Structural Engineering and Mechanics, *Vol. 55, No. 5 (2015) 901-911*
DOI: http://dx.doi.org/10.12989/sem.2015.55.5.901

Generation of OC and MMA topology optimizer by using accelerating design variables

Dongkyu Lee[1a], Nguyen Hong Chan[1b] and Soomi Shin[*2]

[1]Department of Architectural Engineering, Sejong University, Seoul, 143-747, Korea
[2]Research Institute of Industrial Technology, Pusan National University, Busan, 609-735, Korea

(Received July 9, 2014, Revised February 23, 2015, Accepted February 24, 2015)

Abstract. The goal of this study is to investigate computational convergence of optimal solutions, with respect to optimality criteria (OC) method and methods of moving asymptotes (MMA) as optimization model for non-linear programming of material topology optimization using an acceleration method that makes design variables rapidly move toward almost 0 and 1 values. 99 line topology optimization MATLAB code uses loop vectorization and memory pre-allocation as properly exploiting the strengths of MATLAB and moves portions of code out of the optimization loop so that they are only executed once as restructuring the program. Numerical examples of a simple beam under a lateral load and a given material density limitation provide merits and demerits of the present OC and MMA for 99 line topology optimization code of continuous material topology optimization design.

Keywords: acceleration method; OC; MMA; MATLAB; topology optimization; convergence

1. Introduction

Material topology optimization (Bendsoe and Kikuchi 1988, Bendsoe 1995, Fu and Zhang 2014; Lee *et al.* 2014a, 2014b) is now a well-established field. And the Sigmund's 99 line code (Sigmund 2001) demonstrated that basic topology optimization code to intend for educational purposes and serves as an introductory example to topology optimization for students and newcomers to the field. The 99 line code used MATLAB programming language that allows for the solution of numerous scientific problems with a minimum of coding effort.

From the 99 line topology optimization code, there are many MATLAB code were provided to extend and apply for more fields, as the 88 lines of code (Andreassen *et al.* 2011), a 199 line MATLAB code for Pareto-Optimal tracing in topology optimization (Suresh 2010), a discrete lever-set topology optimization code written in Matlab (Challis 2010). A lot of researches were applied and investigated in many science fields.

MATLAB is further from optimally utilized than the other programming languages in comparison, such as the extensive use of *for* and *while* loops, and MATLAB is relatively tolerant

[*]Corresponding author, Ph.D., E-mail: shinsumi82@pusan.ac.kr
[a]Assistant Professor
[b]Master Student

http://www.techno-press.org/?journal=sem&subpage=8 ISSN: 1225-4568 (Print), 1598-6217 (Online)

towards bad programming practices, such as the use of dynamically growing variable arrays. The use of MATLAB, with its accessible syntax, excellent debugging tools, and extensive graphics handling opportunities, allow the user to focus on the physical and mathematical background of the optimization problem without being distracted by technical implementation issues.

The 99 line topology optimization code used loop vectorization and memory pre-allocation as properly exploiting the strengths of MATLAB and moving portions of code out of the optimization loop so that they are only executed once as restructuring the program. For simplicity, the 99 line code solved the optimization problem with using a standard OC method (Lee *et al.* 2012, Yi *et al.* 2013).

The Method of Moving Asymptotes (MMA) (Svanberg, 1987) is based on a special type of convex approximation. MMA is an iterative method. In each interaction, a convex subproblem, which approximates the original problem, is generated and solved. An important role in the generation of these subproblems is played by a set of parameters which influence the "curvature"of the approximations, and also act as "asymptotes" for the subproblems. By moving these asymptotes between each interaction, the convergence of the overall process can be stabilized (Zillober 1993).

The aim of this paper is to present the application of MMA to the 99 line code by using a specific method of accelerating design variables for topology optimization engineering problems to improve convergence of MMA including OC. The revised MATLAB code was extended and given and optimal results were investigated. The comparison between 99 lines original code and applied MMA code was discussed and the convergence of optimal solutions was on advance comments.

Topology design problems may be considered as related formulations under a common SIMP topology optimization umbrella by the penalty method, and more importantly, this general framework may be easily applied to new design problems (Bruns 2007). Our contribution is to illustrate the present acceleration method by which can be replaced the penalty method under a common topology optimization field.

In the following Sections, the brief descriptions of the topology optimization problem are given in Section 2 and the details of MMA also. MATLAB implementation is presented in Section 3. From the MATLAB code running, the results are described in Section 4 and the comparison of two methods also. Finally, conclusions are presented in Section 5.

