Accelerated projected gradient method with adaptive step size for compliance minimization problem

Akatsuki Nishioka\textsuperscript{1}\textsuperscript{*} and Yoshihiro Kanno\textsuperscript{1,2}

\textsuperscript{1} Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
\textsuperscript{2} Mathematics and Informatics Center, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

\textsuperscript{*}Corresponding author: akatsuki.nishioka@mist.i.u-tokyo.ac.jp

Received April 11, 2021, Accepted April 29, 2021

Abstract

We present an accelerated projected gradient method for solving a topology optimization problem. Specifically, we consider a compliance minimization problem of continua. The proposed method has guaranteed convergence to a stationary point and is easy to implement. By numerical experiments, we show that the computational cost of the proposed method is lower than that of the optimality criteria method.

Keywords topology optimization, compliance minimization, accelerated gradient method, projected gradient method

Research Activity Group Mathematical Design

1. Introduction

Topology optimization is a method to find the optimal design of a structure by mathematical optimization. At each iteration, a conventional topology optimization algorithm requires the finite element analysis (FEA) to calculate the value and the gradient (design sensitivity) of the objective function. Since most of the computational time at each iteration is spent on the FEA, many efforts have been made to reduce the computational time of the FEA, such as parallel computing and efficient iterative solvers utilizing preconditioning and subspace recycling [1].

In our research, we do not aim to reduce the computational cost per iteration. Rather, by using a fast converging optimization algorithm, we try to reduce the number of iterations, which makes the total computational cost smaller. One such fast converging algorithm is a second-order method, e.g. the Newton method, which uses the second-order derivative of the objective function. In a second-order method, however, the computational cost per iteration becomes huge in case of a large-scale optimization problem. Therefore, it is impractical to use a second-order method, since the number of design variables can be millions or more in topology optimization. This motivates us to use an accelerated gradient method which converges fast while using only the first-order derivative. Accelerated gradient methods are intensively studied in the machine learning community, since, as well as topology optimization, large-scale optimization problems often appear in machine learning.

Accelerated gradient methods are originally designed for unconstrained convex optimization, e.g. Nesterov’s accelerated gradient method [2]. Recently, the acceleration of the proximal gradient method [3], which is a generalization of the projected gradient method, has been proposed. Also, the accelerated gradient methods for nonconvex optimization problems [4, 5] have been developed. Thus, the applicability of accelerated gradient methods has been extended.

Recently, accelerated proximal gradient methods have been successfully applied to optimization problems in computational plasticity [6, 7]. In contrast, very few applications to topology optimization are found in the literature. For example, Li and Zhang [8] applied the accelerated mirror descent method to a robust topology optimization problem. They reduce the computational cost of the robust topology optimization problem by the stochastic gradient method.

In this paper, we consider a compliance minimization problem and develop an accelerated projected gradient method based on [5]. The proposed method is guaranteed to converge to a stationary point. Such convergence property is essential to check the convergence of the algorithm properly. Besides, the proposed method is easy to implement. By numerical experiments, we show that it converges fast compared to the optimality criteria method, a common heuristic optimization algorithm used in topology optimization, even though the computational cost per iteration is almost the same. The proposed method terminates with a smaller number of iterations, therefore it reduces the computational time.

2. Problem setting

The compliance minimization problem is a representative topology optimization problem that maximizes the stiffness of a structure subjected to static loads; see [9,10] for details. In this paper, we consider the same setting as [9].

First, we divide the design domain into $n$ identical
square finite elements with unit volume. We use the SIMP method. Namely, the design variable is the density vector \( x \in \mathbb{R}^n \), the \( \ell \)th component \( x_{\ell} \) of which denotes the density of the \( \ell \)th finite element, and Young’s modulus is defined as

\[
E_c(x_{\ell}) = E_{\text{min}} + x_{\ell}^p (E_0 - E_{\text{min}}),
\]

where \( p > 1 \) is the penalty parameter and \( E_0 \gg E_{\text{min}} > 0 \) are constants. In addition, we use the density filter to prevent mesh dependency. The density filter is a linear operator on the density vector \( x \), thus, by using a constant matrix \( H \in \mathbb{R}^{n \times n} \), the filtered density vector is written as

\[
\tilde{x} = H x.
\]

The compliance minimization problem is defined as follows:

\[
\begin{aligned}
\text{Minimize} & \quad p^T u \\
\text{subject to} & \quad K(\tilde{x})u = p, \\
& \quad \tilde{x} = H x, \\
& \quad 1^T x = V_0, \\
& \quad 0 \leq x \leq 1.
\end{aligned}
\]

Here, \( p \in \mathbb{R}^n \) is the constant load vector, \( u \) is the global nodal displacement vector, \( m \) is the number of degrees of freedom of the nodal displacements, \( K(x) \) is the global stiffness matrix, \( 0 \) and \( 1 \) are the vectors with all components equal to 0 and 1, respectively, and \( V_0 \) is the upper limit of the volume.

