987) *Introduction to optimization*. Optimization Software, Inc, New York
48. Ghadimi E, Feyzmahdavian HR, Johansson M (2015) Global convergence of the heavy-ball method for convex optimization. In: 2015 European control conference (ECC), pp 310–315
49. Lessard L, Recht B, Packard A (2016) Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM J Optim* 26(1):57–95
50. Sun J, Zhang J (2001) Global convergence of conjugate gradient methods without line search. *Ann Oper Res* 103:161–173
51. Powell MJD (2009) Nonconvex minimization calculations and the conjugate gradient method. In: Griffiths DF (ed) *Lecture notes in mathematics*, vol 1066. Springer, Berlin, pp 122–141
52. Gilbert J, Nocedal J (1992) Global convergence properties of conjugate gradient methods for optimization. *SIAM J Optim* 2(1):21–42
53. Dai YH, Yuan Y (1998) Some properties of a new conjugate gradient method. In: *Advances in nonlinear programming: proceedings of the 96 international conference on nonlinear programming*. Springer, Boston, MA, USA, pp 251–262
54. Hager WW, Zhang H (2006) A survey of nonlinear conjugate gradient methods. *Pac J Optim* 2:35–58
55. Dai Y, Kou C (2016) A Barzilai–Borwein conjugate gradient method. *Sci China Math* 59(8):1511–1524
56. Broyden CG (1970) The convergence of a class of double rank minimization algorithms: 2. The new algorithm. *J Math Anal Appl* 6:222–231
57. Fletcher R (1970) A new approach to variable metric algorithms. *Comput J* 13:317–322
58. Goldfarb D (1970) A family of variable metric methods derived by variational means. *Math Comput* 24:23–26
59. Shanno DF (1970) Conditioning of quasi-Newton methods for function minimization. *Math Comput* 24:647–650
60. Nocedal J (1980) Updating quasi-Newton matrices with limited storage. *Math Comput* 35(151):773–782
61. Liu H, Liu Z (2019) An efficient Barzilai–Borwein conjugate gradient method for unconstrained optimization. *J Optim Theory Appl* 180:879–906
62. Helmke U, Moore J (1996) *Optimization and dynamical systems*. Springer, Berlin
63. Zavriev S, Kostyuk F (1993) Heavy-ball method in nonconvex optimization problems. *Comput Math Model* 4(4):336–341
64. Sun T, Yin P, Li D, Huang C, Guan L, Jiang H (2019) Non-ergodic convergence analysis of heavy-ball algorithms. In: *The thirty-third AAAI conference on artificial intelligence (AAAI-19)*, pp 5033–5040
65. Siegel JW (2019) Accelerated first-order methods: differential equations and Lyapunov functions, pp 1–17. [arXiv:1903.05671](https://arxiv.org/abs/1903.05671)
66. Dai YH (2011) Convergence analysis of nonlinear conjugate gradient methods. In: Wang Y, Yang C, Yagola AG (eds) *Optimization and regularization for computational inverse problems and applications*. Springer, Berlin, pp 1157–171
67. Nesterov Y (1983) A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Dokl Akad Nauk SSSR* 269(3):543–547
68. Nesterov Y (2005) Smooth minimization of non-smooth functions. *Math Program* 103:127–152
69. Su W, Boyd S, Candes E (2014) A differential equation for modeling Nesterov’s accelerated gradient method: theory and insights. In: Ghahramani Z, Welling M, Cortes C, Lawrence N, Weinberger K (eds) *Advances in neural information processing systems 27*. Curran Associates Inc, New York, pp 2510–2518
70. Nesterov Y (2004) *Introductory lectures on convex optimization: a basic course*. Kluwer Academic Publishers, Springer
71. Bhaya A, Kaszkurewicz E (2003) Iterative methods as dynamical systems with feedback control. In: 42nd IEEE international conference on decision and control, vol 3. IEEE, pp 2374–2380
72. Ortiz L, Stainier L (1999) The variational formulation of viscoplastic constitutive update. *Comput Methods Appl Mech Eng* 171:419–444
