A MATLAB implementation for topology optimization of compliance minimization problems based on the standard finite-volume theory for continuum elastic structures

Marcelo Vitor Oliveira Araujo[1], Eduardo Nobre Lages[1], Márcio André Araújo Cavalcante[2]

[1]*Center of Technology, Federal University of Alagoas*
Av. Lourival Melo Mota, Tabuleiro do Martins, Maceió, 57072-900, Alagoas, Brazil
marcelo.vitor.o.a@gmail.com, end@ctec.ufal.br
[2]*Campus of Engineering and Agricultural Sciences, Federal University of Alagoas*
BR-104, Rio Largo, 57100-000, Alagoas, Brazil
marcio.cavalcante@ctec.ufal.br

Abstract. Topology optimization is an important technique for the design of optimum structures. Its main objective is to determine the best material distribution inside of an analysis domain. In the last three decades, a significant part of the advances in structural topology optimization has been achieved by employing finite-element strategies for structural analysis. Therefore, the advantages and disadvantages of this numerical technique are well-known. For instance, the checkerboard pattern is directly associated with the finite-element method numerical assumptions, which leads to some artificial stiffness. An alternative to the finite-element method is the finite-volume theory, which has been shown to be an efficient checkerboard-free numerical technique. Many algorithm implementations have been published for educational purposes over the last decades to promote topology optimization strategies. However, most of these algorithms are constructed based on the finite-element method. Therefore, the present paper proposes a MATLAB® implementation of a topology optimization approach for compliance minimization problems based on the standard finite-volume theory of linear elastic continuum structures. A sensitivity filter is also implemented to control the mesh dependence and length scale issues.

Keywords: topology optimization, finite-volume theory, educational MATLAB® code.

1 Introduction

Topology optimization has grown as an important and robust technique for designing optimized structures. Generally, these algorithms seek to establish the best material distribution inside the analysis domain, since the problem constraints are attended. Recently, different authors have presented educational algorithms for topology optimization, offering foundations for the understanding and development of different topology optimization strategies, Ferrari et al. [1]. Since the pioneer 99-line code proposed by Sigmund [2], other educational codes have appeared, however many of those are directly related to the optimal design of compliance minimization problems employing the traditional finite-element method. Ferrari and Sigmund [3] have presented a list of those contributions.

Since the pioneering work of Bendsøe and Kikuchi [4] in the homogenization method, the finite element-based strategy for structural topology optimization has received great attention and experienced considerable progress, Wang and Wang [5]. Therefore, the advantages and disadvantages are well-known. An alternative technique to the finite-element method is the finite-volume theory, which employs the volume-average of the different fields that define the material behavior and imposes the boundary and continuity conditions in an averaged sense. This technique has shown to be a well suitable method for elastic stress analysis in solid mechanics, investigations of its efficiency can be found in Cavalcante et al. [6-8] and Cavalcante and Pindera [9,10]. The satisfaction of equilibrium equations at the subvolume level, concomitant to kinematic and static continuity conditions established in a surface-averaged sense between common faces of adjacent subvolumes, are features that distinguish the finite-volume theory from the finite-element method.

Some recent studies have promoted the incorporation of the finite-volume theory into the topology optimization framework, starting with Araujo et al. [11], who have proposed a topology optimization approach for compliance minimization based on the standard (or zeroth-order) finite-volume theory. Subsequently, Araujo et al. [12] have proposed a similar topology optimization approach based on the generalized finite-volume theory. Finally, Araujo et al. [13] have discussed the differences between total strain energy and external work done in deforming materials in quasi-static analysis, which has helped to the definition of the proposed compliance function. This contribution addresses an educational MATLAB® code for topology optimization of continuum elastic structures based on the finite-volume theory considering the compliance minimization problem subject to a volume constraint.