Problem (3) can be rewritten as the following optimization problem with a nonconvex objective function and a closed convex feasible set:

\[
\begin{aligned}
\text{Minimize} & \quad f(x) := p^T K(Hx)^{-1} p \\
\text{subject to} & \quad 1^T x = V_0, \\
& \quad 0 \leq x \leq 1.
\end{aligned}
\]

In the following, we present an algorithm to solve problem (4).

### 3. Projected gradient method

The projected gradient method is a classical optimization algorithm for an optimization problem with the closed convex feasible set \( S \subseteq \mathbb{R}^n \) (such as problem (4)). In the projected gradient method, we find a solution by repeating the following formula starting at the initial point \( x^0 \in S \):

\[
x^{k+1} = \Pi_S(x^k - \alpha_k \nabla f(x^k)).
\]

Here, \( \alpha_k \) is the step size, and \( \Pi_S(v) \in S \) is the projection of a given vector \( v \in \mathbb{R}^n \) onto \( S \) defined as follows:

\[
\Pi_S(v) = \arg \min_{w \in S} \| w - v \|^2.
\]

That is, \( \Pi_S(v) \) is the closest point in \( S \) from \( v \). The projection must be easily calculated to use the projected gradient method in practice.

We show that, in the compliance minimization problem (4), the projection is easily calculated. Observe that the projection \( \Pi_S(v) \) is equal to the optimal solution of the following subproblem:

\[
\begin{aligned}
\text{Minimize} & \quad \frac{1}{2} \| w - v \|^2 \\
\text{subject to} & \quad 1^T w = V_0, \\
& \quad 0 \leq w \leq 1.
\end{aligned}
\]

This subproblem is a convex optimization problem, and has a unique optimal solution. The KKT condition is the necessary and sufficient condition for optimality. Let \( \lambda, \nu \in \mathbb{R}^n \) and \( \mu \in \mathbb{R} \) be the Lagrange multipliers. The Lagrangian of problem (7) is

\[
L = \frac{1}{2} \| w - v \|^2 - \lambda^T w + \nu^T (w - 1) + \mu (1^T w - V_0),
\]

and the KKT condition is written as follows:

\[
\begin{aligned}
\nabla_w L &= w - v - \lambda + \nu + \mu 1 = 0, \\
1^T w &= V_0, \\
0 \leq w &\leq 1, \\
\lambda &\geq 0, \\
\nu &\geq 0, \\
\lambda^T w &= 0, \\
\nu^T (w - 1) &= 0.
\end{aligned}
\]

We can find the solution satisfying (9) as follows. First, for a given vector \( v \), define \( g_c(\cdot; v) : \mathbb{R} \rightarrow \mathbb{R}^n \) by

\[
g_c(\mu; v) = \begin{cases} 
0 & (v_e - \mu < 0), \\
(\mu - v_e) & (v_e - \mu \geq 0).
\end{cases}
\]

Then, \( 1^T g_c(\mu; v) \) is a monotonically decreasing piecewise linear function of \( \mu \), and there exits a \( \mu^* \) in \( \{v_1, \ldots, v_n\} \) such that \( 1^T g_c(\mu^*; v) = V_0 \) holds. Thus, if we choose such \( \mu^* \), then \( g_c(\mu^*; v) \) satisfies the KKT condition (9), and hence we have

\[
\Pi_S(v) = g_c(\mu^*; v).
\]

Note that the other Lagrange multipliers \( \lambda \) and \( \nu \) are

\[
\lambda_e = \begin{cases} 
-(v_e - \mu^*) & (v_e - \mu^* \leq 0), \\
0 & (\text{otherwise}).
\end{cases}
\]

(12)

\[
\nu_e = \begin{cases} 
(v_e - \mu^* - 1) & (v_e - \mu^* \geq 1), \\
0 & (\text{otherwise}).
\end{cases}
\]

(13)

In the algorithm, we calculate \( \mu^* \), the root of \( 1^T g_c(\mu; v) - V_0 = 0 \), by the bisection method. We do not need to calculate \( \lambda \) and \( \nu \) explicitly, since \( \Pi_S(v) \) is computed without \( \lambda \) and \( \nu \) as seen in (10) and (11). The computational cost of the bisection algorithm is much lower than that of the FEA. Note that the method described above is an extension of computation of the projection onto simplex [11].

