Computational Mechanics (2021) 68:211–227
https://doi.org/10.1007/s00466-021-02027-z

ORIGINAL PAPER

An engineering interpretation of Nesterov's convex minimization algorithm and time integration: application to optimal fiber orientation

P. Areias[1,2] · T. Rabczuk[3]

Received: 15 January 2021 / Accepted: 29 April 2021 / Published online: 19 May 2021

Abstract
Nesterov's 1983 first-order minimization algorithm is equivalent to the numerical solution of a second-order ODE with non-constant damping. It is known that the algorithm can be obtained by time discretization with asynchronous damping, a long-standing technique in computational explicit dynamics. We extend the solution of the ODE to other time discretization algorithms, analyze their properties and provide an engineering interpretation of the process as well as a prototype implementation, addressing the estimation of the relevant parameters in the computational mechanics context. The main result is that a standard Newmark-type time-integration finite element code can be adopted to perform classical optimization in mechanics. Standard FE analysis, sensitivity analysis and optimization are performed in sequence using a typical FE framework. Geometric nonlinearities in the conservative case are addressed by the adjoint-variable method and examples of optimal fiber orientation are shown, exhibiting remarkable advantages with respect to more traditional optimization algorithms.

Keywords First-order optimization · Nesterov's algorithm · Time-integration · Optimal fiber orientation · Geometrically nonlinear

1 Introduction

1.1 Context

It is accepted that optimal gradient algorithms for smooth convex minimization were inaugurated by Yurii Nesterov in 1983 [1] as a version of the *momentum* algorithm by Polyak [2], see also [3]. The word *momentum* indicates the presence of underlying equations of motion. This is known [4] and has been explored before in applied convex optimization. It is worth mentioning that a unification of the conjugate-gradient method, i.e. Hestenes and Stiefel [5], was proposed by Karimi et al. [6], with a state-of-the-art paper on first-order methods published in 2020, cf. Drori and Taylor [7] pointing out to a unification of the first-order algorithms.

Numerous versions of Nesterov's algorithm have been successful in the machine-learning community [8] with reported advantages with respect to momentum (or heavy ball) algorithms. In computational mechanics, few applications were published, and none achieved by explicit Newmark time integration. Carlon *et al.* [9] adopted Donoghue and Candès [10] adaptive restart technique for Nesterov's algorithm in the context of optimal experimental design, but without the time-integration equivalence shown here. In the context of computational homogenization, Schneider [11] made use of the original asynchronous integrator presented by Su, Boyd and Candès [4]. Convergence of Nesterov's algorithm in [9,11] shows the same behavior we observe in this work and the same advantages with respect to the steepest descent method were reported. However, marked differences exist, as no attempt was made toward the use of a damping-insensitive time integrator.

The present work introduces this simple and efficient technique using established time-stepping algorithms to achieve convergence. The fact that a time-stepping algorithm such as Newmark [12] is shown to be *equivalent* to the application

✉ P. Areias
pedro.areias@tecnico.ulisboa.pt

[1] DEM - Mechanical Engineering Department, Instituto Superior Técnico, University of Lisbon, Avenida Rovisco Pais, 1049-001 Lisbon, Portugal

[2] IDMEC, Lisbon, Portugal

[3] Institute of Structural Mechanics, Bauhaus-University Weimar, Marienstraße 15, 99423 Weimar, Germany

of Nesterov's algorithm, allows the use of standard dynamic finite element codes to perform the optimization step.

Beyond the original algorithm, we discuss new time integration methods, the line-search scheme and the restarting program. Applications to fiber orientation optimization (DMO), see Stegmann and Lund [13] are performed in both linear and nonlinear regime, with the complete sensitivity analysis being described in detail for the conservative case.

The paper is organized as follows: Sect. 2 presents Nesterov's algorithm cast as an ordinary differential equation and provides further insights concerning the convergence and solution properties. In Sect. 3 time-integrators are discussed and stability regions are analyzed and the final optimization algorithm is described. In Sect. 4, the adjoint-variable method is described in detail, for both linear and nonlinear cases, with discretization-equivalent formulas and specialization for the strain-energy function. The equations for fiber optimization are introduced in the same section with a particular DMO-like [13–15] parametrization. Validation with More's functions [16] is presented in Sect. 5.2 and Engineering applications of fiber optimization are also presented. The geometrically nonlinear case is also tested, with a simplified hyperelastic constitutive law. Comparisons between the new and existing algorithms are performed. Finally, in Sect. 6 conclusions are drawn concerning this work and its results.

1.2 The original algorithm

We are here concerned with convex unconstrained minimization. We only address functions which are $l-$strongly convex, twice continuously differentiable and with a $L-$Lipschitz-continuous gradient on $\mathbb{R}^n$ $\phi(\mathbf{q}) \in \mathcal{S}^{2,1}_{L,l}(\mathbb{R}^n)$ [3]. Here, $\mathbf{q} \in \mathbb{R}^n$ are the design-variables and $\phi(\mathbf{q})$ is the objective function. The problem consists in

$$\min_{\mathbf{q}\in\mathbb{R}^n} \phi(\mathbf{q}) \tag{1}$$

using information on value $\phi(\mathbf{q})$ and gradient $\nabla_{\mathbf{q}}\phi(\mathbf{q})$ for $\mathbf{q} \in \mathbb{R}^n$. From the aforementioned conditions, the Hessian of $\phi(\mathbf{q})$ is eigenvalue-bounded by l and L

$$L\boldsymbol{I} \succeq \nabla^2_{\mathbf{qq}}\phi(\mathbf{q}) \succeq l\boldsymbol{I} \tag{2}$$

Conditions with the function images are easily interpreted:

$$\phi(\mathbf{q}') \le \phi(\mathbf{q}) + [\nabla_{\mathbf{q}}\phi(\mathbf{q})] \cdot (\mathbf{q}' - \mathbf{q}) + \frac{L}{2}\|\boldsymbol{q} - \mathbf{q}'\|^2 \tag{3}$$

$$\phi(\mathbf{q}') \ge \phi(\mathbf{q}) + [\nabla_{\mathbf{q}}\phi(\mathbf{q})] \cdot (\mathbf{q}' - \mathbf{q}) + \frac{l}{2}\|\boldsymbol{q} - \mathbf{q}'\|^2 \tag{4}$$

hence, our $\mathcal{S}^{2,1}_{L,l}$-function is enveloped by two quadratic models. For first-order methods, the lower complexity bound [3] is given by

$$\frac{\|\mathbf{q}_{k+1} - \mathbf{q}_{\text{sol}}\|}{\|\mathbf{q}_0 - \mathbf{q}_{\text{sol}}\|} \ge \left(\frac{\chi^{1/2} - 1}{\chi^{1/2} + 1}\right)^{2k} \tag{5}$$

where $\mathbf{q}_{\text{sol}}$ is the minimizer, k is the iteration (or step) number and χ is the condition number of the Hessian, satisfying the equality $\chi = L/l$.

Nesterov's algorithm (see [3,4]) makes use of an auxiliary variable $\boldsymbol{p}$, see [4,10], to obtain the required momentum. Starting with the initial condition $\mathbf{q}_0$ and $\boldsymbol{p}_0 = \mathbf{q}_0$, Nesterov's algorithm with step-size h (see [3,4]) is based on the following recurrence formula, $k = 1, 2, \ldots$ is the step number:

$$\mathbf{q}_k = \boldsymbol{p}_{k-1} - h\nabla_{\mathbf{q}}\phi(\boldsymbol{p}_{k-1}) \tag{6}$$

$$\boldsymbol{p}_k = \mathbf{q}_k + \frac{k-1}{k+2}(\mathbf{q}_k - \mathbf{q}_{k-1}) \tag{7}$$

It achieves the lower bound (5) for $h = 1/L$ (the corresponding proof is provided on pages 72–77 of [3]). For computational mechanics applications, the difficulty is the estimation of L. Dedicated iterative algorithms [17] exist for estimating L but this is a problem by itself and inconvenient in the computational mechanics context. The rationale for our approach is the following: since an equivalent ODE can be established for (6–7), it can be solved by an alternative form which is compatible with computational mechanics codes, specifically Finite-Element packages, see [18]. We develop a time-stepping algorithm appropriate for optimization and integration in standard Finite-Element packages. Only the unconstrained case is herein treated, but constrained problems can be solved using an extension of this algorithm. For example SUMT [19] can be used for computational mechanics where elements or cliques are used.

2 Equivalent differential equation and insight on Nesterov's algorithm

Proof of equivalence between Nesterov's algorithm and the solution of an ODE is provided by Su, Boyd and Candès [4]. For the sake of description, we consider a the scalar case, $n = 1$. The final n-dimensional ODE is decoupled and therefore no loss of generality occurs due to this procedure. Let us then consider the following scalar ODE where $t \in \mathbb{R}^{\star}_{+}$ and $c \in \mathbb{N}$:

$$\begin{gathered} t\ddot{q}(t) + c\dot{q}(t) + tq(t) - t = 0 \\ \lim_{t\to 0^+} q(t) = q_{0^+} \end{gathered} \tag{8}$$

Table 1 Time for first solution ($q_{\text{sol}} = 1$) and verification of convergence rate

	t_{sol}	$\lim_{t\to+\infty} t^2\overline{q}^2(t)$
$c = 1$	2.4048	unbounded
$c = 3$	3.8317	$\overline{q}_{\text{sol}}$
$c = 5$	5.1356	$\overline{q}_{\text{sol}}$
$c = 7$	6.3802	$\overline{q}_{\text{sol}}$
$c = 9$	7.5883	$\overline{q}_{\text{sol}}$

(note that $\dot{q}(0) = 0$). Solution of (8) is written introducing the constant α:

$$q(t, c) = 1 + \frac{\alpha!\,(q_{0^+} - 1)}{2^{-\alpha}}\,\frac{J_\alpha(t)}{t^\alpha} \quad \text{with} \quad \alpha = {}^{(c-1)}/_{2} \tag{9}$$

where $J_\alpha(t)$ is the Bessel function of the first kind and $\alpha = {}^{(c-1)}/_{2}$. This is an adaptation of a result given by Apostol [20]. To avoid fractional factorials, in (9) we adopt odd values of c so that α is a non-negative integer. Invariance under time-scaling requires a constant c in (8), see [4]. The solution at the limit is one, i.e. $q_{\text{sol}} = \lim_{t\to+\infty} q(t, c) = 1$. Of course, this limit is insufficient to assess convergence to the equilibrium point, we require two additional quantities. By shifting $q(t)$ as $\overline{q}(t) = q(t) - 1$ (which implies that $\overline{q}_{\text{sol}} = q_{\text{sol}} - 1 = 0$), we determine time for the first root and the limit of the product $t^2\overline{q}^2(t)$, $\lim_{t\to+\infty} t^2\overline{q}^2(t)$. The time for first intersection and the convergence rate are shown in Table 1. The following conclusions are drawn from Table 1:

- Convergence to the exact solution requires that c must be greater or equal than 3.
- The optimal time to obtain the exact solution for the first time is $t_{\text{sol}} = 3.8317$ and it is reached for $c = 3$.

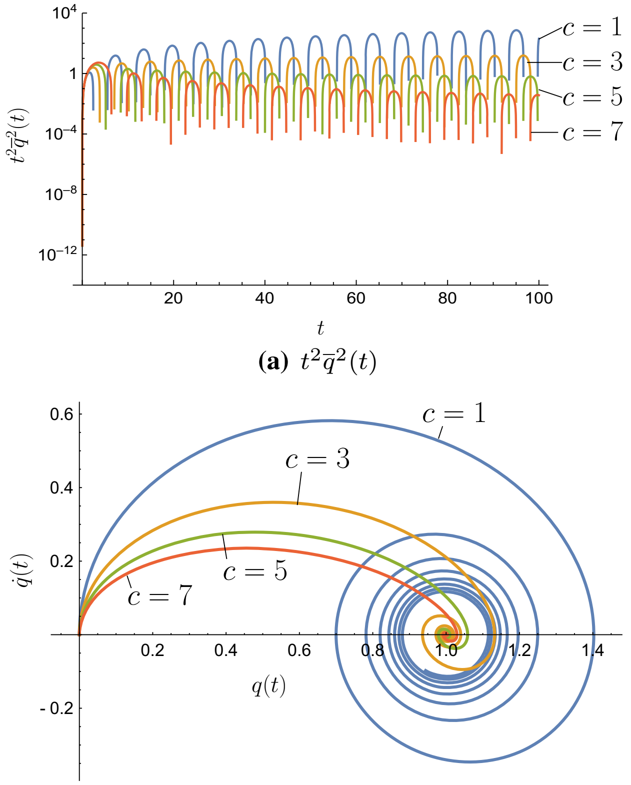
Arguments of time-scaling invariance and optimality for $c = 3$ are key in understanding Nesterov's idea. We point out that (6–7) is an application of an asynchronous explicit integrator (see, e.g. [21]) to the ODE in (8). The algorithm is not stationary, in the sense that in general $\dot{q}(t_{\text{sol}}) \neq 0$. A depiction of the solution (9) is presented in Fig. 1 both in the time-domain and in state space.