2 Finite-volume theory

Considering a rectangular domain in $x_1 - x_2$ plane with $0 \leq x_1 \leq L$ and $0 \leq x_2 \leq H$, which is discretized in N_β horizontal subvolumes and N_γ vertical subvolumes. The subvolume dimensions are l_q and h_q for $q = 1, \dots, N_q$, where $N_q = N_\beta \cdot N_\gamma$ is the total number of subvolumes. Following Cavalcante and Pindera [9], the displacement of a subvolume q can be approximated by an incomplete quadratic version of Legendre polynomial expansion in the local coordinate system as follows:

$$u_i^{(q)} = W_{i(00)}^{(q)} + x_1^{(q)} W_{i(10)}^{(q)} + x_2^{(q)} W_{i(01)}^{(q)} + \frac{1}{2}\left(3\left(x_1^{(q)}\right)^2 - \frac{l_q^2}{4}\right) W_{i(20)}^{(q)} + \frac{1}{2}\left(3\left(x_2^{(q)}\right)^2 - \frac{h_q^2}{4}\right) W_{i(02)}^{(q)}, \tag{1}$$

where $i = 1,2$ and $W_{i(mn)}^{(q)}$ are unknown coefficients of the displacement field.

2.1 Local stiffness matrix

Following Bansal and Pindera [14], the surface-averaged displacement components of a generic subvolume can be defined as

$$\begin{aligned} \bar{u}_i^{(1,3)} &= \frac{1}{l_q}\int_{-\frac{l_q}{2}}^{\frac{l_q}{2}} u_i\left(x_1^{(q)}, \mp \frac{h_q}{2}\right) dx_1^{(q)} \\ \bar{u}_i^{(2,4)} &= \frac{1}{h_q}\int_{-\frac{h_q}{2}}^{\frac{h_q}{2}} u_i\left(\pm \frac{l_q}{2}, x_2^{(q)}\right) dx_2^{(q)} \end{aligned}, \tag{2}$$

where the superscript indicates the subvolume face number.

Substituting eq. (1) into eq. (2), eight expressions are obtained for the surface-averaged displacements, which can be organized as follows:

$$\bar{\boldsymbol{u}}^{(q)} = \boldsymbol{A}_{(8\times8)}^{(q)} \boldsymbol{W}^{(q)} + \boldsymbol{a}_{(8\times2)}^{(q)} \boldsymbol{W}_{(00)}^{(q)}, \tag{3}$$

where $\bar{\boldsymbol{u}}^{(q)}$ is the local surface-averaged displacement vector, $\boldsymbol{W}^{(q)}$ is the vector containing the first and second-order unknown coefficients and $\boldsymbol{W}_{(00)}^{(q)}$ is the vector containing the zeroth-order unknown coefficients. $\boldsymbol{A}_{(8\times8)}^{(q)}$ and $\boldsymbol{a}_{(8\times2)}^{(q)}$ are matrixes that depend on the geometric features of the subvolume q.

Based on linear elastic stress analysis, the surface-averaged traction components can be evaluated as

$$\begin{aligned} \bar{t}_i^{(1,3)} &= \mp \frac{1}{l_q}\int_{-\frac{l_q}{2}}^{\frac{l_q}{2}} \sigma_{2i}\left(x_1^{(q)}, \mp \frac{h_q}{2}\right) dx_1^{(q)} \\ \bar{t}_i^{(2,4)} &= \pm \frac{1}{h_q}\int_{-\frac{h_q}{2}}^{\frac{h_q}{2}} \sigma_{1i}\left(\pm \frac{l_q}{2}, x_2^{(q)}\right) dx_2^{(q)} \end{aligned}. \tag{4}$$

Considering linear elastic isotropic materials, eight expressions for the surface-averaged tractions can be obtained in terms of the unknown coefficients

$$\bar{\boldsymbol{t}}^{(q)} = \boldsymbol{B}_{(8\times8)}^{(q)} \boldsymbol{W}^{(q)}, \tag{5}$$

where $\boldsymbol{B}^{(q)}_{(8\times8)}$ is the local surface-averaged traction vector.