2. Topology optimization problem and computational optimizer algorithm

2.1 Formulations of material topology optimization

Topology optimization problem presented in this study is to determine the layout of material of specified volume in a domain that maximizes stiffness for a given set of loads and boundary conditions.

A topology optimization problem can be written as

$$\min f(x)(x \in R^n),\ s.t.\ \ h_j \leq 0,\ \ j = 1 \dots M,\ \ x \in X \tag{1}$$

where

$$X := \left\{x \middle| \underline{x_i} \leq x_i \leq \overline{x_i}, i = 1 \dots n\right\} \tag{2}$$

The functions f and h_j (j =1…M) are assumed to be continuously differentiable and the feasible region is assumed to be non-empty. x_i is design variables of material density. $\underline{x_i}$ and $\bar{x}_i$ are, respectively, lower and upper bound values of design variables.

A 2D linear elastostatic problem (Lee and Shin 2014c, 2015) is considered in this study. The problem is associated with a two-phase material topology optimization of continuous structures distributing a given amount of material into void and solid regions. The objective function can be written as a minimal strain energy form as

$$f = \min \left\{ \frac{1}{2} \int_{\Omega_X} \boldsymbol{\varepsilon}^{\mathrm{T}} \boldsymbol{C\varepsilon} \;\; d\Omega_X \right\} \tag{3}$$

where $\boldsymbol{C}$ is an isotropic elastic material tensor in a plane stress state, which is a function of Young's modulus E and Poisson's ratio v, and ε denotes the strain. The superscript T denotes the transpose of the matrix or vector.

The inequality constraint is a limit of relative design parameters $0 \leq \Phi = \Phi^{real}/\Phi^{bulk} \leq 1$. The equilibrium equation is an essential equality constraint. Another equality constraint is an assumed material volume fraction V_0 which has to be placed in the design domain at every step in the optimization process. It is defined as

$$\int_{\Omega_S} d\Omega_X - V_0 = 0 \tag{4}$$

2.2 Formulations of computational optimizers of MMA and OC

Svanberg (1987) introduced a well-established general approach for setting such a problem which generates and solves a sequence of explicit sub-problems according to the following iterative scheme:

Step 0. Choose a starting point $x^{(0)}$, and let the iteration index k=0.

Step I. Give an iteration point $x^{(k)}$, calculate $f_i(x^{(k)})$ and the gradients $\triangledown f_i(x^{(k)})$ for i=0, 1, .., m.

Step II. Generate a sub-problem $P^{(k)}$ by replacing, in P, the function f_i by approximating explicit functions $f_i^{(k)}$, based on the calculations from step I.

Step III. Solve $P^{(k)}$ and let the optimal solution of this sub-problem be the next iteration point $x^{(k+1)}$. Let k=k+1 and go to step I.

The Method of Moving Asymptotes (MMA) (Svanberg 1987) proposed a linearization with respect to substituted variables $\frac{1}{U_i - x_i}$ and $\frac{1}{x_i - L_i}$ respectively, where U_i and L_i are some chosen parameters.

Given the iteration point $x^{(k)}$ (at iteration k), values of the parameters $L_i^{(k)}$ and $U_i^{(k)}$ are chosen, for j=1,…,n, such that

$$L_i^{(k)} < x_i^{(k)} < U_i^{(k)} \tag{5}$$

Then, for each i=0,1,…,m, $f_i^{(k)}$ is defined by

$$f_i^{(k)}(x) = \mathrm{r}_i^{(k)} + \sum_{j=1}^{n} \left(\frac{p_{ij}^{(k)}}{U_j^{(k)} - x_j} + \frac{q_{ij}^{(k)}}{x_j - L_j^{(k)}} \right) \tag{6}$$

In MMA, the values $L_i^{(k)}$ and $U_i^{(k)}$ are always finite. With the approximating functions $f_i^{(k)}$ is defined, the following sub-problem, called $P^{(k)}$, is obtained:

$P^{(k)}$: minimize

$$\sum_{j=1}^{n} \left(\frac{p_{oj}^{(k)}}{U_j^{(k)} - x_j} + \frac{q_{oj}^{(k)}}{x_j - L_j^{(k)}} \right) + r_o^{(k)} \tag{7}$$

Subject to

$$\sum_{j=1}^{n} \left(\frac{p_{ij}^{(k)}}{U_j^{(k)} - x_j} + \frac{q_{ij}^{(k)}}{x_j - L_j^{(k)}} \right) + r_i^{(k)} \le \hat{f}_\iota, \quad for\ i = 1, \dots, m \tag{8}$$

and

$$max\left\{\underline{x}_j, \alpha_j^{(k)}\right\} \le x_j \le min\left\{\overline{x}_j, \beta_j^{(k)}\right\}, for\ j = 1, \dots, n \tag{9}$$