We now introduce the following linear variable transformations, where $a > 0$ and b are new parameters, to be discussed later:

$$t = \sqrt{a}t_\star \tag{10}$$

and

$$q(t_\star) = ({}^{a}/_{b})\, r(t_\star) \tag{11}$$



(a) $t^2\overline{q}^2(t)$

(b) Convergence to the solution, state space $q(t) - \dot{q}(t)$. $q_0 = 0$ and $t \in [0, 50]$

Fig. 1 Behavior of Nesterov ODE for $c = 1, 3, 5$ and 7

With these transformations, the ODE becomes:

$$t_\star \overset{\star\star}{r}(t_\star) + c\overset{\star}{r}(t_\star) + t_\star a r(t_\star) - t_\star b = 0 \tag{12}$$

where now a can be identified as "stiffness" and b as the constant "force". Equation (12) has the following solution:

$$r(t_\star, c) = \frac{b}{a}\left[1 + \frac{\alpha!\left(\frac{a}{b} r_{0^+} - 1\right)}{2^{-\alpha}}\,\frac{J_\alpha\left(\sqrt{a}t_\star\right)}{\left(\sqrt{a}t_\star\right)^\alpha}\right] \quad \text{with} \quad \alpha = {}^{(c-1)}/_{2} \tag{13}$$

where the limit of $r(t_\star, c)$ is now $\lim_{t_\star\to+\infty} r(t_\star, c) = {}^{b}/_{a} = r_{\text{sol}}$. We note that further time-scaling is possible for the same limit by modifying simultaneously a and b :

$$r(t_\star, c) = r_{\text{sol}}\left[1 + \frac{\alpha!\left(\frac{r_{0+}}{r_{\text{sol}}} - 1\right)}{2^{-\alpha}}\,\frac{J_\alpha(\sqrt{a}t_\star)}{(\sqrt{a}t_\star)^\alpha}\right] \quad \text{with} \quad \alpha = {}^{(c-1)}/_{2} \tag{14}$$

With (14), we can clearly source term $ar(t_\star) - b$ affects the time scale maintaining the solution at the limit. Since damping is inversely proportional to time, scaling of the source term is in fact scaling of damping. Returning to the original notation, the n-degree-of-freedom case is obtained by

$$t\ddot{\mathbf{q}}(t) + c\dot{\mathbf{q}}(t) + t\boldsymbol{A}\cdot\mathbf{q}(t) - t\boldsymbol{b} = 0 \tag{15}$$

with $\mathbf{q}(t) \in \mathbb{R}^n$. In (15), $\boldsymbol{A}$ is a symmetric positive-definite $n \times n$ matrix and $\boldsymbol{b}$ is a n-dimensional vector. By spectral decomposition of $\boldsymbol{A}$, we have

$$\boldsymbol{A} = \sum_{i=1}^{n} \lambda_i \hat{\boldsymbol{X}}_i \tag{16}$$

where λ_i are the eigenvalues of $\boldsymbol{A}$ and $\hat{\boldsymbol{X}}_i$ are the corresponding normalized eigenvectors, with $\lambda_i \leq \lambda_{i+1}$.

Modal transformation (see, e.g. [22]) is performed as $\mathbf{q}(t) = \hat{\boldsymbol{X}} \cdot \boldsymbol{\eta}(t)$ with $\hat{\boldsymbol{X}}^T \cdot \hat{\boldsymbol{X}} = \boldsymbol{I}$ with $\boldsymbol{I}$ being the identity matrix. New unknowns $\boldsymbol{\eta}(t)$ are defined from $\mathbf{q}(t)$ by inverting matrix $\hat{\boldsymbol{X}}$. Matrix $\hat{\boldsymbol{X}}$ contains the normalized eigenvectors $\hat{\boldsymbol{X}}_i$ disposed by column. This is a standard technique in n-dof vibration [22] and structural dynamics [23]. Classical textbook derivations [22] results in the following modal solution:

$$q_j(t, c) = \frac{\hat{X}_{ji}\omega_i}{\lambda_i}\left[1 - \frac{\alpha!}{2^{-\alpha}} \frac{J_\alpha\left(\sqrt{\lambda_i}t\right)}{\left(\sqrt{\lambda_i}t\right)^\alpha}\right] \quad \text{with} \quad \alpha = {}^{(c-1)}/{}_2 \tag{17}$$

Since the solution converges to $\boldsymbol{q} = \boldsymbol{A}^{-1} \cdot \boldsymbol{b}$, which is the static limit for a damped system, it is also the solution of the following quadratic minimization problem

$$\min_{\mathbf{q}\in\mathbb{R}^n} \left[\phi(\mathbf{q})\right] \text{ where} \tag{18a}$$

$$\phi(\mathbf{q}) = \frac{1}{2}\mathbf{q}^T \cdot \boldsymbol{A} \cdot \mathbf{q} - \mathbf{q} \cdot \boldsymbol{b} \tag{18b}$$

The relevant contribution of this ODE is that we can obtain the solution of the linear system $\boldsymbol{A} \cdot \mathbf{q} = \boldsymbol{b}$ by solely using the product $\boldsymbol{A} \cdot \mathbf{q}$. Generalizing for the condition $\nabla_{\mathbf{q}}\phi(\mathbf{q}) = \mathbf{0}$, the ODE for the optimization problem becomes:

$$t\ddot{\mathbf{q}}(t) + c\dot{\mathbf{q}}(t) + t\nabla_{\mathbf{q}}\phi(\mathbf{q}) = 0 \tag{19}$$

Using principal components and introducing the new unknown θ_i, reduction to a first-order system reads:

$$\theta_i = \dot{\eta}_i \tag{20}$$

$$\begin{Bmatrix} \dot{\theta}_i \\ \dot{\eta}_i \end{Bmatrix} = \begin{bmatrix} -\frac{c}{t} & -\lambda_i \\ 1 & 0 \end{bmatrix} \cdot \begin{Bmatrix} \theta_i \\ \eta_i \end{Bmatrix} + \begin{Bmatrix} \omega_i \\ 0 \end{Bmatrix} \tag{21}$$

Corresponding eigenvalues are given by:

$$\chi_{1,2}^i = -\sqrt{\lambda_i}\left[\xi_i(t) \pm \sqrt{\xi_i(t)^2 - 1}\right] \tag{22}$$

with $\xi_i(t) = \frac{c}{2t\sqrt{\lambda_i}}$ being a time-dependent damping ratio (see, e.g. Meirovitch [22] for this nomenclature). We note that the solution changes from overdamped to underdamped at transition time

$$t^i_{\text{transition}} = \frac{c}{2\sqrt{\lambda_i}}$$

This actually establishes a one-to-one relation between ξ_i and time for each decoupled mode. A global damping scale (replacing the time scale) can also be established, with three regions (all $\xi_i < 1$, at least one $\xi_i < 1$ and all $\xi_i > 1$). This property will be further explored in the following sections.

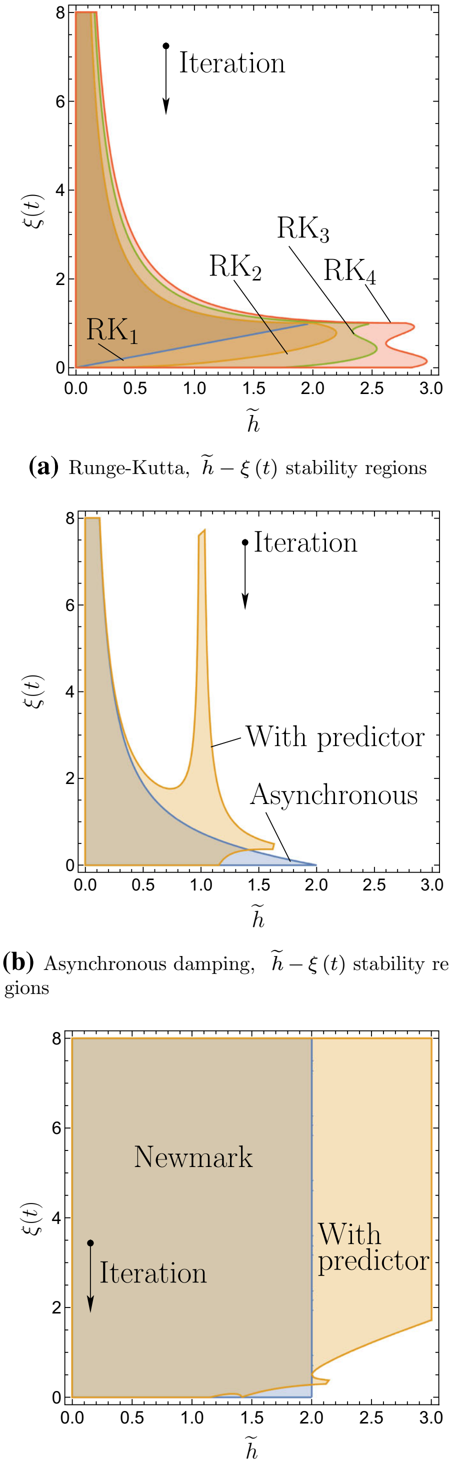
3 Algorithm

Concerning the properties of the problem, we ensure that the explicit numerical integrator is stable for all damping ratios, $\xi \in]0, +\infty[$. Three integrator families are assessed: the Runge–Kutta (RK) of orders 1 up to 4, the central difference explicit integrator, which is a particular case of the Newmark-β and the asyncronous explicit integrator. Introducing the scaled step $\widetilde{h}_i = h\sqrt{\lambda_i}$, amplification matrices and stability regions are shown in Table 4 and Fig. 2, respectively. Specializing for the Nesterov algorithm, for the first step we have $\mathbf{q}_1 = \mathbf{q}_0 - h^2\nabla_{\mathbf{q}}\phi\left(\mathbf{q}_k\right)$ and with$k \geq 1$ we have:

1. Explicit Newmark: $\mathbf{q}_{k+1} = \mathbf{q}_k + \frac{2-hc_{\text{eq}}(k)}{2+hc_{\text{eq}}(k)}\left(\mathbf{q}_k - \mathbf{q}_{k-1}\right) - \frac{2h^2}{[2+hc_{\text{eq}}(k)]}\nabla_{\mathbf{q}}\phi\left(\mathbf{q}_k\right)$
2. Explicit Newmark with predictor/corrector: $\widetilde{\mathbf{q}}_k = \mathbf{q}_k + \frac{2-hc_{\text{eq}}(k)}{2+hc_{\text{eq}}(k)}\left(\mathbf{q}_k - \mathbf{q}_{k-1}\right)$
   $\mathbf{q}_{k+1} = \widetilde{\mathbf{q}}_k - \frac{2h^2}{[2+hc_{\text{eq}}(k)]}\nabla_{\mathbf{q}}\phi\left(\widetilde{\mathbf{q}}_k\right)$
3. Explicit asyncronous: $\mathbf{q}_{k+1} = \mathbf{q}_k + \left[1 - hc_{\text{eq}}(k)\right]\left(\mathbf{q}_k - \mathbf{q}_{k-1}\right) - h^2\nabla_{\mathbf{q}}\phi\left(\mathbf{q}_k\right)$
4. Explicit asyncronous with predictor/corrector: $\widetilde{\mathbf{q}}_k = \mathbf{q}_k + \left[1 - hc_{\text{eq}}(k)\right]\left(\mathbf{q}_k - \mathbf{q}_{k-1}\right)$
   $\mathbf{q}_{k+1} = \widetilde{\mathbf{q}}_k - h^2\nabla_{\mathbf{q}}\phi\left(\widetilde{\mathbf{q}}_k\right)$

Figure 2 shows that only the explicit Newmark integrator is immune to the stability region shrinking in overdamped systems. In terms of stability, predictor versions are also not advantageous. Runge–Kutta (RK) integrators are not appropriate for this application, although RK of orders 3 and 4 could be satisfactory in underdamped systems. The corresponding amplification matrices $\boldsymbol{C}$ (see, Hughes [24] page 492 for the term “amplification matrix” in this context) are shown in Table 4.