73. Miehe C (2002) Strain-driven homogenization of inelastic microstructures and composites based on an incremental variational formulation. *J Numer Methods Eng* 55:1285–1322
74. Schneider M (2017) An FFT-based fast gradient method for elastic and inelastic unit cell homogenization problems. *Comput Methods Appl Mech Eng* 315:846–866
75. Milton GW (1990) On characterizing the set of possible effective tensors of composites: the variational method and the translation method. *Commun Pure Appl Math* 43:63–125
76. Grimm-Strehle H, Kabel M (2019) Runtime optimization of a memory efficient CG solver for FFT-based homogenization: implementation details and scaling results for linear elasticity. *Comput Mech* 64(5):1339–1345
77. Kabel M, Fliegner S, Schneider M (2016) Mixed boundary conditions for FFT-based homogenization at finite strains. *Comput Mech* 57(2):193–210

78. Schneider M (2015) Convergence of FFT-based homogenization for strongly heterogeneous media. *Math Methods Appl Sci* 38(13):2761–2778
79. Daphalapurkar NP, Wang F, Fu B, Lu H, Komanduri R (2011) Determination of mechanical properties of sand grains by nanoindentation. *Exp Mech* 51:719–728
80. Wichtmann T, Triantafyllidis T (2010) On the influence of the grain size distribution curve on P-wave velocity, constrained elastic modulus M_{max} and Poisson's ratio of quartz sands. *Soil Dyn Earthq Eng* 30(8):757–766
81. Sanditov DS, Mantatov VV, Sanditov BD (2009) Poisson ratio and plasticity of glasses. *Tech Phys* 54(4):594–596
82. Wu L, Adam L, Doghri I, Noels L (2017) An incremental-secant mean-field homogenization method with second statistical moments for elasto-visco-plastic composite materials. *Mech Mater* 114:180–200
83. Segurado J, Llorca J, González C (2002) On the accuracy of mean-field approaches to simulate the plastic deformation of composites. *Scr Mater* 46(7):525–529
84. Schneider M, Hofmann T, Andrä H, Lechner P, Etemeyer F, Volk W, Steeb H (2018) Modelling the microstructure and computing effective elastic properties of sand core materials. *Int J Solid Struct* 143:1–17
85. Etemeyer F, Lechner P, Hofmann T, Andrä H, Schneider M, Grund D, Volk W, Günther D (2020) Digital sand core physics: predicting physical properties of sand cores by simulations on digital microstructures. *Int J Solids Struct* 188–189:155–168
86. Barzilai J, Borwein JM (1988) Two-point step size gradient methods. *IMA J Numer Anal* 8:141–148
87. Dai YH, Liao LZ (2002) R-linear convergence of the Barzilai and Borwein gradient method. *IMA J Numer Anal* 22:1–10
88. Chen Y, Gélébart L, Chateau C, Bornert M, Sauder C, King A (2019) 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* 161:111–126
89. Fang H-R, Saad Y (2009) Two classes of multisection methods for nonlinear acceleration. *Numer Linear Algebra Appl* 16:197–221
90. Toth A, Kelley C (2015) Convergence analysis for Anderson acceleration. *SIAM J Numer Anal* 53(2):805–819
91. Saad Y, Schultz MH (1986) GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J Sci Stat Comput* 7:856–869
92. Li Z, Li J (2019) A fast Anderson–Chebyshev acceleration for nonlinear optimization, pp 1–17. [arXiv:1809.02341](https://arxiv.org/abs/1809.02341)
93. Schneider M (2017) The sequential addition and migration method to generate representative volume elements for the homogenization of short fiber reinforced plastics. *Comput Mech* 59:247–263
94. Anderson DG (1965) Iterative procedures for nonlinear integral equations. *J ACM* 12(4):547–560
95. Dembo RS, Eisenstat SC, Steihaug T (1982) Inexact Newton methods. *SIAM J Numer Anal* 19:400–408

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.