Stablishing the equilibrium at the subvolume level in the absence of body forces, we have

$$\sum_{p=1}^{4} \bar{\boldsymbol{t}}^{(q,p)} L_p^{(q)} = \boldsymbol{0}_{(2\times1)}, \tag{6}$$

where $L_1^{(q)} = l_q$, $L_2^{(q)} = h_q$, $L_3^{(q)} = l_q$ and $L_4^{(q)} = h_q$ are the faces' lengths of a generic subvolume q, and $\bar{\boldsymbol{t}}^{(q,p)}$ can be defined as

$$\bar{\boldsymbol{t}}^{(q,p)} = \boldsymbol{B}^{(q,p)}_{(2\times8)} \left(\boldsymbol{A}^{(q)}_{(8\times8)}\right)^{-1} \bar{\boldsymbol{u}}^{(q)} - \boldsymbol{B}^{(q,p)}_{(2\times8)} \left(\boldsymbol{A}^{(q)}_{(8\times8)}\right)^{-1} \boldsymbol{a}^{(q)}_{(8\times2)} \boldsymbol{W}^{(q)}_{(00)}. \tag{7}$$

By substituting eq. (7) in eq. (6), the following expression can be obtained

$$\left(\sum_{p=1}^{4} \boldsymbol{B}^{(q,p)}_{(2\times8)} L_p^{(q)}\right) \left(\boldsymbol{A}^{(q)}_{(8\times8)}\right)^{-1} \bar{\boldsymbol{u}}^{(q)} - \left(\sum_{p=1}^{4} \boldsymbol{B}^{(q,p)}_{(2\times8)} L_p^{(q)}\right) \left(\boldsymbol{A}^{(q)}_{(8\times8)}\right)^{-1} \boldsymbol{a}^{(q)}_{(8\times2)} \boldsymbol{W}^{(q)}_{(00)} = \boldsymbol{0}_{(2\times1)}. \tag{8}$$

From eq. (8), the vector containing the zeroth unknown coefficients can be expressed as

$$\boldsymbol{W}^{(q)}_{(00)} = \bar{\boldsymbol{a}}^{(q)}_{(2\times8)} \bar{\boldsymbol{u}}^{(q)}, \tag{9}$$

where $\bar{\boldsymbol{a}}^{(q)}_{(2\times8)} = \left[\left(\sum_{p=1}^{4} \boldsymbol{B}^{(q,p)}_{(2\times8)} L_p^{(q)}\right) \left(\boldsymbol{A}^{(q)}_{(8\times8)}\right)^{-1} \boldsymbol{a}^{(q)}_{(8\times2)}\right]^{-1} \left(\sum_{p=1}^{4} \boldsymbol{B}^{(q,p)}_{(2\times8)} L_p^{(q)}\right) \left(\boldsymbol{A}^{(q)}_{(8\times8)}\right)^{-1}$. By replacing eq. (9) in eq. (3), the following expression is obtained

$$\boldsymbol{W}^{(q)} = \bar{\boldsymbol{A}}^{(q)}_{(8\times8)} \bar{\boldsymbol{u}}^{(q)}, \tag{10}$$

where $\bar{\boldsymbol{A}}^{(q)}_{(8\times8)} = \left(\boldsymbol{A}^{(q)}_{(8\times8)}\right)^{-1} - \left(\boldsymbol{A}^{(q)}_{(8\times8)}\right)^{-1} \boldsymbol{a}^{(q)}_{(8\times2)} \bar{\boldsymbol{a}}^{(q)}_{(2\times8)}$. Then, the local system of equations for a generic subvolume can be stablished as

$$\bar{\boldsymbol{t}}^{(q)} = \boldsymbol{K}^{(q)}_{(8\times8)} \bar{\boldsymbol{u}}^{(q)}, \tag{11}$$

where $\boldsymbol{K}^{(q)}_{(8\times8)} = \boldsymbol{B}^{(q)}_{(8\times8)} \bar{\boldsymbol{A}}^{(q)}_{(8\times8)}$ is the local stiffness matrix. The global stiffness matrix is assembled by considering the individual contribution of each subvolume in the discretized domain.