A physical interpretation of the previous ODE integrators, which provides a straightforward estimation of h and $c_{\text{eq}}(t)$ is possible. Given a characteristic number for the elements of the Hessian of $\phi\left(\mathbf{q}_k\right)$, a “stiffness” χ, the step size is given



(a) Runge-Kutta, $\widetilde{h} - \xi(t)$ stability regions

(b) Asynchronous damping, $\widetilde{h} - \xi(t)$ stability regions

(c) Explicit Newmark, $\widetilde{h} - \xi(t)$ stability regions

Fig. 2 Stability region in the $\widetilde{h} - \xi(t)$ domain for various time-step integrators

```
Initialization
    $\boldsymbol{q}_{-2} = \boldsymbol{q}_0$
    $\boldsymbol{q}_{-1} = \boldsymbol{q}_0$
    $k = 0$
Iteration loop
    $\phi_k$ and $\nabla\phi_k$
Line-search/restart conditions
    $c_1 = (\boldsymbol{q}_k - \boldsymbol{q}_{k-2}) \cdot \nabla\phi_{k-1} > 0$
    $c_2 = (\boldsymbol{q}_k - \boldsymbol{q}_{k-2}) \cdot \nabla\phi_k > 0$
    $c_3 = (\phi_2 \gg \phi_0) \vee (\phi_2 \gg \phi_1)$
    If$(c_1 \vee c_2 \vee c_3)$
        • Determine $h$ satisfying (27) using backtracking.
          Largest admissible step is considered to be 4
          times the last step in use
        • Set $k = 1$
    End if
Design variable update
    If(k==0)
        $\boldsymbol{q}_1 = \boldsymbol{q}_0 - h^2\nabla\phi_0$
    Else
        $t = hk$
        $c_{\text{eq}} = {}^{c}/_{t}$
        $k_1 = {}^{(2-hc_{\text{eq}})}/_{(2+hc_{\text{eq}})}$
        $k_2 = {}^{2h^2}/_{(2+hc_{\text{eq}})}$
        $\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + k_1(\boldsymbol{q}_k - \boldsymbol{q}_{k-1}) - k_2\nabla\phi_k$
    End if
Check for convergence
    $k = k + 1$
```

Fig. 3 Algorithmic flowchart for the time-integration based on Nesterov's algorithm

by:

$$h = \mathfrak{s}\,(2\chi)^{-1/2} \tag{23}$$

where $\mathfrak{s}$ is a step-scaling, typically a function of the condition number of the Hessian of $\phi\left(\mathbf{q}_k\right)$. Of course, "time" t is now the product of time step h by k: $t = hk$. This results in the following recurrence formula for the explicit Newmark algorithm:

$$\boxed{\mathbf{q}_{k+1} = \mathbf{q}_k + \frac{2k-c}{2k+c}\left(\mathbf{q}_k - \mathbf{q}_{k-1}\right) - \frac{k\mathfrak{s}^2}{\chi\,(2k+c)}\nabla_{\mathbf{q}}\phi\left(\mathbf{q}_k\right) \quad k \geq 1} \tag{24}$$

From which a direction $\boldsymbol{d}_k = \mathbf{q}_{k+1} - \mathbf{q}_k$ can be extracted. A few remarks are in order with respect to this direction:

1. Nesterov's choice $c = 3$ is clear in (24) as negative momentum would be present otherwise for the first $c/2$ steps.
2. Stationarity at the solution is not satisfied, but a more favorable direction is obtained when compared with the original algorithm.
3. Not all directions are descent. If $\left(\mathbf{q}_k - \mathbf{q}_{k-1}\right) \cdot \nabla_{\mathbf{q}} \phi\left(\mathbf{q}_k\right) > 0$ the descent condition $\left(\mathbf{q}_{k+1} - \mathbf{q}_k\right) \cdot \nabla_{\mathbf{q}} \phi\left(\mathbf{q}_k\right) < 0$ is only conditionally satisfied for $k \geq 2$ by

$$\mathfrak{s} > \sqrt{\frac{\chi\left(2k - c\right)\left(\mathbf{q}_k - \mathbf{q}_{k-1}\right) \cdot \nabla_{\mathbf{q}} \phi\left(\mathbf{q}_k\right)}{k\left[\nabla_{\mathbf{q}} \phi\left(\mathbf{q}_k\right)\right]^2}} \tag{25}$$

The constant χ can be replaced by an estimate of the gradient Lipschitz constant L. A value of χ can be also required in the context of constrained minimization, as part of the penalty term calculations. We now introduce two essential ingredients: conditional restart and line-search procedures. We reset the equivalent damping $c_{\text{eq}}(k)$ when one of the following descent conditions occurs:

- $k - k_\star \geq 2$ and $\left(\mathbf{q}_{k+1} - \mathbf{q}_{k-1}\right) \cdot \nabla_{\mathbf{q}} \phi\left(\mathbf{q}_k\right) > 0$ or $\left(\mathbf{q}_{k+1} - \mathbf{q}_k\right) \cdot \nabla_{\mathbf{q}} \phi\left(\mathbf{q}_k\right) > 0$
- $c_{\text{eq}}(k) = \frac{c}{k - k_\star}$ where $k_\star$ is the last step which failed the aforementioned conditions, which it is initialized as $k_\star = 1$.

A version with a two-step descent condition was introduced by Donoghue and Candès [10] who related the interval between restarts with the condition number of a quadratic function as $\Delta k_\star \cong (\pi + 3/2)\sqrt{L/l}$. The ratio $\mathfrak{s}^2/\chi$ is here replaced by $2h$, where h is the step size:

$$\mathfrak{s}^2/\chi = 2h^2 \tag{26}$$

The value of h^2 is determined by the sufficient decrease rule, see [25,26]:

$$\phi\left(\mathbf{q}_k\right) - c_s h^2 \nabla\phi_k^2 \geq \phi\left(\mathbf{q}_k - h^2 \nabla\phi_k\right) \tag{27}$$

by the usual backtracking algorithm. A fixed value of $c_s = 1 \times 10^{-4}$ is adopted. In addition, the Dai-Yuan [27] extension of the Barzilai–Borwein method [28] is adopted to calculate h for steps other than the first:

$$h^2 = \frac{\Delta\mathbf{q}_{k-1} \cdot \Delta\mathbf{q}_{k-1}}{2\left[\phi\left(\mathbf{q}_{k-1}\right) - \phi\left(\mathbf{q}_k\right) + \nabla_{\mathbf{q}} \phi\left(\mathbf{q}_k\right) \cdot \Delta\mathbf{q}_{k-1}\right]} \tag{28}$$

where $\Delta\mathbf{q}_{k-1} = \mathbf{q}_k - \mathbf{q}_{k-1}$. We therefore employ the minimum step from equations (27–28). The recalculation of h within an explicit integration algorithm such as ours is a delicate operation and cannot be performed too frequently [21] since there is a destabilization effect. As an algorithmic policy, we only update the step at each conditional restart. This ensures that each complete Nesterov iteration is performed with the same $\mathfrak{s}$. The Algorithm is shown in Fig. 3. The complete code, including all line-search variants and restart policies is public-domain at Github [29].

4 Adjoint variable method and application to fiber-reinforced laminae

When the optimization problem statement contains one function and multiple design variables, the adjoint variable method is the more efficient sensitivity technique to adopt (for the first work in control theory, see [30]). We consider $\boldsymbol{x}$ to be the state solution, a list of degrees-of-freedom corresponding to displacements. In addition, the design variables are, as previously introduced, arranged in an array $\mathbf{q}$. Let us consider one function $\Psi(\boldsymbol{x}, \mathbf{q})$ which is here identified as the objective function. We use a one-to-one mapping between $\phi(\mathbf{q})$ and $\Psi(\boldsymbol{x}, \mathbf{q})$ as:

$$\phi(\mathbf{q}) = \Psi\left[\hat{\boldsymbol{x}}(\mathbf{q}), \mathbf{q}\right] \tag{29}$$

where an implicit relation between $\boldsymbol{x}$ and $\mathbf{q}$ exists, in this case formalized as $\hat{\boldsymbol{x}}(\mathbf{q})$. To materialize the alternating minimization algorithm, the total derivative with respect to $\mathbf{q}$ follows the analysis stage where $\boldsymbol{x}$ satisfies equilibrium. Two cases are considered: Case I where state is linear in $\boldsymbol{x}$ and Case II where state is nonlinear in $\boldsymbol{x}$.

4.1 Case I: state is linear in $\boldsymbol{x}$

A linear governing state equation is assumed, $\boldsymbol{A}(\mathbf{q}) \cdot \boldsymbol{x} = \boldsymbol{b}(\mathbf{q})$ with $\boldsymbol{b}(\mathbf{q})$ being the load vector and $\boldsymbol{A}(\mathbf{q})$ the stiffness matrix. Both are for now considered to be functions of $\mathbf{q}$. Sensitivity of $\Psi(\boldsymbol{x}, \mathbf{q})$ is defined as:

$$\frac{\mathrm{d}\Psi(\boldsymbol{x}, \mathbf{q})}{\mathrm{d}q_j} = \frac{\partial\Psi(\boldsymbol{x}, \mathbf{q})}{\mathrm{d}q_j} + \frac{\partial\Psi(\boldsymbol{x}, \mathbf{q})}{\partial x_k}\frac{\mathrm{d}x_k}{\mathrm{d}q_j} \tag{30}$$

Using the state equation in terms of components, we obtain:

$$\frac{\mathrm{d}x_i}{\mathrm{d}q_j} = \left[\boldsymbol{A}^{-1}(\mathbf{q})\right]_{ik}\left(\frac{\mathrm{d}b_k(\mathbf{q})}{\mathrm{d}q_j} - \frac{\partial A_{kl}(\mathbf{q})}{\partial q_j}x_l\right) \tag{31}$$

By pre-multiplying both sides by $\frac{\partial\Psi(\boldsymbol{x},\mathbf{q})}{\partial x_i}$, the following result is achieved:

$$\frac{\partial\Psi(\boldsymbol{x}, \mathbf{q})}{\partial x_i}\frac{\mathrm{d}x_i}{\mathrm{d}q_j} = \frac{\partial\Psi(\boldsymbol{x}, \mathbf{q})}{\partial x_i}\left[\boldsymbol{A}^{-1}(\mathbf{q})\right]_{ik} \times\left(\frac{\mathrm{d}b_k(\mathbf{q})}{\mathrm{d}q_j} - \frac{\partial A_{kl}(\mathbf{q})}{\partial q_j}x_l\right) \tag{32}$$

Therefore, we solve the following adjoint linear system for $\boldsymbol{\lambda}$:

$$\boxed{\boldsymbol{A}\left(\mathbf{q}\right)^{T}\cdot\boldsymbol{\lambda}=\frac{\partial\Psi\left(\boldsymbol{x},\mathbf{q}\right)}{\partial\boldsymbol{x}}} \tag{33}$$

From this solution for $\boldsymbol{\lambda}$, the final sensitivity obtained from (32) is given by:

$$\boxed{\frac{\mathrm{d}\Psi(\boldsymbol{x},\mathbf{q})}{\mathrm{d}q_j}=\frac{\partial\Psi(\boldsymbol{x},\mathbf{q})}{\partial q_j}+\left(\frac{\mathrm{d}b_k}{\mathrm{d}q_j}-\frac{\partial A_{kl}\left(\mathbf{q}\right)}{\partial q_j}x_l\right)\lambda_k} \tag{34}$$

4.2 Case II: state is nonlinear in $\boldsymbol{x}$

In this case, the discrete form of equilibrium is written as:

$$\boldsymbol{f}\left(\boldsymbol{x},\mathbf{q}\right)=\mathbf{0} \tag{35}$$

where $\boldsymbol{f}$ is typically identified as the residual. In mechanical engineering circles, it is common to define the residual $\boldsymbol{f}\left(\boldsymbol{x},\mathbf{q}\right)$ as the difference between the internal $\boldsymbol{i}\left(\boldsymbol{x},q\right)$ and external forces $\boldsymbol{e}(\mathbf{q})$:

$$\boldsymbol{f}\left(\boldsymbol{x},\mathbf{q}\right)=\boldsymbol{i}\left(\boldsymbol{x},\mathbf{q}\right)-\boldsymbol{e}(\mathbf{q}) \tag{36}$$