Here, the parameter $\alpha_j^{(k)}$ and $\beta_j^{(k)}$ are ‘move limits’ which are probably not very crucial. However, to avoid the possibility of any unexpected ‘division by zero’ while solving the sub-problem, $\alpha_j^{(k)}$ and $\beta_j^{(k)}$ should at least be chosen such that

$$L_j^{(k)} < \alpha_j^{(k)} < x_j^{(k)} < \beta_j^{(k)} < U_j^{(k)} \tag{10}$$

MMA use the flexibility of the asymptotes which influence the curvature of the approximations. It is possible to omit the concavity assumption. However, the resulting restriction on the choice of the asymptotes leads to a very slow numerical convergence of the algorithm, so that the theoretical improvement does not result in a more efficient algorithm.

An OC (optimality criteria) method (Rozvany 1989, Patnaik *et al.* 1995, Bendsoe 1995) of gradient-based concepts is used for the optimization method because it can reduce the computational cost associated with having many design variables. The design parameters can be updating using the Lagrangian multiplier λ, which can be solved by a bisection algorithm.

$$\frac{k(\rho_e)^{k-1}(\boldsymbol{u}_e)^T \boldsymbol{K}_e \boldsymbol{u}_e}{\lambda V_e} = \frac{-\dfrac{\partial f}{\partial \rho_e}}{\lambda V_e} \tag{11}$$

and

$$\rho_e^{\,i+1} = \left(\frac{-\dfrac{\partial f}{\partial \rho_e}}{\lambda V_e} \right)^{i} \rho_e^{\,i} \tag{12}$$

A moving limit l_m is introduced to avoid the situation in which ρ changes significantly in one iteration. By considering a heuristic updating scheme [60], as follows, the design parameters can now be updated.

$$\rho_e^{i+1} = \begin{cases} \max\left(\rho_{min}, \rho_e^i - l_m\right) & if\ \rho_e^i\left(\left(\dfrac{-\dfrac{\partial f}{\partial \rho_e}}{\lambda V_e}\right)_e^i\right)^{d_c} \le \max\left(\rho_{min}, \rho_e^i - l_m\right) \\ \rho_e^i\left(\left(\dfrac{-\dfrac{\partial f}{\partial \rho_e}}{\lambda V_e}\right)_e^i\right)^{d_c} & if\ \max\left(\rho_{min}, \rho_e^i - l_m\right) < \rho_e^i\left(\left(\dfrac{-\dfrac{\partial f}{\partial \rho_e}}{\lambda V_e}\right)_e^i\right)^{d_c} < \min\left(1, \rho_e^i + l_m\right) \\ \min\left(1, \rho_e^i + l_m\right) & if\ \rho_e^i\left(\left(\dfrac{-\dfrac{\partial f}{\partial \rho_e}}{\lambda V_e}\right)_e^i\right)^{d_c} \ge \min\left(1, \rho_e^i + l_m\right) \end{cases} \tag{13}$$

where d_c is a numerical damping coefficient, usually 0.5 intended to stabilize the iteration.

3. The method of accelerating design variables for computational optimizers

In order to overcome the problem, a strategy accelerating design variable values (ADV) (Lee *et al.* 2007) is treated in this paper and is based on convex and concave of a curve of regularized Heaviside function. In spite of fast convergence through ADV, fundamentally the ADV approach may take ill-posed problems such as numerical singularities by numerical error or perturbations, since design variable values move fast toward 0 (void) or 1 (solid) and especially take overestimated approximations toward 0. For the purpose of well-posedness of optimal solution, a regularization of the procedure of replacing the given ill-posed problem with a better-posed counterpart is used in this study.

The fast changes of updated design variable values from the nature of moved regularized Heaviside function, i.e. MRHF, as show in Fig. 1.