3 Topology optimization problem

A significant part of the advances in topology optimization has been achieved by considering compliance minimization problems, whose concepts are well-stablished in the context of finite-element strategies. In this contribution, the minimum compliance minimization problem is implemented considering linear elastic stress analysis based on the finite-volume theory. Following Araujo et al. [13], the topology optimization problem based on the power-law approach for compliance minimization based on the standard finite-volume theory can be written as

$$\begin{cases} \min c(\boldsymbol{\rho}) = \sum_{q=1}^{N_q} \left(\rho_q\right)^p \boldsymbol{W}^{(q)} \boldsymbol{D}^{(q)} \boldsymbol{W}^{(q)} \\ \text{subject to:} \\ \dfrac{V(\boldsymbol{\rho})}{\bar{V}} = f \\ 0 < \rho_{min} \le \rho_q \le 1 \end{cases}, \tag{12}$$

where $\boldsymbol{D}^{(q)} = \int_{-\frac{h_q}{2}}^{\frac{h_q}{2}} \int_{-\frac{l_q}{2}}^{\frac{l_q}{2}} \frac{1}{2} \left(\boldsymbol{E}^{(q)}\right)^T \boldsymbol{C}^{(q)} \boldsymbol{E}^{(q)}\, dx_1^{(q)}\, dx_2^{(q)}$, $\boldsymbol{E}^{(q)}$ is the kinematic matrix that relates the strain tensor components with the unknown coefficients, $\boldsymbol{C}^{(q)}$ is the stiffness tensor, $V(\boldsymbol{\rho})$ and $\bar{V}$ are the material and reference domain volumes, respectively, $\boldsymbol{\rho}$ is the relative density tensor, p is the penalty factor, f is the prescribed volume fraction and ρ_{min} is the minimum relative density to avoid singularity in the stiffness matrix. This optimization problem is solved using the optimality criteria (OC) method.

3.1 Mesh-independency filter

Following Sigmund [2], to avoid the occurrence of mesh dependence, it is suggested to modify the subvolume sensitivities by using the following expression:

$$\frac{\partial c}{\partial \rho_q} = \frac{1}{\rho_q \sum_{e=1}^{N_q} \widehat{H}_e} \sum_{e=1}^{N_q} \widehat{H}_e \rho_e \left(\frac{\partial c}{\partial \rho_e}\right), \tag{13}$$

where $\widehat{H}_e$ is the convolution operator (weighting function) given as

$$\widehat{H}_e = R - \mathrm{dist}(q,e) \text{ for } \mathrm{dist}(q,e) \leq R \text{ and } \widehat{H}_e = 0 \text{ otherwise}, \tag{14}$$

where $\mathrm{dist}(q,e)$ is the distance between the subvolume center of q and e. To consider only the contributions of neighbor subvolumes (with shared nodes), it is adopted a filter radius of $R = 1.01\sqrt{l_q^2 + h_q^2}$.

4 MATLAB implementation

The proposed topology optimization algorithm was implemented using the MATLAB R2016a (64-bits) version and it is presented in the Appendix. This code is composed of a main program, that calls other algorithms for the problem solution, a finite-volume theory code, containing its local stiffness matrix, a mesh-independent filter, and the OC method implementation. This proposed algorithm is inspired by the renowned 99 line MATLAB® code proposed by Sigmund [2].