Newton iteration to solve (35) for $\boldsymbol{x}$ is obtained as:

$$\boldsymbol{A}\left(\boldsymbol{x}_{(i)},\mathbf{q}\right)\cdot\left(\boldsymbol{x}_{(i+1)}-\boldsymbol{x}_{(i)}\right)=-\boldsymbol{f}\left(\boldsymbol{x}_{(i)},\mathbf{q}\right) \tag{37}$$

with

$$\boldsymbol{A}(\boldsymbol{x}_{(i)},\mathbf{q})=\left.\frac{\partial\boldsymbol{f}(\boldsymbol{x},\mathbf{q})}{\partial\boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_{(i)}} \tag{38}$$

Matrix $\boldsymbol{A}(\boldsymbol{x},\mathbf{q})$, which is the Jacobian of $\boldsymbol{f}(\boldsymbol{x},\mathbf{q})$ with respect to $\boldsymbol{x}$ is often called the ≪ stiffness ≫ matrix. This iteration (37) continues until $\|\boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{i}},\mathbf{q})\|\cong 0$. After this condition is satisfied, the solution of (35) for $\boldsymbol{x}$ is typically performed by Newton iteration:

$$\boldsymbol{A}(\boldsymbol{x},\mathbf{q})\cdot\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\mathbf{q}}=-\frac{\partial\boldsymbol{f}(\boldsymbol{x},\mathbf{q})}{\partial\mathbf{q}}\quad\text{if}\quad\boldsymbol{f}(\boldsymbol{x},\mathbf{q})=\mathbf{0}$$

Component-wise, the sensitivity of $\Psi(\boldsymbol{x},\boldsymbol{q})$ is written as:

$$\frac{\mathrm{d}\Psi(\boldsymbol{x},\mathbf{q})}{\mathrm{d}q_j}=\frac{\partial\Psi(\boldsymbol{x},\mathbf{q})}{\mathrm{d}q_j}+\frac{\partial\Psi\left(\boldsymbol{x},\mathbf{q}\right)}{\partial x_k}\frac{\mathrm{d}x_k}{\mathrm{d}q_j}$$

Repeating the process used for the linear case, we obtain

$$\boxed{\boldsymbol{A}\left(\mathbf{q}\right)^{T}\cdot\boldsymbol{\lambda}=\frac{\partial\Psi\left(\boldsymbol{x},\mathbf{q}\right)}{\partial\boldsymbol{x}}} \tag{39}$$

Therefore, the final formula for the sensitivity is given by:

$$\boxed{\frac{\mathrm{d}\Psi(\boldsymbol{x},\mathbf{q})}{\mathrm{d}q_j}=\frac{\partial\Psi(\boldsymbol{x},\mathbf{q})}{\partial q_j}-\frac{\mathrm{d}f_k\left(\boldsymbol{x},\mathbf{q}\right)}{\mathrm{d}q_j}\lambda_k} \tag{40}$$

4.3 Discretization and specialization to compliance

We now use the FE discretization to calculate the second term. If $\Psi(\boldsymbol{x},\mathbf{q})$ is defined at the element level as a sum over n_e elements, we can write its sensitivity as:

$$\frac{\mathrm{d}\Psi(\boldsymbol{x},\mathbf{q})}{\mathrm{d}q_j}=\sum_{e=1}^{n_e}\frac{\partial\Psi^{e}(\boldsymbol{x},\mathbf{q})}{\partial q_j^{e}}-\sum_{e=1}^{n_e}\frac{\partial f_k^{e}(\boldsymbol{x},\mathbf{q})}{\partial q_j^{e}}\lambda_k^{e} \tag{41}$$

with

$$\boldsymbol{A}^{T}\cdot\boldsymbol{\lambda}=\frac{\partial\Psi\left(\boldsymbol{x},\mathbf{q}\right)}{\partial\boldsymbol{x}}$$

where element summation applies:

$$\boldsymbol{A}^{T}=\sum_{e=1}^{n_e}\boldsymbol{A}_e^{T} \tag{42}$$

$$\frac{\partial\Psi\left(\boldsymbol{x},\mathbf{q}\right)}{\partial\boldsymbol{x}}=\sum_{e=1}^{n_e}\frac{\partial\Psi^{e}\left(\boldsymbol{x},\mathbf{q}\right)}{\partial\boldsymbol{x}} \tag{43}$$

The three steps of our algorithm are summarized in Table 2. A formalization of similar algorithms is presented by Byrne [31] with technical details not explored here.

4.4 Specialization for the strain energy function

We now consider the strain energy to be adopted for $\Psi\left(\boldsymbol{x},\mathbf{q}\right)$ and assume that the external forces are independent of $\mathbf{q}$, $\boldsymbol{e}\left(\mathbf{q}\right)\equiv\boldsymbol{e}$. The objective function is written as:

$$\Psi(\boldsymbol{x},\mathbf{q})=\sum_{e=1}^{n_e}\Psi^{e}\left(\boldsymbol{x},\mathbf{q}\right)=\sum_{e=1}^{n_e}\int_{\Omega_e}\psi\left[\boldsymbol{C}\left(\boldsymbol{x}\right),\mathbf{q}\right]\mathrm{d}V \tag{44}$$

where Ω_e is the element integration domain, $\boldsymbol{C}\left(\boldsymbol{x}\right)$ is the right Cauchy-Green tensor and ψ is the strain energy density function (see, e.g. Wriggers [32]) for this nomenclature. This results as follows:

$$\frac{\partial\Psi\left(\boldsymbol{x},\mathbf{q}\right)}{\partial\mathbf{q}}=\sum_{e=1}^{n_e}\frac{\partial\Psi_e\left(\boldsymbol{x},\mathbf{q}\right)}{\partial\mathbf{q}}=\sum_{e=1}^{n_e}\int_{\Omega_e}\frac{\partial\psi\left[\boldsymbol{C}\left(\boldsymbol{x}\right),\mathbf{q}\right]}{\partial\mathbf{q}}\mathrm{d}V \tag{45}$$

$$\frac{\partial\Psi\left(\boldsymbol{x},\mathbf{q}\right)}{\partial\boldsymbol{x}}=\boldsymbol{i}\left(\boldsymbol{x},\mathbf{q}\right)=\sum_{e=1}^{n_e}\int_{\Omega_e}\frac{\partial\psi\left[\boldsymbol{C}\left(\boldsymbol{x}\right),\mathbf{q}\right]}{\partial\boldsymbol{x}}\mathrm{d}V \tag{46}$$

Table 2 Optimization substeps in the nonlinear case

I - Solution for the state $\boldsymbol{x}$	
Iterate for $\boldsymbol{x}$	$\boldsymbol{A}(\boldsymbol{x}_{(i)}, \mathbf{q}) \cdot \left(\boldsymbol{x}_{(i+1)} - \boldsymbol{x}_{(i)}\right) = -\boldsymbol{f}(\boldsymbol{x}_{(i)}, \mathbf{q})$
II - Sensitivity analysis	
(1) Determine specific sensitivities	$\frac{\partial \Psi(\boldsymbol{x},\mathbf{q})}{\partial \mathbf{q}}$, $\frac{\partial \Psi(\boldsymbol{x},\mathbf{q})}{\partial \boldsymbol{x}}$ and $\frac{\partial \boldsymbol{f}(\boldsymbol{x},\mathbf{q})}{\partial \mathbf{q}}$
(2) Determine $\partial\Psi(\boldsymbol{x},\mathbf{q})/\partial\boldsymbol{x}$ and solve for $\boldsymbol{\lambda}$	$\boldsymbol{\lambda} = \boldsymbol{A}^{-T} \cdot \left[\frac{\partial \Psi(\boldsymbol{x},\mathbf{q})}{\partial \boldsymbol{x}}\right]$
(3) Determine $\frac{\mathrm{d}\Phi(\boldsymbol{x},\mathbf{q})}{\mathrm{d}\mathbf{q}}$	$\sum_{e=1}^{n_e} \left[\frac{\partial \Psi^e(\boldsymbol{x},\mathbf{q})}{\partial \mathbf{q}^e} - \boldsymbol{\lambda}^e \cdot \frac{\mathrm{d}\boldsymbol{f}^e(\boldsymbol{x},\mathbf{q})}{\mathrm{d}\boldsymbol{q}^e}\right]$
III - Optimization	
Update $\boldsymbol{p}$	$\mathbf{q}_{(j+1)}\left[\mathbf{q}_{(j)}, \mathbf{q}_{(j-1)}, \Psi(\boldsymbol{x}, \mathbf{q}), \frac{\partial \Psi(\boldsymbol{x},\mathbf{q})}{\partial \mathbf{q}}\right]$
↻	

Table 3 Number of function evaluations for the test problems (see [16])

Problem	Function	CG		Nesterov/Newmark	
		Func+grad	Grad	Func+grad	Grad
1	Rosenbrock	925	229	**167**	**127**
2	Freudenstein and Roth	1016	233	**286**	**198**
3	Beale	410	94	**84**	**56**
4	Powell singular	867	200	**133**	**102**
5	Wood	1609	370	**446**	**335**
6	Brown and Dennis ($m = 4$)	233	**55**	**150**	129
7	Biggs EXP6 ($m = 6$)	842	195	**187**	**145**
8	Trigonometric function ($m = 6$)	169	39	67	34

Problem	Details concerning the test functions	
	$\phi(\mathbf{q})$	$n = \#\boldsymbol{q}_0$
1	$100(q_2 - q_1^2)^2 + (1 - q_1)^2$	$\mathbf{q}_0 = \{-1.2, 1\}$
2	$\{-13 + q_1 + [(5 - q_2)q_2 - 2]q_2\}^2 +$ $\{-29 + q_1 + [(q_2 + 1)q_2 - 14]q_2\}^2$	$\mathbf{q}_0 = \{0.5, -2\}$
3	$[1.5 - q_1(1 - q_2)]^2 + [2.25 - q_1(1 - q_2^2)]^2$ $+[2.625 - q_1(1 - q_2^3)]^2$	$\mathbf{q}_0 = \{1, 1\}$
4	$(q_1 + 10q_2)^2 + 5(q_3 - q_4)^2$ $+(q_2 - 2q_3)^4 + 10(q_1 - q_4)^4$	$\mathbf{q}_0 = \{3, -1, 0, 1\}$
5	$100(q_2 - q_1^2) + (1 - q_1)^2$ $+90(q_4 - q_3^2)^2 + (1 - q_3)^2$ $+10(q_2 + q_4 - 2)^2 + 0.1(q_2 - q_4)^2$	$\mathbf{q}_0 = \{-3, -1, -3, -1\}$
6	$\sum_{i=1}^{4}\{[q_1 + {}^{iq_2}/_5 - \exp({}^{i}/_5)]^2 + [q_3 + q_4 \sin({}^{i}/_5) - \cos({}^{i}/_5)]^2\}^2$	$\mathbf{q}_0 = \{25, 5, -5, -1\}$
7	$\sum_{i=1}^{6}\{q_3 \exp(-{}^{iq_1}/_{10}) - q_4 \exp(-{}^{iq_2}/_{10}) + q_6 \exp(-iq_5) - \exp(-{}^{i}/_{10}) + 5 \exp(-i) - 3 \exp(-{}^{2i}/_5)\}^2$	$\mathbf{q}_0 = \{1, 2, 1, 1, 1, 1\}$
8	$\sum_{i=1}^{6}[6 + i(1 - \cos q_i) - \sin q_i - \sum_{j=1}^{6} \cos q_j]^2$	$\mathbf{q}_0 = \frac{1}{6}\{1, 1, 1, 1, 1, 1\}$

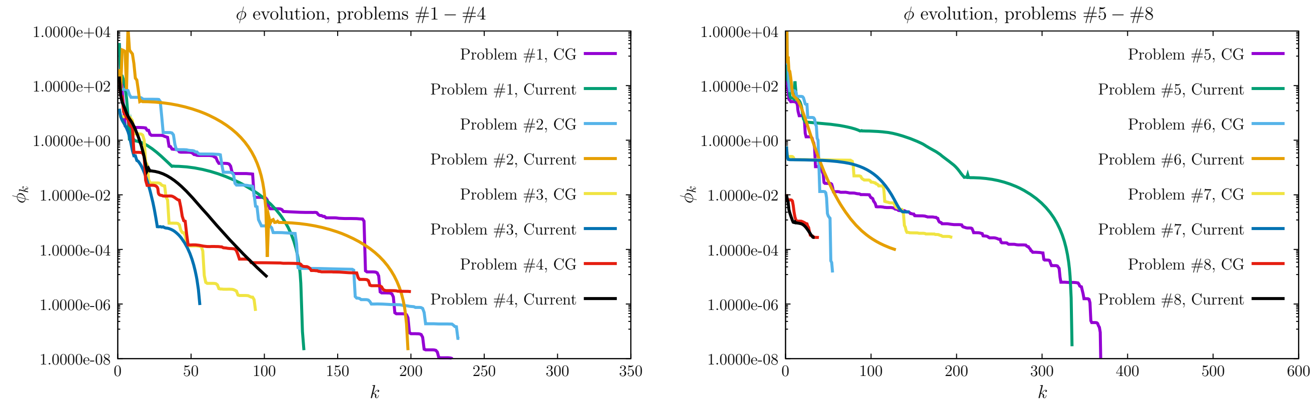
$$\frac{\partial^2 \Psi(\boldsymbol{x}, \mathbf{q})}{\partial \boldsymbol{x}^2} = \boldsymbol{A}(\boldsymbol{x}, \mathbf{q}) = \sum_{e=1}^{n_e} \int_{\Omega_e} \frac{\partial^2 \psi\left[\boldsymbol{C}(\boldsymbol{x}), \mathbf{q}\right]}{\partial \boldsymbol{x}^2} \mathrm{d}V \quad (47)$$

$$\frac{\partial^2 \Psi(\boldsymbol{x}, \mathbf{q})}{\partial \boldsymbol{x} \partial \mathbf{q}} = \frac{\partial \boldsymbol{i}(\boldsymbol{x}, \mathbf{q})}{\partial \mathbf{q}} = \sum_{e=1}^{n_e} \int_{\Omega_e} \frac{\partial^2 \psi\left[\boldsymbol{C}(\boldsymbol{x}), \mathbf{q}\right]}{\partial \boldsymbol{x} \partial \mathbf{q}} \mathrm{d}V \quad (48)$$

These operations are performed in SimPlas [18] resorting to Wolfram Mathematica [33] and Acegen [34].