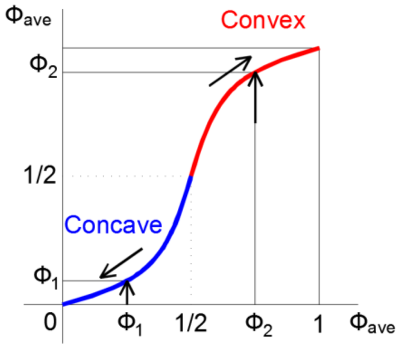


Fig. 1 Accelerating design variables (ADV) toward 0 (void) and 1 (solid) through moved regularized Heaviside function (MRHF) with interface thickness χ=0.5

In case of a design variable $\boldsymbol{\phi_1}$ which is updated more than the value 0.5, it is assumed that final design variable value is not $\boldsymbol{\phi_1}$ but $\boldsymbol{\phi_2}$=MRHF($\boldsymbol{\phi_1}$) since the convex of MRHF is used and the limit box of *x, y* coordinated in MRHF is a square. From $\boldsymbol{\phi_1} < \boldsymbol{\phi_2}$ it is clear that the original design variable $\boldsymbol{\phi_1}$ makes toward the value 1.0. In case of a updated design variable $\boldsymbol{\phi_3}$ which is less than the value 0.5, it is assumed that final design variable is $\boldsymbol{\phi_4}$ and it can be seen that the original design variable makes fast toward the value 0.

Various types of the moved regularized Heaviside function (MRHF) can be used as follows

$$MRHF_1(\emptyset_{ave}) = \frac{3}{4}\left[\frac{(\emptyset_{ave}-0.5)}{\chi} - \frac{1}{3}\left(\frac{\emptyset_{ave}-0.5}{\chi}\right)^3\right] + \frac{1}{2}, \quad 0 \le \emptyset_{ave} \le 1 \tag{14}$$

$$MRHF_2(\emptyset_{ave}) = \frac{1}{2} + \frac{2}{\pi} arctan\left(\frac{\emptyset_{ave}-0.5}{\chi}\right), \quad 0 \le \emptyset_{ave} \le 1 \tag{15}$$

$$MRHF_3(\emptyset_{ave}) = \frac{1}{2}\left[1 + \frac{(\emptyset_{ave}-0.5)}{\chi} + \frac{1}{\pi} sin\left(\frac{\pi(\emptyset_{ave}-0.5)}{\chi}\right)\right], 0 \le \emptyset_{ave} \le 1 \tag{16}$$

$$MRHF_4(\emptyset_{ave}) = \frac{1}{2}\left[1 + sin\left(\frac{\pi(\emptyset_{ave}-0.5)}{2\chi}\right)\right], \quad 0 \le \emptyset_{ave} \le 1 \tag{17}$$

where ϕ_{ave} denotes element densities which are averaged by nodal information. χ is a parameter describing interface thickness. In numerical calculations, the interface thickness should be fixed, and set proportional to the spatial mesh size Δh, i.e., χ=$m\Delta h$, here m is a constants coefficient, and χ=0.5.

4. Numerical examples and discussion

Following Sigmund (2001), several assumptions are made to simplify the Matlab code. The main program is call from the Matlab prompt by the line.

topmma (nelx,nely,volfrac,penal,rmin) where *nelx* and *nely* are the number of elements in the horizontal and vertical directions, respectively, *volfrac* is the volume fraction, *penal* is the penalization power and *rmin* is the filter size (divided by element size).

The example on MMB beam is tested and the result will be printed as demonstration for the efficiency of the method. The default boundary conditions correspond to half of the “MBB-beam”, as shown in Fig. 2.

Load is applied vertically in the upper left corner and there are symmetric boundary conditions along the left edge and the structure is supported horizontally in the lower right corner.

Based on the 99 line code (Sigmund 2001); the advance details of Matlab code was described in the following subsections.

4.1 Codes of MMA based optimizer

The MMA was used to update design variables. The starting function was written in Matlab code as:

L=*x*-0.1*(xhigh-xlow)*ones(nely,nelx);

with

high=(*x*-*L*).^2.*-dc./(xlow-*L*).^2 ;

low=(*x*-*L*).^2.*-dc./(xhigh-*L*).^2;

therefore

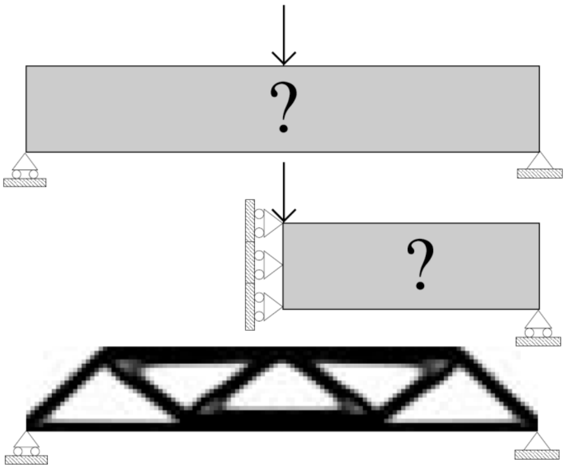


Fig. 2 Topology optimization of the MBB-beam. Top: full design domain, middle: half design domain with symmetry boundary conditions and bottom: resulting topology optimized beam (both halves)

xnew=max(xlow,min(xhigh,*L*+abs(*x*-*L*).*sqrt(-dc./lmid)));
where lmid = 0.5*(l2+l1);
while l2 = min(min(high));
l1 = max(max(low));
And sum(sum(xnew)) was also known as the material volume.