The main program can be called in the prompt by

`FVT_TOP(SP,L,H,nH,nL,MY,nu,bd,bf,vf,penal,rmax),` (15)

where `SP` holds the information for state of plane stress or strain, `L` and `H` are the beam length and height, respectively, `nL` and `nH` are the number of horizontal and vertical subvolumes, respectively, `MY` is the material Young modulus, `nu` is the Poisson ratio, `bd` holds the information for essential boundary conditions, `bf` holds the information for natural boundary conditions, `vf` is the initial volume fraction percentage, `penal` is the adopted penalty factor, and `rmax` is the filter maximum radius. The algorithm output informations are `U` and `x`, which represent respectively the global displacement and relative density vectors. The main program starts by indexing the degrees of freedom (`dof`) of the analysis domain. Then, the prescribed boundary conditions are inserted in the analysis by creating the vectors `udof` and `tdof` with the prescribed surface-averaged displacements and tractions.

The code STIFF_0TH calls the formation of the local stiffness matrix based on the finite-volume theory

`STIFF_0TH(SP,MY,nu,l,h),` (16)

where `l` and `h` are the subvolume horizontal and vertical dimensions, respectively. Then, the main program distributes the material in the design domain and the gradient filter is initialized. The mesh-independent filter code modifies the subvolume sensitivity by employing a weighting function obtained considering the neighbor subvolumes with shared nodes. Returning to the main code, the looping is initialized considering two different alternatives. Firstly, it can be considered a fixed penalization scheme, where the penalty factor is constant throughout the optimization process. The second procedure considers a continued penalization scheme, where the penalty factor increases gradually, and it is implemented as a vector. For instance, Araujo et al. [11,12] have employed a continued penalization scheme, where `penal` was set up as `1:0.5:4`.

As convergence criteria, the maximum tolerance for the change in design variables between successive iterations is established as 1%, which is designated by `while change>0.01` in the main code. Then, the looping is initialized followed by the global stiffness matrix assemblage, where the individual contribution of each subvolume is considered. The global surface-averaged displacement vector is obtained, and the objective function and its gradient are calculated by the following called in the main program:

`strainenergy2(U,l,h,SP,MY,nu,Nq,x,p,dof),` (17)

where `p` represents the current penalty factor. The compliance function for this approach is calculated considering twice the total strain energy in a deforming material, as defined in Araujo et al. [13]. Finally, the relative density vector is updated considering the OC method, which is called by

`OC_METHOD(Vm,Ve,x,dc,p),` (18)

where `Vm` is the effective model volume, `Ve` is the subvolume volume, and `dc` is the vector containing the subvolume sensitivities.

5 Conclusions

This contribution pretends to become more popular the application of the finite-volume theory in structural topology optimization problems. Araujo et al. [11,12] have demonstrated the efficiency of the proposed approach by reducing the optimum topology perimeter, which is a desired feature for manufacturing once it optimizes the use of material in the designed structure. This algorithm has the aim to help researchers by proposing an easy-to-follow code in MATLAB® for topology optimization of continuum elastic structures applying the finite-volume theory.

Acknowledgements. The authors acknowledge the financial support provided by CAPES and CNPq.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors or has the permission of the owners to be included here.