4.5 Application to fiber-reinforced laminae

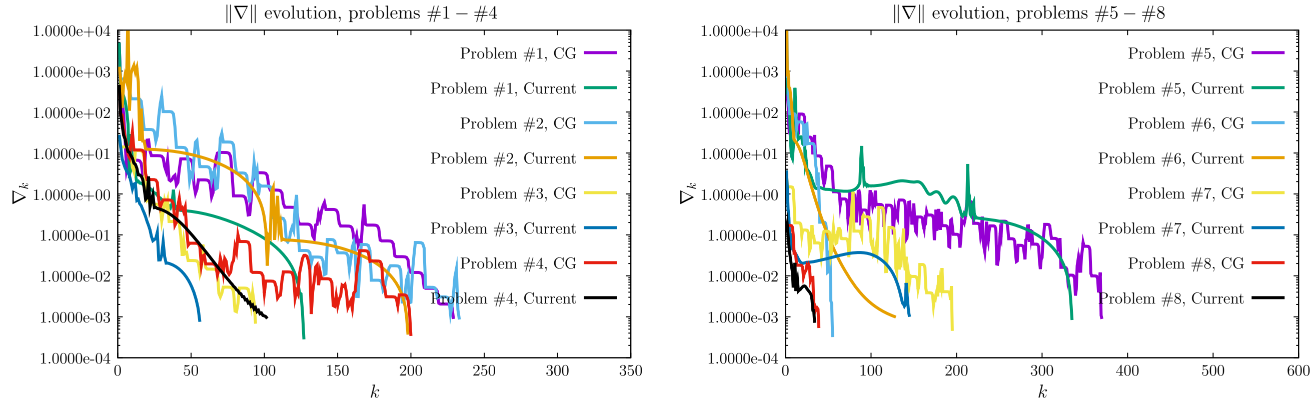
Discrete material optimization (DMO) was pioneered by Sigmund and Torquato [35] and further formalized by Stegmann and Lund [13,36]. We consider a fiber-reinforced material which has a single angle of orientation. For a given finite element e we consider the optimization variables $\mathbf{q}_e$ which are here adopted to represent an interpolated quan-



(a) ϕ_k for problems #1 − #4

(b) ϕ_k for problems #5 − #8

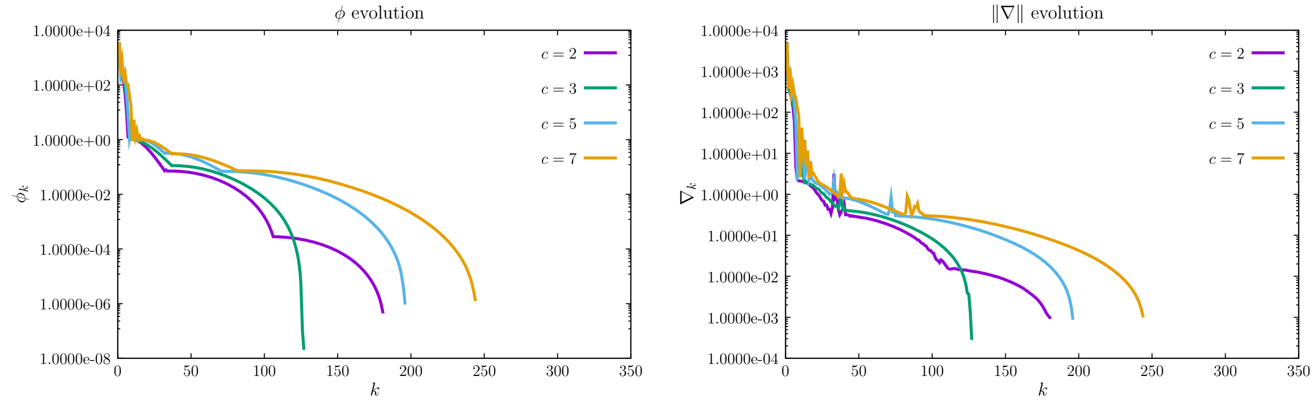
Fig. 4 Evolution of ϕ_k with k for the benchmark functions of Table 3



(a) $\|\nabla\phi_k\|$ for problems #1 − #4

(b) $\|\nabla\phi_k\|$ for problems #5 − #8

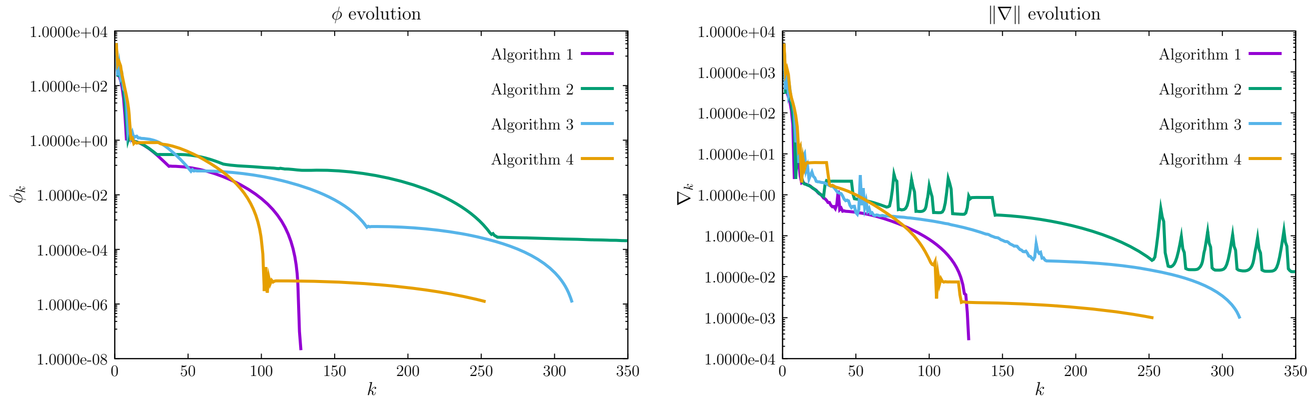
Fig. 5 Evolution of ϕ_k and $\|\nabla\phi_k\|$ with k for the benchmark functions of Table 3



(a) Evolution of ϕ_k

(b) Evolution of $\|\nabla\phi_k\|$

Fig. 6 Evolution of ϕ_k and $\|\nabla\phi_k\|$ with k for the Rosenbrock problem with $c = 2, 3, 5$ and 7.



(a) Evolution of ϕ_k **(b)** Evolution of $\|\nabla\phi_k\|$

Fig. 7 Evolution of ϕ_k and $\|\nabla\phi_k\|$ with k for the Rosenbrock problem with $c = 2, 3, 5$ and 7.

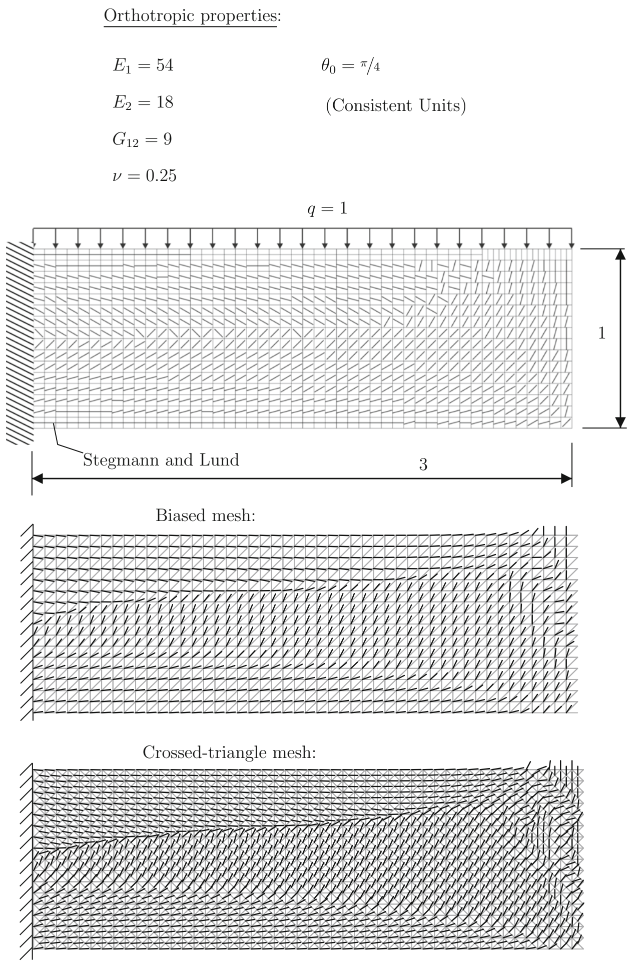


Fig. 8 Cantilever beam from Stegmann and Lund [13]

tity $w_e \in]-1, 1[$. With that goal, we use the parent-domain coordinates $\boldsymbol{\xi}$ and obtain w_e as:

$$w_e(\boldsymbol{\xi}) = \tanh\left[\boldsymbol{N}(\boldsymbol{\xi}) \cdot \mathbf{q}_e\right] \tag{49}$$

where $\boldsymbol{N}(\boldsymbol{\xi})$ is the traditional shape-function vector and $\boldsymbol{\xi}$ are the parent-domain coordinates. With this notation, we write the following interpolation:

$$\mathcal{C}(w_e) = \sum_{k=1}^{7} L_k(w_e)\,\mathcal{C}_{\alpha_k} \tag{50}$$

with $\alpha_k = -\frac{\pi}{2} + \left[\frac{k-1}{6}\right]\pi$. In (50), $L_k(w_e)$ with $k = 1, \ldots, 7$ are Lagrange shape function polynomials and $\mathcal{C}_{\alpha_k}$ is given in plane-stress by:

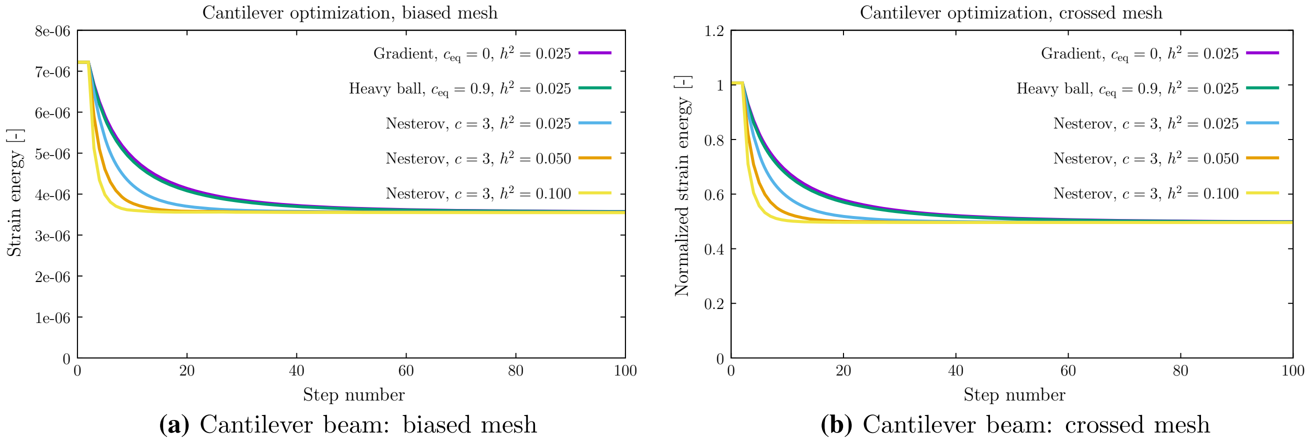
$$\mathcal{C}_{\alpha_k} = \boldsymbol{T}_{\varepsilon}^{T}(\alpha_k) \begin{bmatrix} 1/E_1 & -\nu_{12}/E_1 & 0 \\ -\nu_{12}/E_1 & 1/E_2 & 0 \\ 0 & 0 & 1/G_{12} \end{bmatrix}^{-1} \boldsymbol{T}_{\varepsilon}(\alpha_k) \tag{51}$$

with $\boldsymbol{T}_{\varepsilon}(\alpha_k)$ being the strain transformation matrix in Voigt form:

$$\boldsymbol{T}_{\varepsilon}(\alpha_k) = \begin{bmatrix} \cos^2(\alpha_k) & \sin^2(\alpha_k) & \cos(\alpha_k)\sin(\alpha_k) \\ \sin^2(\alpha_k) & \cos^2(\alpha_k) & -\cos(\alpha_k)\sin(\alpha_k) \\ -\sin(2\alpha_k) & \sin(2\alpha_k) & \cos(2\alpha_k) \end{bmatrix} \tag{52}$$

As a side note, $\boldsymbol{T}_{\sigma}(\alpha_k) = [\boldsymbol{T}_{\varepsilon}(\alpha_k)]^{-T} = [\boldsymbol{T}_{\varepsilon}(-\alpha_k)]^{T}$. Lagrange shape functions in (50) correspond to a 6−th degree polynomial:

$$\boldsymbol{L}(w_e) = \begin{pmatrix} \frac{1}{80}(w_e-1)w_e(3w_e-2)(3w_e-1)(3w_e+1)(3w_e+2) \\ -\frac{9}{40}(w_e-1)w_e(w_e+1)(3w_e-2)(3w_e-1)(3w_e+1) \\ \frac{9}{16}(w_e-1)w_e(w_e+1)(3w_e-2)(3w_e-1)(3w_e+2) \\ \frac{1}{4}\left(4-w_e^2\left(7-9w_e^2\right)^2\right) \\ \frac{9}{16}(w_e-1)w_e(w_e+1)(3w_e-2)(3w_e+1)(3w_e+2) \\ -\frac{9}{40}(w_e-1)w_e(w_e+1)(3w_e-1)(3w_e+1)(3w_e+2) \\ \frac{1}{80}w_e(w_e+1)(3w_e-2)(3w_e-1)(3w_e+1)(3w_e+2) \end{pmatrix} \tag{53}$$



(a) Cantilever beam: biased mesh

(b) Cantilever beam: crossed mesh

Fig. 9 Compliance evolution as a function of the *step* number

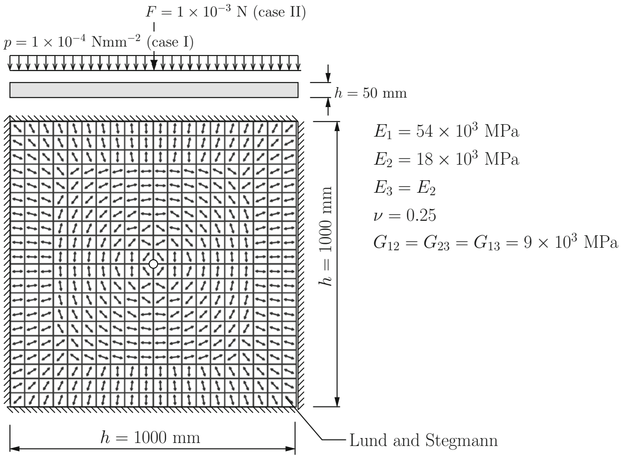

$F = 1 \times 10^{-3}$ N (case II)
$p = 1 \times 10^{-4}$ Nmm^{-2} (case I)
$h = 50$ mm
$E_1 = 54 \times 10^3$ MPa
$E_2 = 18 \times 10^3$ MPa
$E_3 = E_2$
$\nu = 0.25$
$G_{12} = G_{23} = G_{13} = 9 \times 10^3$ MPa
$h = 1000$ mm
$h = 1000$ mm
Lund and Stegmann

Fig. 10 Square plate: relevant data and results from Lund and Stegmann [36]

This is inserted in the framework discussed above into our code [18]. We omit the finite element technology involved, as several papers by our group have been published discussing the details. Note that the derivatives are determined by using Wolfram Mathematica [33] with the Acegen add-on [34].

5 Numerical testing

5.1 Benchmark test functions

For comparison, we use the Polak–Ribière version [37] of the nonlinear conjugate-gradient method [25,26] with the same line-search algorithm adopted in this work. Tolerance for the gradient norm is here $tol = 1 \times 10^{-3}$ in both methods. Table 3 shows the number of function evaluations for 8 benchmark problems reported in [16]. In Table 3, the best values are highlighted in bold. We can observe that our algorithm makes use of fewer function and gradient evaluations in all problems except problem #6 where much fewer iterations (and hence gradient evaluations) are performed by the conjugate-gradient method.

The evolution of ϕ_k for the 8 problems is shown in Fig. 4 and for $\|\nabla\phi_k\|$ is shown in Fig. 5. We conclude that behavior is clearly excellent for the last part of iteration process, whereas CG tends to perform better in the beginning. An exception is problem 6, where CG outperforms the Nesterov/Newmark algorithm. For the Rosenbrock problem, we inspect the effect of c and the integrator. Figure 6 confirms the advantage of $c = 3$ with respect to other choices. In addition, the superiority of the Newmark integrator is confirmed in Fig. 7, even with respect to the asynchronous original proposal by Nesterov. An advantage of the Nesterov/Newmark integration algorithm is that line-search is used infrequently, only at the first iteration and when restarting is flagged. This is particularly useful for Finite Element analysis, since the cost of the objective function evaluation can be nearly as high as the gradient.

5.2 Verification examples: optimization of fiber orientations

We first assess the robustness of the algorithm using solely fiber orientations. The first test is the cantilever beam by Stegmann and Lund [13]. Orthotropic properties and plane-stress conditions are considered, with the relevant properties shown in Fig. 8. In this figure we also show the results from Stegmann and Lund for comparison.

Figure 9 shows the results for the evolution of the strain energy as a function of the step number. Stegmann and Lund mention 100 iterations for convergence, and it can be observed that much fewer steps/iterations are required in our algorithm. We can conclude the following:

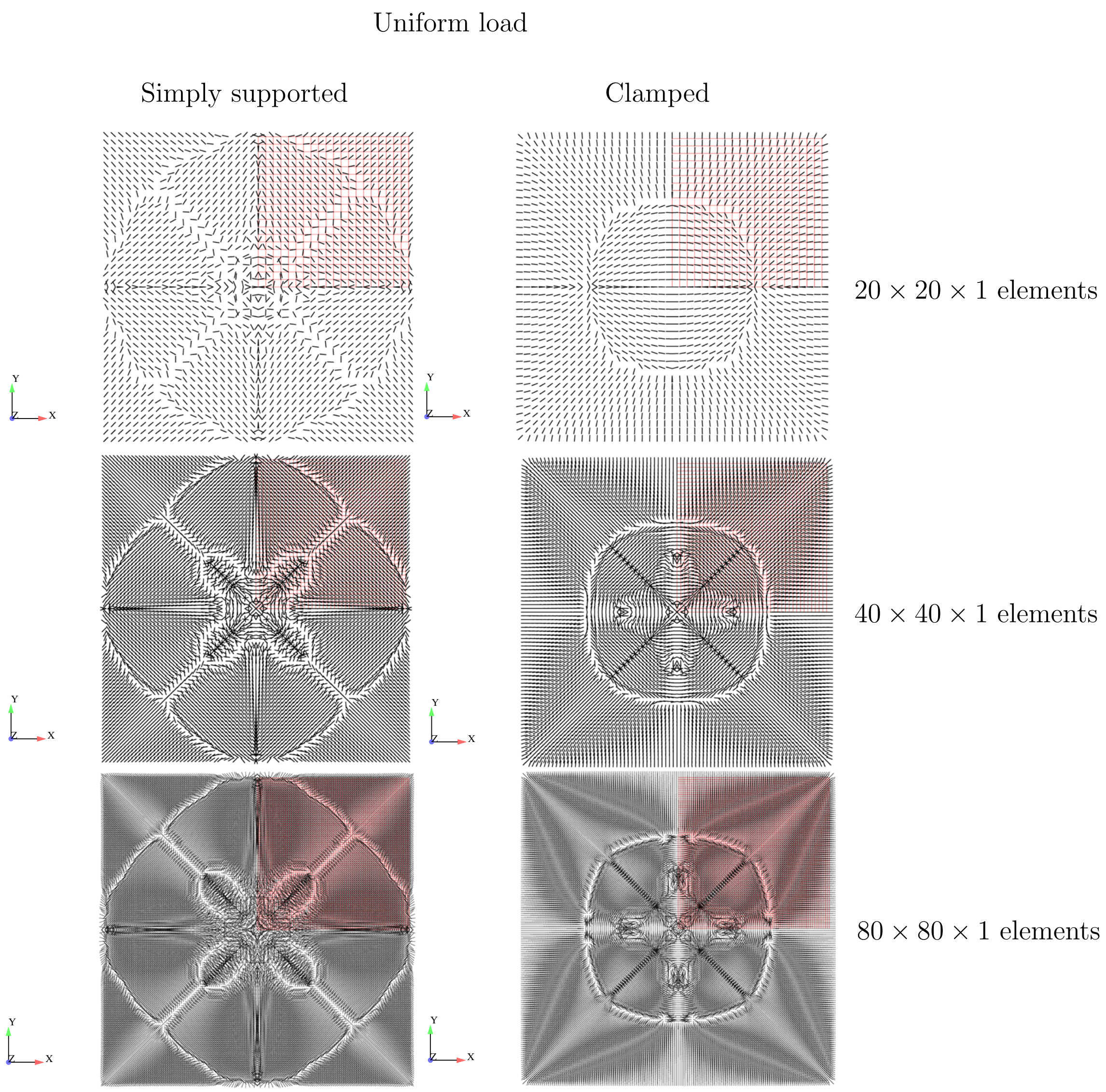


Fig. 11 Square plate under uniform pressure: orientation results as a function of mesh density

- The heavy ball version with $c = 0.9$ presents faster convergence than the standard gradient-based optimization.
- Nesterov implementation always outperforms the other two algorithms. We note that with $h^2 = 0.1$ only 20 steps are sufficient for a converged solution, in contrast with the 100 steps reported in [13].

Using a square plate, see Fig. 10, we inspect the effect of the algorithm for the clamped boundary (represented in Fig. 10) and the simply-supported boundary for the two load cases represented in Fig. 10. This example is described in [15,36]. For simplicity, we here adopt the standard displacement-based H8 hexahedron element, which is known to lock in shear. However, thickness is here 5% of the plate width, which is favorable for our analysis. We test both linear and nonlinear versions of this problem.

One-quarter of the plate is adopted, with up to 80 × 80 × 1 elements, which produces much higher resolution than in Lund and Stegmann [36]. For the two load cases and the two boundary conditions, the fibers are shown in Fig. 11 for the uniform pressure and in Fig. 12 for the point load. Convergence of the strain energy is shown in Fig. 13. The heavy ball algorithm clearly outperforms the standard gradient approach and Nesterov's algorithm converges much faster, even with small values of h^2. In addition, lower values of the strain energy are reached, with slightly different solutions near the center of the plate.