4.2 Codes of accelerating design variables (ADV) using MRHF

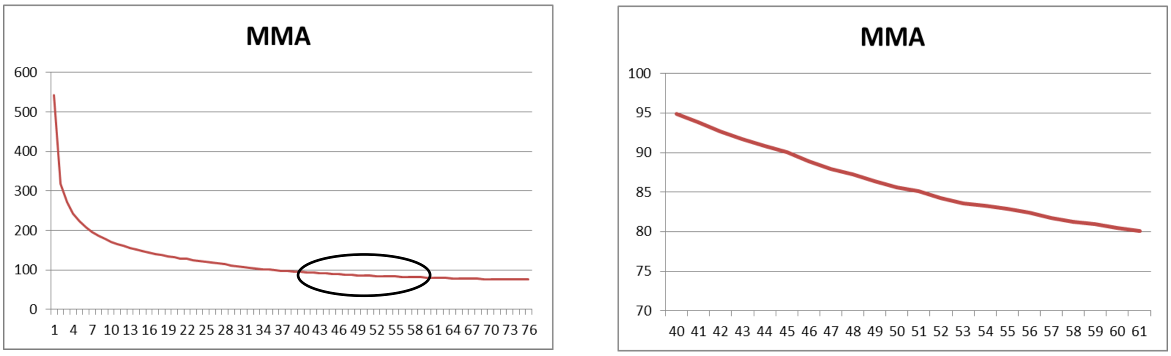
The previous section was discussed the moved regularized Heaviside function (MRHF) in Eqs. (14) (15) (16) and (17). These functions rewrote in the code as below:

$\mathrm{MRHF}_1(\phi_{ave})$:
x(1:nely,1:nelx)=0.75*(*x*(1:nely,1:nelx)-0.5)./chi-0.75*1/3*(*x*(1:nely,1:nelx)-0.5).^3/chi^3+1/2;
$\mathrm{MRHF}_2(\phi_{ave})$:
x(1:nely,1:nelx)=1/2+2/3.141592654*atan((*x*(1:nely,1:nelx)-0.5)/chi);
$\mathrm{MRHF}_3(\phi_{ave})$:
x(1:nely,1:nelx)=1/2+1/2.*(x(1:nely,1:nelx)-0.5)/chi+1/2*1/3.141592654*sin(3.141592654 *(x(1:nely,1:nelx)-0.5)/chi);
$\mathrm{MRHF}_4(\phi_{ave})$:
x(1:nely,1:nelx)=1/2+1/2*sin(3.141592654*(*x*(1:nely,1:nelx)-0.5)/(2*chi));

Many repetitions of ADV were also used to expect the fast convergence. The repeatable times will be mentioned on the numerical example section.

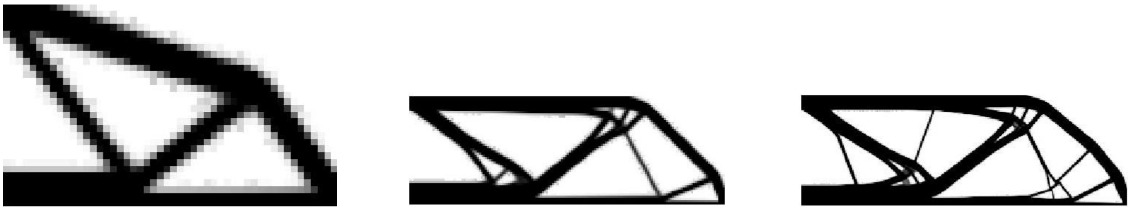
4.3 Comparisons of optimal results of OC and MMA

Fig. 4(a) shows the resulting density distribution obtained by the code given in the Appendix called with the input line: *topmma*(50,30,0.4,3,1.5). Then, the calculation from MATLAB shows the convergent of MMA method as shown in Fig. 3 below. The shape of chart shows that MMA convergence takes unsatisfactory results. Beside, in the end of the curve, the oscillations mean