References

[1] F. Ferrari, O. Sigmund, and J. K. Guest. "Topology optimization with linearized buckling criteria in 250 lines of Matlab". *Structural and Multidisciplinary Optimization*, vol. 63, pp. 3045–3066, 2021.
[2] O. Sigmund. "A 99 line topology optimization code written in Matlab". *Structural and Multidisciplinary Optimization*, vol. 21, pp. 120–127, 2001.
[3] F. Ferrari and O. Sigmund. "A new generation 99 line Matlab code for compliance topology optimization and its extension to 3D". *Structural and Multidisciplinary Optimization*, vol. 62, pp. 2211–2228, 2020.
[4] M. P. Bendsøe and N. Kikuchi. "Generating optimal topologies in structural design using a homogenization method". *Computer Methods in Applied Mechanics and Engineering*, vol. 71, n. 2, pp. 197–224, 1988.
[5] S. Wang and M. Y. Wang. "Radial basis function and level set method for structural topology optimization". *International Journal for Numerical Methods in Engineering*, vol. 65, n. 12, pp. 2060–2090, 2006.
[6] M. A. A. Cavalcante, S. P. C. Marques, and M.-J. Pindera. "Parametric formulation of the finite-volume theory for functionally graded materials – Part I: analysis". *Journal of Applied Mechanics*, vol. 74, n. 5, pp. 935–945, 2007.
[7] M. A. A. Cavalcante, S. P. C. Marques, and M.-J. Pindera. "Parametric formulation of the finite-volume theory for functionally graded materials – Part II: numerical results". *Journal of Applied Mechanics*, vol. 74, n. 5, pp. 946–957, 2007.
[8] M. A. A. Cavalcante, S. P. C. Marques, and M.-J. Pindera. "Computational aspects of the parametric finite-volume theory for functionally graded materials". *Computational Materials Science*, vol. 44, n. 2, pp. 422–438, 2008.
[9] M. A. A. Cavalcante and M.-J. Pindera. "Generalized finite-volume theory for elastic stress analysis in solid mechanics – part I: framework". *Journal of Applied Mechanics*, vol. 79, n. 5, pp. 051006, 2012.
[10] M. A. A. Cavalcante and M.-J. Pindera. "Generalized finite-volume theory for elastic stress analysis in solid mechanics – part I: results". *Journal of Applied Mechanics*, vol. 79, n. 5, pp. 051007, 2012.
[11] M. V. O. Araujo, E. N. Lages, and M. A. A. Cavalcante. "Checkerboard free topology optimization for compliance minimization applying the finite-volume theory". *Mechanics Research Communications*, vol. 108, pp. 103581, 2020.
[12] M. V. O. Araujo, E. N. Lages, and M. A. A. Cavalcante. "Checkerboard-free topology optimization for compliance minimization of continuum elastic structures based on the generalized finite-volume theory". *Latin American Journal of Solids and Structures*, vol. 17, n. 8, pp. 1–21, 2020.
[13] M. V. O. Araujo, E. N. Lages, and M. A. A. Cavalcante. "Energy analysis of continuum elastic structures by the generalized finite-volume theory". *Acta Mechanica*, 2021.
[14] Y. Bansal and M.-J. Pindera. "Efficient reformulation of the thermoelastic higher-order theory for functionally graded materials". *Journal of Thermal Stress*, vol. 26, n. 11-12, pp. 1055–1092, 2003.