For the clamped with uniform pressure, we assess the effect of nonlinearities in the fiber distribution. For that objective we use a Kirchhoff/Saint-Venant (orthotropic) strain energy density function, $\psi(\boldsymbol{u}, \mathbf{q}) = {}^1\!/\!_2\boldsymbol{E}^T \cdot \mathcal{C}_{\alpha_k} \cdot \boldsymbol{E}$ where $\boldsymbol{E}$ is the Green-Lagrange strain. In addition, only one *load-step*

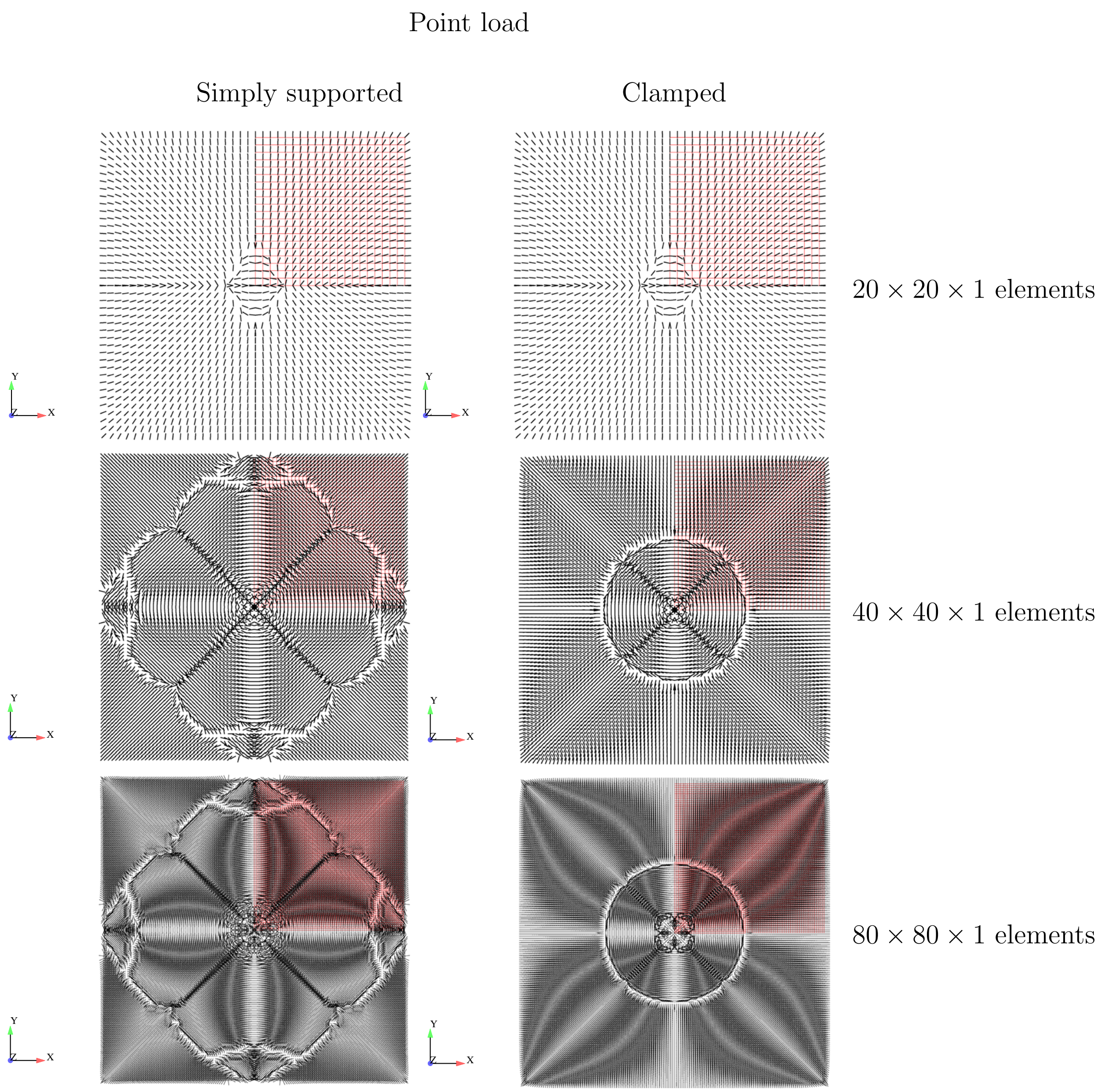
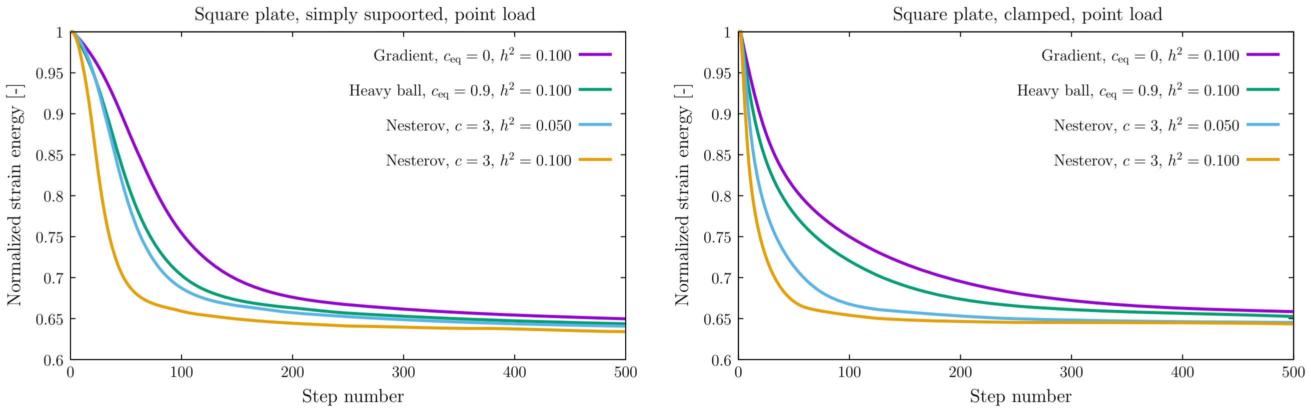


Fig. 12 Square plate under point load: orientation results as a function of mesh density



(a) Simply supported plate, point load.

(b) Clamped plate, point load.

Fig. 13 Strain energy evolution as a function of the *step* number. Plate under point load

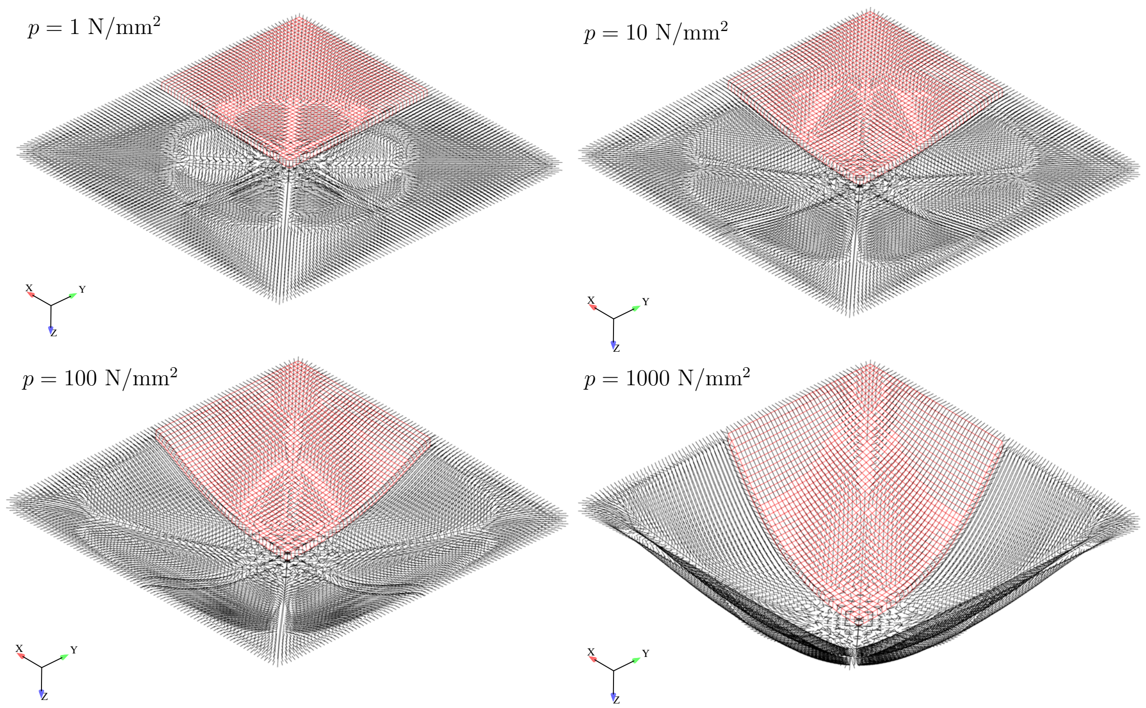


Fig. 14 Geometrically nonlinear fiber distribution. Unmagnified configurations

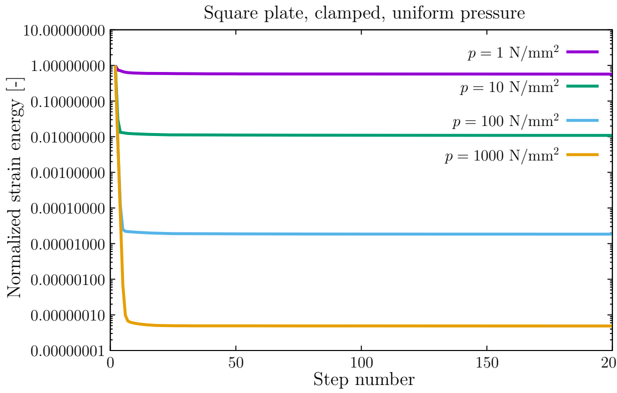


Fig. 15 Geometrically nonlinear case: Normalized strain energy evolution as a function of the optimization step number and the imposed pressure

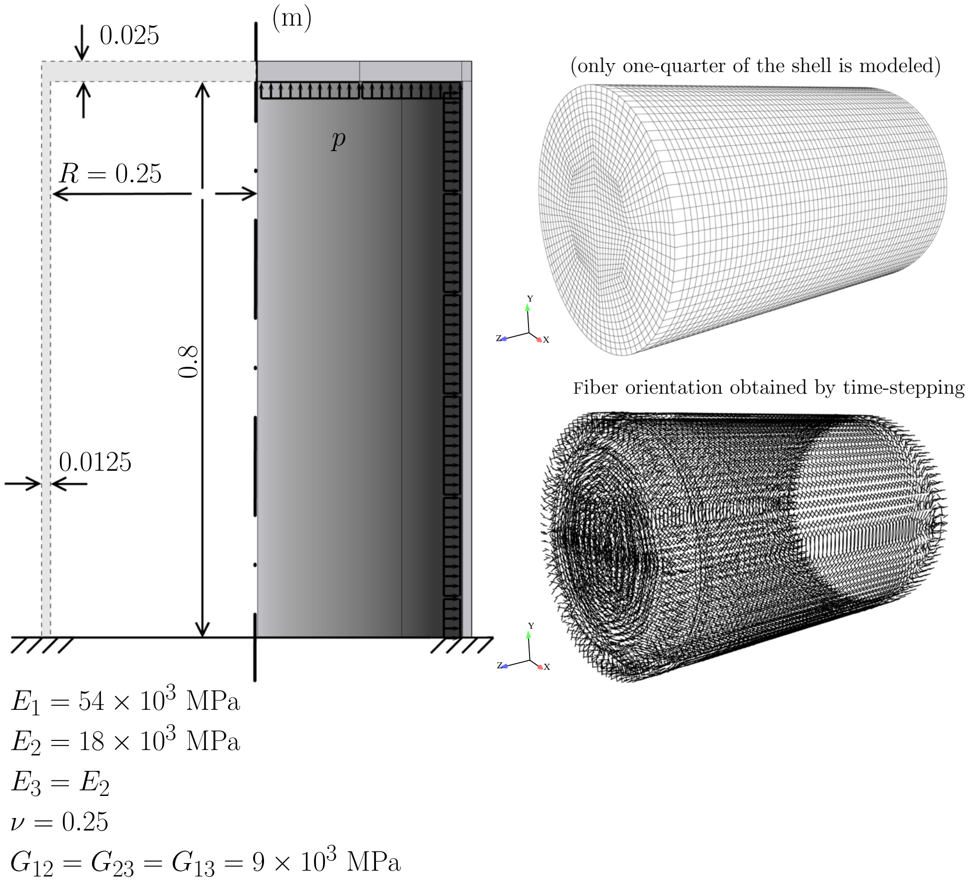
is employed. The configurations of fiber distributions using the deformed mesh are shown in Fig. 14. Upper and lower fiber orientations are distinct since membrane strains are now significant. We normalized the strain energy sensitivity to allow for changes in each state solution stage. This allows for very fast convergence, as depicted in Fig. 15. Higher values of pressure result in higher stiffening effect and stronger decrease in strain energy.