Now, we can apply the projected gradient method to problem (4), since we can easily calculate the projection. However, the projected gradient method converges slowly. Hence, in the next section, we consider acceleration of the projected gradient method.
4. Proposed method

4.1 Accelerated projected gradient method

There are many different kinds of the accelerated projected gradient methods, but some of them are not suited for topology optimization. For example, the method in Li and Lin [4] uses the gradients at two different points at each iteration of the algorithm. This means we need to perform twice FEA at each iteration, which is computationally expensive. Besides, if an accelerated projected gradient method based on [5] which requires only one FEA to update the design variables at each iteration. The proposed method is guaranteed to converge to a stationary point even if the objective function is nonconvex, and it is theoretically guaranteed that it converges faster than the original projected gradient method.

In the proposed method, we set the initial point \( x_0 = y^0 \in S \) and calculate \( x^k \) \((k = 1, 2, \ldots)\) using the following update scheme:

\[
\begin{align*}
  z^k &= \left(1 - \frac{2}{k + 1}\right) x^{k-1} + \frac{2}{k + 1} y^{k-1}, \\
  y^k &= \Pi_S \left( y^{k-1} - \frac{k}{4L} \nabla f(z^k) \right), \\
  x^k &= \Pi_S \left( z^k - \frac{1}{2L} \nabla f(z^k) \right),
\end{align*}
\]

(14)

where \( y^k \) and \( z^k \) are the auxiliary variables to obtain \( x^k \), and \( L \) is the Lipschitz constant of the gradient of the objective function. The update scheme (14) is computationally tractable and easy to implement, since it consists of additions of vectors, scalar multiplications, and projections.

4.2 Stepsize policy

We say that \( f: \mathbb{R}^n \to \mathbb{R} \) is \( L \)-smooth if there exists a positive constant \( L \) such that the inequality

\[
\| \nabla f(x) - \nabla f(y) \| \leq L \| x - y \| \quad (\forall x, y \in \mathbb{R}^n)
\]

(15)

holds. Namely, \( L \) is the Lipschitz constant of \( \nabla f \). The proposed method assumes \( L \)-smoothness of the objective function. In our problem, the objective function \( f \) is \( L \)-smooth, since \( f \) is a rational function and twice differentiable on \([0, 1]^n\). See [12] for more details on \( L \)-smoothness.

In the update scheme (14), the inverse of \( L \) is used as the step size. However, it is difficult to obtain \( L \) analytically, and hence we need to estimate \( L \). If an estimate of \( L \) is sufficiently large, (15) is satisfied, but too large value means small step size and leads to slow convergence.

When \( L \) is difficult to obtain analytically, the backtracking is often used [12]. In the backtracking procedure, we start with sufficiently small \( L \) and gradually increase \( L \) until a certain condition is satisfied. Using the backtracking, convergence to a stationary point is still guaranteed. However, the termination criterion of the backtracking procedure requires calculating the objective value. This means we need to perform the FEA many times to decide the step size, which is computationally expensive. Besides, if \( L \) becomes too large, the convergence becomes slow.

Therefore, based on (15), we propose that we use

\[
L_k = \frac{\| \nabla f(z^k) - \nabla f(z^{k-1}) \|}{\| z^k - z^{k-1} \|} \quad (k = 2, 3, \ldots)
\]

(16)

instead of \( L \) to calculate \( y^k \) and \( z^k \) in (14). Here \( L_k \) is sufficiently large constant. Note that \( L_k \) is the lower bound of \( L \), and varies at every iteration. Although this method is heuristic, we can automatically adjust the step size regardless of the problem setting. By numerical examples, we show that the proposed method does not cause any numerical instabilities, and converges fast.

5. Numerical results

Consider compliance minimization problems of the MBB beam and the cantilever shown in Fig. 1. The parameters are as follows: the penalty parameter of the SIMP method \( p = 3 \), the filter radius \( r = 0.04n_b \) \((n_b \) is the number of elements in the horizontal axis), the upper limit of the volume \( V_0 = 0.5n \) in the MBB beam and \( V_0 = 0.4n \) in the cantilever, \( x^0 = (V_0/n^1)I \). The tests have been run on a desktop computer equipped with an Intel(R) Core i9 3.6GHz CPU, 128GB of RAM, and MATLAB R2020b. We compare the proposed method (the accelerated projected gradient method, APG) with the optimality criteria method (OC). The stopping criteria of the algorithms are \( \| x^k - x^{k-1} \|_\infty < \epsilon \) in the OC [9] and \( \| x^k - z^k \|_\infty < \epsilon \) in the APG [5], respectively. We set \( \epsilon = 10^{-3} \). Note that, in the APG, \( \| x^k - z^k \|_\infty = 0 \) is the necessary and sufficient condition that \( x^k \) is the stationary point (KKT point), thus we can say that the algorithm has converged when \( \| x^k - z^k \|_\infty \) became sufficiently small. In contrast, the OC does not have such convergence property since it is heuristic, thus it is difficult to check the convergence of the algorithm properly.