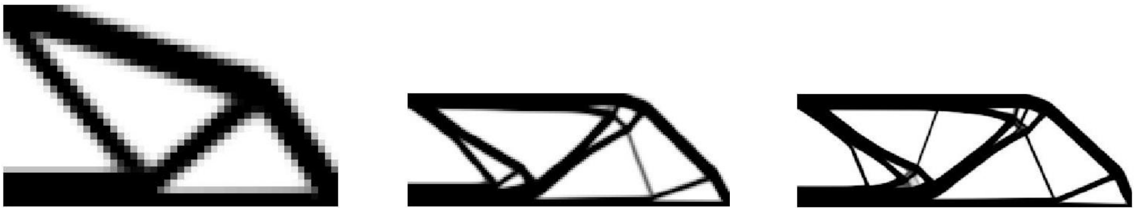
(a) (b)

Fig. 3 The convergence curves of MMA in mesh 50 × 30 elements with (a) is full chart and (b) is zoomed in the circle of (a)



(a) (b) (c)

Fig. 4 Optimized design of the MMB beam with the MMA code in order to (a) 50×30 elements, (b) 150×50 elements, (c) 300×100 elements

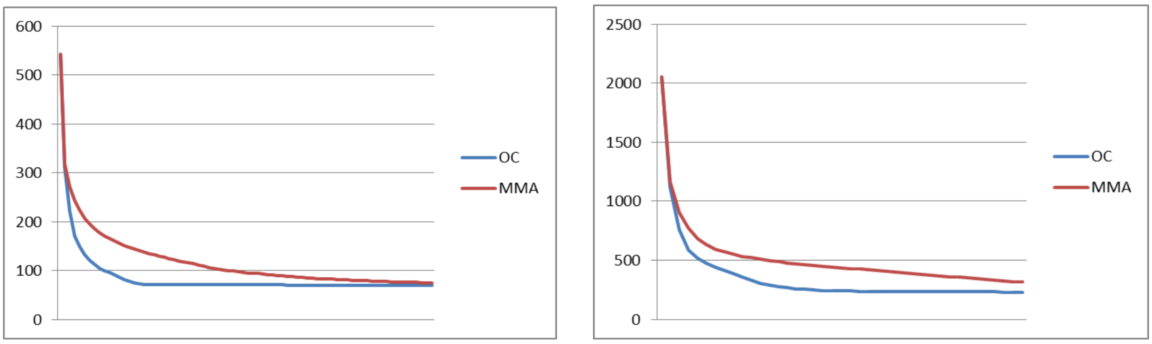


(a) (b) (c)

Fig. 5 Optimized design of the MMB beam with the OC code in order to (a) 50×30 elements, (b) 150×50 elements and (c) 300×100 elements

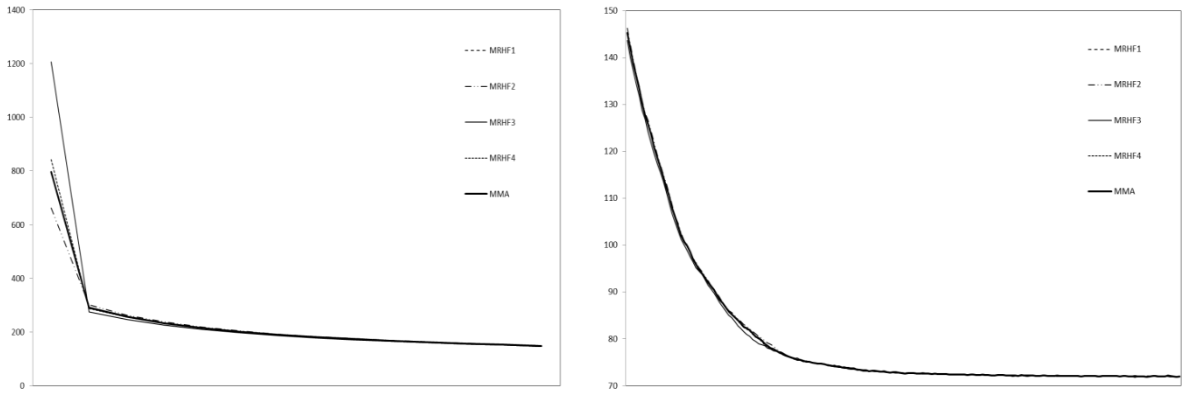
MMA can not get the convergence (Sen and Roy 2013). Fig. 3(b) is zoomed in the tail of the Fig. 3(a) to show the oscillations of MMA. In order to compare the optimized designs of the MBB beam between the 99 line and updated MMA code, the output were divided in two parts in Figs. 4 and 5 to show the differences between two methods.