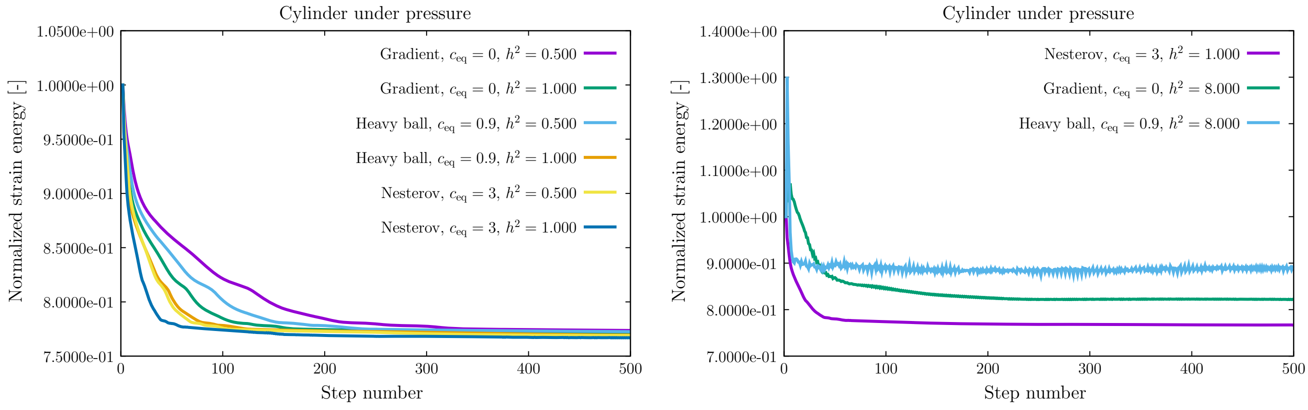
We now present a generalization to shells. With the goal of exhibiting the capability of our time-integration algorithm in this case, a pressure vessel is considered. Dimensions and constitutive properties are presented in Fig. 16. Tensor transformation is adopted to orient the directions to be tangent to the outer surface. Optimized fiber orientation can be observed in the same Figure. A concentric distribution of fibers can be observed in the shell end cap (*head*) and in the cylindrical surface a crossed pattern is obtained. Near the cap, a transition zone is observed where the fibers become longitudinally aligned. For this problem, two values of h^2 are adopted: $h^2 = 0.5, 1$. From the observation of Fig. 17a, we can conclude that the Nesterov's algorithm converges much faster, even with small values of h^2. Attempts to match the performance of Nesterov's algorithm by the gradient or the heavy ball algorithms with higher values of h^2 result in instability, see Fig. 17b.

6 Conclusions

In this work, we proposed a time-step integration based on an explicit Newmark method for the application of Nesterov's minimization algorithm in engineering applications. This allows the use of time-integration in a classical FE code to perform optimization. Stability analysis, benchmark test functions and careful assessment of all options were performed. Optimization of fiber orientations was carried out with this alternative method, including the geometrically nonlinear case. For that, a variant of DMO ([13,36]) design variable definition was used. A complete assessment was presented and the new method consistently outperformed the

Fig. 16 Cylinder under pressure: geometry, relevant elastic properties and boundary conditions


(m)
0.025
$R = 0.25$
p
0.8
0.0125
(only one-quarter of the shell is modeled)
Fiber orientation obtained by time-stepping
Y
Z
X
$E_1 = 54 \times 10^3$ MPa
$E_2 = 18 \times 10^3$ MPa
$E_3 = E_2$
$\nu = 0.25$
$G_{12} = G_{23} = G_{13} = 9 \times 10^3$ MPa



(a) Comparison between gradient method, heavy ball and Nesterov

(b) Attempt to match Nesterov descent performance with large values of h^2 in classical methods

Fig. 17 Strain energy evolution as a function of the *step* number for three methods (gradient, heavy ball and Nesterov). Cylinder under pressure

existing gradient, heavy ball and vanilla conjugate-gradient methods in classical benchmarks and converges faster than gradient and heavy ball methods in the fiber optimization application. Source code for the benchmark tests is available in Github [29].

Acknowledgements The first author acknowledges the support of FCT, through IDMEC, under LAETA, Project UIDB/50022/2020. The first author would like to thank Professor Leonel Fernandes at IST for the outstanding insight concerning time integration algorithms.

Appendix

See Table 4.

Table 4 First order amplification matrices (using Hughes terminology [24]) for four ODE explicit integrators

<table>
<tr><th>Algorithm</th><th>Amplification matrix $\boldsymbol{C}$</th></tr>
<tr><td>Explicit Newmark:</td><td>$\begin{bmatrix} 1-\frac{\widetilde{h}^2}{2} & \widetilde{h}-\widetilde{h}^2\xi \\ \frac{\widetilde{h}\left(\widetilde{h}^2-4\right)}{4\widetilde{h}\xi+4} & \frac{\left(\widetilde{h}{}^{-}2\right)\left(\widetilde{h}\xi-1\right)}{2\widetilde{h}\xi+2} \end{bmatrix}$</td></tr>
<tr><td>Explicit Newmark with predictor:</td><td>$\begin{bmatrix} \frac{2\widetilde{h}(\xi-\widetilde{h})+2}{-\widetilde{h}^2+\left(\widetilde{h}^2+2\right)\xi\widetilde{h}+2} & \frac{2\widetilde{h}\left(\widetilde{h}^2-\xi\widetilde{h}-1\right)\left(\widetilde{h}\xi-1\right)}{-\widetilde{h}^2+\left(\widetilde{h}^2+2\right)\xi\widetilde{h}+2} \\ \frac{\widetilde{h}\left(\widetilde{h}^2-1\right)}{-\widetilde{h}^2+\left(\widetilde{h}^2+2\right)\xi\widetilde{h}+2}-\frac{\widetilde{h}}{2\widetilde{h}\xi+2} & -\frac{2\left(\widetilde{h}^2-1\right)\left(\widetilde{h}^2-\xi\widetilde{h}-1\right)\left(\widetilde{h}\xi-1\right)}{\left(\widetilde{h}\xi+1\right)\left(-\widetilde{h}^2+\left(\widetilde{h}^2+2\right)\xi\widetilde{h}+2\right)} \end{bmatrix}$</td></tr>
<tr><td>Explicit asynchronous:</td><td>$\begin{bmatrix} 1-2\widetilde{h}\xi & -\widetilde{h} \\ \widetilde{h}-2\widetilde{h}^2\xi & 1-\widetilde{h}^2 \end{bmatrix}$</td></tr>
<tr><td>Explicit asynchronous with predictor:</td><td>$\left[\begin{array}{c|c} 2\xi\widetilde{h}^3-\widetilde{h}^2-2\xi\widetilde{h}+1 & -\widetilde{h} \\ 2\xi\widetilde{h}^4-\widetilde{h}^3-2\xi\widetilde{h}^2+\widetilde{h} & 1-\widetilde{h}^2 \end{array}\right]$</td></tr>
</table>

References

1. Nesterov Y (1983) A method of solving a convex programming problem with convergence rate $\bigcirc(1/k^2)$. Sov Math Dokl 27(2):372–376
2. Polyak BT (1964) Some methods of speeding up the convergence of iteration methods. USSR Comput Math Math Phys 4(5):1–17
3. Nesterov Y (2004) Introductory lectures on convex optimization. A basic course, Applied optimization. Kluwer Academic Publishers, Boston
4. Su W, Boyd S, Candès EJ (2016) A differential equation for modeling Nesterov's accelerated gradient method: theory and insights. J Mach Learn Res 17:1–43
5. Hestenes MR, Stiefel E (1952) Methods of conjugate gradients for solving linear systems. J Res Natl Bureau Stand 49:409–436
6. Karimi S, Vavasis SA (2016) A unified convergence bound for conjugate gradient and accelerated gradient
7. Drori Y, Taylor AB (2020) Efficient first-order methods for convex minimization: a constructive approach. Math Programm Ser A 184:183–220
8. Sutskever I, Martens J, Dahl G, Hinton G (2013) On the importance of initialization and momentum in deep learning. In: ICML'13 proceedings of the 30th international conference on machine learning, volume 28, pp III-1139–III-1147
9. Carlon AG, Dia BM, Espath L, Lopez RH, Tempone R (2020) Nesterov-aided stochastic gradient methods using Laplace approximation for Bayesian design optimization. Comput Method Appl Mech Eng 363:112909
10. Donoghue BO, Candès E (2015) Adaptive restart for accelerated gradient schemes. Found Comput Math 15:715–732
11. Schneider M (2017) An fft-based fast gradient method for elastic and inelastic unit cell homogenization problems. Comput Method Appl Mech Eng 315:846–866
12. Newmark NM (1959) A method of computation for structural dynamics. J Eng Mech Div 85(EM3):67–94
13. Stegmann J, Lund E (2005) Discrete material optimization of general composite shell structures. Int J Numer Methods Eng 62:2009–2027
14. Lund E (2018) Discrete material and thickness optimization of laminated composite structures including failure criteria. Strut Multidisc Optim 57:2357–2375
15. Hvejsel CF, Lund E, Stolpe M (2011) Optimization strategies for discrete multi-material stiffness optimization. Strut Multidisc Optim 44:149–163
16. Moré JJ, Garbow BS, Hillstrom KE (1981) Testing unconstrained optimization software. ACM Trans Math 7(1):17–41
17. Fazlyab M, Robey A, Hassani H, Morari M, Pappas G (2019) Efficient and accurate estimation of Lipschitz constants for deep neural networks. In: Wallach H, Larochelle H, Beygelzimer A, dAlché F, Fox E, Garnett R (eds) Advances in neural information processing systems. Curran Associates, Red Hook, pp 11427–11438
18. Areias P. Simplas. http://www.simplassoftware.com. Portuguese Software Association (ASSOFT) registry number 2281/D/17
19. Fiacco AV, McCormick GP (1968) Nonlinear programming: sequential unconstrained minimization techniques. Wiley, New York. Reprinted by SIAM Publications in 1990
20. Apostol TM (1967) Calculus, vol 1, 2nd edn. Wiley, New York, p 443
21. Belytschko T, Liu WK, Moran B (2000) Nonlinear finite elements for continua and structures. Wiley, New York
22. Meirovitch L (2001) Fundamentals of vibrations. Mechanical engineering series. McGraw-Hill International, New York, NY
23. Clough RW, Penzien J (2003) Dynamics of structures, 3rd edn. Computers & Structures Inc, Berkeley, CA
24. Hughes TJR (2000) The finite element method. Dover Publications. Reprint of Prentice-Hall edition 1987
25. Gilbert JC, Nocedal J (1992) Global convergence properties of conjugate gradient methods for optimization. SIAM J Optim 2(1):21–42
26. Nocedal J, Wright S (2006) Numerical optimization. Series operations research. Springer, Berlin
27. Dai Y, Yuan J, Yuan Y-X (2002) Modified two-point stepsize gradient methods for unconstrained optimization. Comput Optim Appl 22:103–109
28. Barzilai J, Borwein JM (1988) Two-point step size gradient methods. IMA J Numer Anal 8:141–148
29. Areias P (2020) Nesterov/Newmark optimizer at IST. https://github.com/PedroAreiasIST/NesterovNewmark
30. Gavrilovic M, Petrovic R, Siljak D (1963) Adjoint method in sensitivity analysis of optimal systems. J Frankl Inst Eng Appl Math 276(1):26
31. Byrne CL (2013) Alternating minimization as sequential unconstrained minimization: a survey. J Optim Theory Appl 156:554–566
32. Wriggers P (2008) Nonlinear finite element methods. Springer, Berlin
33. Wolfram Research, Inc., Mathematica, Version 9.0, Champaign, IL (2012)
34. Korelc J (2002) Multi-language and multi-environment generation of nonlinear finite element codes. Eng Comput 18(4):312–327
35. Sigmund O, Torquato S (1997) Design of materials with extreme thermal expansion using a three-phase topology optimization method. J Mech Phys Solids 45(6):1037–1067

36. Lund E, Stegmann J (2005) On structural optimization of composite shell structures using a discrete constitutive parametrization. Wind Energy 8:109–124
37. Polak E, Ribière G (1969) Note sur la convergence de méthodes de directions conjuguées. Rev Française Informat Recherche Opérationelle 1(3):35–43