Appendix

```
function [U,x]=FVT_TOP(SP,L,H,nH,nL,MY,nu,bd,bf,vf,penal,rmax)
Nq=nH*nL; l=L/nL; h=H/nH; Nhf=nL*(nH+1); faces=zeros(4,Nq);
for i=1:nH
    for j=1:nL
        q=j+(i-1)*nL; % Subvolume index
        faces(:,q)=[j+(i-1)*nL; Nhf+j+1+(i-1)*(nL+1); j+i*nL; Nhf+j+(i-1)*(nL+1)];
    end
end
dof(2:2:8,:)=2*faces; dof(1:2:7,:)=2*faces-1; ndof=max(max(dof)); U=zeros(ndof,1);
udof=[]; % Degrees of freedom with prescribed kinematic variables
```

```
for i=1:length(bd)
    udof=[udof;bd(i).dof];
end
f=zeros(ndof,1); % Global force vector
tdof=[]; % Degrees of freedom with prescribed static variables
for i=1:length(bf)
    f(bf(i).dof,1)=bf(i).sv; tdof=[tdof;bf(i).dof];
end
fdof=setdiff(1:ndof,udof); kl=STIFF_0TH(SP,MY,nu,l,h); x(1:Nq,1)=vf'/100;
Ve=l*h; Vm=Ve*sum(x); r2=rmax*rmax; % Initializing the mesh-independent filter
filter=repmat(struct('cg',[0;0],'neib',[],'fac',[]),Nq,1);
for i=1:nH
    for j=1:nL
        filter((i-1)*nL+j).cg=[L/nL/2+L/nL*(j-1);H/nH/2+H/nH*(i-1)];
    end
end
for i=1:Nq
    for j=i:Nq
        d2=(filter(i).cg(1)-filter(j).cg(1))^2+(filter(i).cg(2)-filter(j).cg(2))^2;
        if d2<r2
            filter(i).neib = [filter(i).neib j];
            filter(i).fac  = [filter(i).fac rmax-sqrt(d2)];
            if i~=j
                filter(j).neib = [filter(j).neib i];
                filter(j).fac  = [filter(j).fac filter(i).fac(end)];
            end
        end
    end
end
for p=penal
    loop=0; change=1;
    while change>0.01 % Starting iteration process
        loop=loop+1; xold=x; Kg=sparse(ndof,ndof);
        for q=1:Nq
            Kg(dof(:,q),dof(:,q))=Kg(dof(:,q),dof(:,q))+x(q)^p*kl;
        end
        U(fdof)=Kg(fdof,fdof)\f(fdof);
        [c,dc]=strainenergy2(U,l,h,SP,MY,nu,Nq,x,p,dof); dc=min(0.0,dc);
        dc=gradfilter(filter,x,dc); [x]=OC_METHOD(Vm,Ve,x,dc,p);
        change=max(abs(x-xold)); % Maximum change in the design variable
    end
end

function dcn = gradfilter(filter,ovf,dc)
for i=length(filter):-1:1
    dcn(i,1) = sum([filter(i).fac]'.*ovf(filter(i).neib).*dc(filter(i).neib))/...
               (ovf(i)*sum(filter(i).fac));
end

function [c,dc]=strainenergy2(U,l,h,SP,MY,nu,Nq,x,p,dof)
switch SP
    case 'PlaneStress'
        C11=MY/(1-nu^2); C12=nu*C11; C44=MY/(2*(1+nu));
    case 'PlaneStrain'
        C11=MY*(1-nu)/((1-2*nu)*(1+nu)); C12=nu*MY/((1-2*nu)*(1+nu));
        C44=MY/(2*(1+nu));
end
A0=[0 0 1/l 0 0 0 -1/l 0;...
    -1/h 0 0 0 1/h 0 0 0;...
    -24*C44/(l*(12*C44*l/h+12*h*C11/l)*h) 0 2/l^2-
24*h*C11/(l^3*(12*C44*l/h+12*h*C11/l)) 0 -24*C44/(l*(12*C44*l/h+12*h*C11/l)*h) 0
2/l^2-24*h*C11/(l^3*(12*C44*l/h+12*h*C11/l)) 0;...