According to the results, the optimal topologies are almost the same, but MMA produces the more details. In Fig. 6, the convergence of OC methods and MMA is compared. For all cases of Fig. 6(a) and (b), the convergence shapes of MMA are almost the same while those of OC are different. The quantity of elements used to discretize the design domain of OC method as much, the convergence take longer; while MMA was still the same. MMA was the best in case of many element domains.



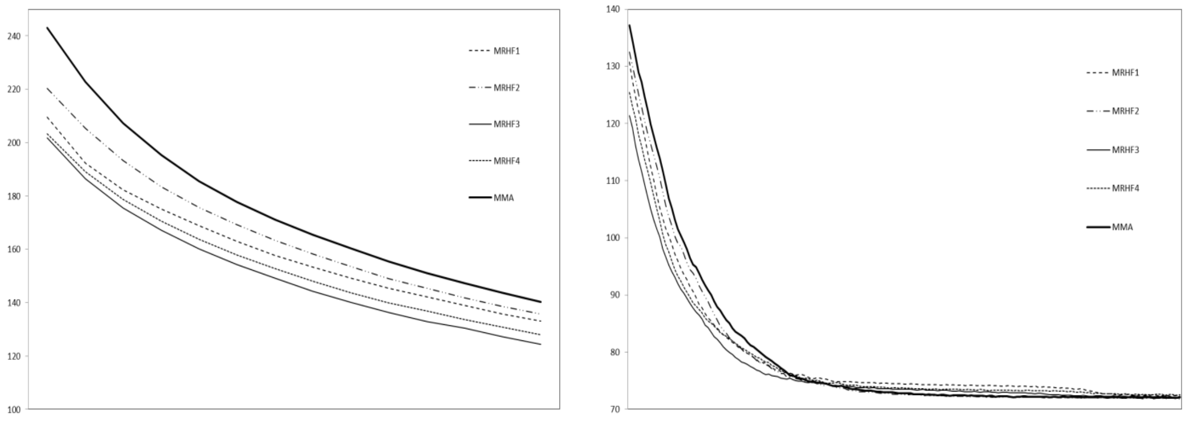
(a) (b)

Fig. 6 The convergence curves of OC methods and MMA in mesh (a) 50×30 elements and (b) 300×100 elements



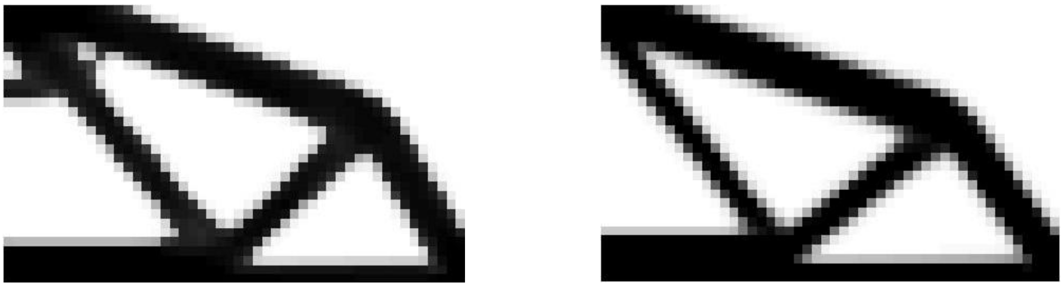
(a) Iterations from 1 to 14 (b) Iterations from 14 to 200

Fig. 7 Convergence histories of objective function under ADV with once repetition of different MRHF 1~4 with χ=0.5



(a) Iterations from 1 to 14 (b) Iterations from 14 to 200

Fig. 8 Convergence histories of objective function under ADV with fourth repetition of different MRHF 1~4 with χ=0.5



(a) ADV of MRHF2 for MMA (b) ADV of MRHF2 for OC

Fig. 9 Comparisons of oscillation results of optimal topology configuration

4.4 Effects of ADV on the convergence of MMA

Fig. 7 shows the histories of convergence of objective function between MMA and MMA added ADV. Once repetition of ADV yields (type of MRHF from 1 to 4) produces the results that the convergence was similar to the original MMA method. Then, the four times repetition of ADV yields (type of MRHF from 1 to 4) is showed in Fig. 8. In this case, ADV yields the quick convergence of objective function. Finally, MRHF of type 3 obtained the best convergence.