The number of iterations “iter.”, computational time per iteration in seconds “time” and the objective value (compliance) \( f(x) \) for various numbers of elements in the MBB beam and the cantilever are shown in Table 1. Also, the total computational time is shown in Fig. 2.

It is observed from Table 1 that, although computational time per iteration is not much different in both methods, the number of iterations and the compliance are smaller in the APG in all cases. Fig. 2 shows that the APG terminates in a shorter time, and the gap between the OC and the APG gets larger as the number of elements increases.

Also, as shown in Fig. 3, the obtained solutions look the same in the cantilever example but slightly differ-
Table 1. Computational results.

|                | iter. | \(t_n\) (s) | \(f(x)\) |
|----------------|-------|--------------|----------|
| MBB beam       |       |              |          |
| 90 \times 30   | APG   | 448          | 1.34 \times 10^{-2} | 231.65  |
| ( = 2700)      | OC    | 1730         | 1.39 \times 10^{-2} | 233.83  |
| 210 \times 70  | APG   | 1967         | 8.27 \times 10^{-2} | 234.83  |
| ( = 14700)     | OC    | 5034         | 8.52 \times 10^{-2} | 236.63  |
| 300 \times 100 | APG   | 2414         | 1.66 \times 10^{-1} | 236.28  |
| ( = 29400)     | OC    | 10854        | 1.72 \times 10^{-1} | 238.82  |
| Cantilever     |       |              |          |
| 90 \times 60   | APG   | 632          | 2.72 \times 10^{-2} | 52.53   |
| ( = 5400)      | OC    | 1088         | 2.72 \times 10^{-2} | 52.87   |
| 150 \times 100 | APG   | 657          | 7.98 \times 10^{-2} | 52.42   |
| ( = 15000)     | OC    | 4132         | 7.87 \times 10^{-2} | 53.68   |
| 210 \times 140 | APG   | 1089         | 1.76 \times 10^{-1} | 57.60   |
| ( = 29400)     | OC    | 7538         | 1.80 \times 10^{-1} | 57.71   |

Fig. 2. Computational time.

Fig. 3. Solutions for (a) MBB beam (210 \times 70) and (b) Cantilever (150 \times 100) obtained by the APG (left) and the OC (right).

6. Conclusion

In this paper, we have developed an accelerated projected gradient method for solving a compliance mini-

References

[1] J. D. Deaton and R. V. Grandhi, A survey of structural and multidisciplinary continuum topology optimization: post 2000, Struct. Multidiscipl. Optim., 49 (2014), 1–38.
[2] Y. Nesterov, A method for solving the convex programming problem with convergence rate \(O(1/k^2)\), Soviet Math. Dokl., 27 (1983), 372–376.
[3] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci., 2 (2009), 183–202.
[4] H. Li and Z. Lin, Accelerated proximal gradient methods for nonconvex programming, in: Proc. of NIPS 2015.
[5] S. Ghadimi and G. Lan, Accelerated gradient methods for nonconvex nonlinear and stochastic programming, Math. Program., 156 (2016), 59–99.
[6] W. Shimizu and Y. Kanno, Accelerated proximal gradient method for elastoplastic analysis with von Mises yield criterion, Jpn. J. Ind. Appl. Math., 35 (2018), 1–32.
[7] W. Shimizu and Y. Kanno, A note on accelerated proximal gradient method for elastoplastic analysis with Tresca yield criterion, J. Oper. Res. Soc. Jpn., 63 (2020), 78–92.
[8] W. Li and X. S. Zhang, Momentum-based accelerated mirror descent stochastic approximation for robust topology optimization under stochastic loads, Int. J. Numer. Methods Eng., 108 (2011), 1–27.
[9] W. Ankersen, A. Clausen, M. Schevenels, B. S. Lazarov and O. Sigmund, Efficient topology optimization in MATLAB using 88 lines of code, Struct. Multidiscipl. Optim., 43 (2011), 1–16.
[10] M. P. Bendsoe and O. Sigmund, Topology Optimization: Theory, Methods and Applications, 2nd edn., Springer-Verlag, Berlin, 2004.
[11] N. Parikh and S. Boyd, Proximal algorithms, Found. Trends Optim., 1 (2013), 123–231.
[12] A. Beck, First-Order Methods in Optimization, SIAM, Philadelphia, 2017.