    2/h^2-24*C44*l/(h^3*(12*C44*l/h+12*h*C11/l)) 0 -
24*C11/(h*(12*C44*l/h+12*h*C11/l)*l) 0 2/h^2-24*C44*l/(h^3*(12*C44*l/h+12*h*C11/l))
0 -24*C11/(h*(12*C44*l/h+12*h*C11/l)*l) 0;...
    0 0 0 1/l 0 0 0 -1/l;...
    0 -1/h 0 0 0 1/h 0 0;...
```

```
    0 -24*C11/(l*(12*C11*l/h+12*h*C44/l)*h) 0 2/l^2-
24*h*C44/(l^3*(12*C11*l/h+12*h*C44/l)) 0 -24*C11/(l*(12*C11*l/h+12*h*C44/l)*h) 0
2/l^2-24*h*C44/(l^3*(12*C11*l/h+12*h*C44/l));...
    0 2/h^2-24*C11*l/(h^3*(12*C11*l/h+12*h*C44/l)) 0 -
24*C44/(h*(12*C11*l/h+12*h*C44/l)*l) 0 2/h^2-24*C11*l/(h^3*(12*C11*l/h+12*h*C44/l))
0 -24*C44/(h*(12*C11*l/h+12*h*C44/l)*l)];
d00=[C11*l*h 0 0 0 0 C12*l*h 0 0;...
     0 C44*l*h 0 0 C44*l*h 0 0 0;...
     0 0 3*C11*l^3*h/4 0 0 0 0 0;...
     0 0 0 3*C44*l*h^3/4 0 0 0 0;...
     0 C44*l*h 0 0 C44*l*h 0 0 0;...
     C12*l*h 0 0 0 0 C11*l*h 0 0;...
     0 0 0 0 0 0 3*C44*l^3*h/4 0;...
     0 0 0 0 0 0 0 3*C11*l*h^3/4];
c=0; dc=zeros(Nq,1);
for q=1:Nq
    Ue=U(dof(:,q)); ce=Ue'*A0'*d00*A0*Ue; c=c+x(q)^p*ce; dc(q)=-p*x(q)^(p-1)*ce;
end

function k00=STIFF_0TH(SP,MY,nu,l,h)
switch SP
    case 'PlaneStress'
        C11=MY/(1-nu^2); C12=nu*C11; C44=MY/(2*(1+nu));
    case 'PlaneStrain'
        C11=MY*(1-nu)/((1-2*nu)*(1+nu)); C12=nu*MY/((1-2*nu)*(1+nu));
        C44=MY/(2*(1+nu));
end
alpha=[l^2*C44/(l^2*C44+h^2*C11) l^2*C11/(l^2*C11+h^2*C44)];
beta =[h^2*C11/(l^2*C44+h^2*C11) h^2*C44/(l^2*C11+h^2*C44)];
k00=[C44*(4-3*alpha(1))/h 0 -3*C44*beta(1)/h 0 C44*(2-3*alpha(1))/h 0 -
3*C44*beta(1)/h 0;...
     0 C11*(4-3*alpha(2))/h -C12/l -3*C11*beta(2)/h 0 C11*(2-3*alpha(2))/h C12/l -
3*C11*beta(2)/h;...
     -3*C11*alpha(1)/l -C12/h C11*(4-3*beta(1))/l 0 -3*C11*alpha(1)/l C12/h C11*(2-
3*beta(1))/l 0;...
     0 -3*C44*alpha(2)/l 0 C44*(4-3*beta(2))/l 0 -3*C44*alpha(2)/l 0 C44*(2-
3*beta(2))/l;...
     C44*(2-3*alpha(1))/h 0 -3*C44*beta(1)/h 0 C44*(4-3*alpha(1))/h 0 -
3*C44*beta(1)/h 0;...
     0 C11*(2-3*alpha(2))/h C12/l -3*C11*beta(2)/h 0 C11*(4-3*alpha(2))/h -C12/l -
3*C11*beta(2)/h;...
     -3*C11*alpha(1)/l C12/h C11*(2-3*beta(1))/l 0 -3*C11*alpha(1)/l -C12/h C11*(4-
3*beta(1))/l 0;...
     0 -3*C44*alpha(2)/l 0 C44*(2-3*beta(2))/l 0 -3*C44*alpha(2)/l 0 C44*(4-
3*beta(2))/l];
k00(1,4)=-C44/l; k00(1,8)=C44/l; k00(5,4)=C44/l; k00(5,8)=-C44/l;
k00(4,1)=-C44/h; k00(4,5)=C44/h; k00(8,1)=C44/h; k00(8,5)=-C44/h;
end

function [xnew]=OC_METHOD(Vm,Ve,x,dc)
l1=0; l2=100000; move=0.2;
while (l2-l1)/(l1+l2)>1e-5
    lmid=(l1+l2)/2;
    xnew=max(0.001,max(x-move,min(1.,min(x+move,x.*(-dc./Ve/lmid).^(1/2)))));
    if Ve*sum(xnew)>Vm
        l1=lmid;
    else
        l2=lmid;
    end
end
```