Fig. 9 shows optimal topology by using MMA and OC combined with ADV. As can be seen, ADV of MRHF2 removes the oscillation of MMA convergence as shown in Fig. 4(a).

5. Conclusions

This study presents a revised MATLAB code considering ADV for improving computational convergence of typical material topology optimization using OC and MMA as optimizers. According to the result of OC and MMA, MMA provide the convergence longer than the OC. Numerical examples by the revised MATLAB codes verify that improvement of convergence and the removal of oscillation at MMA can be guaranteed by using the present acceleration method of design variables through applying appropriate MRHF.

Acknowledegements

This research was supported by a grant (code# 2013R1A1A2057502 & 2014R1A1A3A04051296) from the National Research Foundation of Korea (NRF) funded by the Korea government.

References

Andreassen, E., Anders, C., Mattias, S., Stefanov, L.B. and Sigmund, O. (2011), "Efficient topology optimization in MATLAB using 88 lines of code", *Struct. Multidiscip. O.*, **43**(1), 1-16.

Bendsøe, M.P. and Kikuchi, N. (1988), "Generating optimal topologies in optimal design using a homogenization method", *Comput. Meth. Appl. Mech. Eng.*, **71**, 197-224.

Bendsøe, M.P. (1995), *Optimization of Structural Topology, Shape, and Material*, Berlin, Heidelberg, New York, Springer.

Challis, V.J. (2010), "A discrete level-set topology optimization code written in Matlab", *Struct. Multidiscip. O.*, **41**(3), 453-464.

Fu, Y. and Zhang, X. (2014), "An optimization approach for black-and-white and hinge-removal topology designs", *J. Mech. Sci. Tech.*, **28**(2), 581-593.

Lee, D.K., Shin, S.M. and Park, S.S. (2007), "Computational morphogenesis based structural design by using material topology optimization", *Mech. Bas. Des. Struct. Mach.*, **35**(1), 39-58.

Patnaik, S.N., Guptill, D.J. and Berke, L. (1995), "Merits and limitations of optimality criteria method for structural optimization", *Internat. J. Numer. Mech. Eng.*, **38**, 3087-3120.

Rozvany, G.I.N. (1989), "Structural design via optimality criteria", *Mech. Elast. Inelast. Solid.*, **8**, 1-461.

Sen, M. and Roy, S. (2013), "On paranormed type fuzzy I-convergent double multiplier sequence", *Kuwait J. Sci.*, **40**(1), 1-12.

Sigmund, O. (2001), "A 99 line topology optimization code written in Matlab", *Struct. Multidiscip. O.*, **21**, 120-127.

Suresh, K. (2010), "A 199-line Matlab code for Pareto-optimal tracing in topology optimization", *Struct. Multidiscip. O.*, **42**(5), 665-679.

Lee D.K. and Shin S.M. (2014c), "Advanced high strength steel tube diagrid using TRIZ and nonlinear pushover analysis", *J. Construct. Steel Res.*, **96**, 151-158.

Lee D.K. and Shin S.M. (2015), "Nonlinear pushover analysis of concrete column reinforced by multi-layered, high strength steel UL700 plates", *Eng. Struct.*, **90**, 1-14.

Svanberg, K. (1987), "The method of moving asymptotes - a new method for structural optimizztion", *Int. J. Numer. Meth. Eng.*, **24**, 359-373.

Zillober, C. (1993), "A globally convergent version of the method of moving asymptotes", *Struct. Optim.*, **6**, 166-174.

Bruns, T.E. (2007), "Topology optimization by penalty (TOP) method", *Comput. Meth. Appl. Mech. Eng.*, **196**(45-48), 4430-4443.

Lee, D.K., Kim, J.H., Starossek, U. and Shin, S.M. (2012), "Evaluation of structural outrigger belt truss layouts for tall buildings by using topology optimization", *Struct. Eng. Mech.*, **43**(6), 711-724.

Yi, J., Rong, J., Zeng, T. and Huang, X. (2013), "A topology optimization method for multiple load cases and constraints based on element independent nodal density", *Struct. Eng. Mech.*, **45**(6), 759-777.

Lee, D.K., Lee, J.H. and Ahn, N.S. (2014a), "Generation of structural layout in use for '0-1' material considering n-order eigenfrequency dependence", *Mater. Res. Inn.*, **18**(2), 833-839.

Lee, D.K., Lee, J.H., Lee, K.H. and Ahn, N.S. (2014b), "Evaluating topological optimized layout of building structures by using nodal material density based bilinear interpolation", *J. Asian Arch. Build. Eng.*, **13**(2), 421-428.

CC