COBEM 2023 27TH INTERNATIONAL CONGRESS OF MECHANICAL ENGENEERING

ABCM Brazilian Society of Mechanical Sciences and Engineering
27th ABCM International Congress of Mechanical Engineering
December 04-08, 2023, Florianópolis-SC, Brazil

COB-2023-1154
AN EDUCATIONAL MATLAB CODE FOR TOPOLOGY OPTIMIZATION OF THICK-THIN PLATES USING ARBITRARY POLYGONAL MESHES

Diego S. Duarte
Pontifical Catholic University of Rio de Janeiro - R. Marques de São Vicente, 225, Rio de Janeiro - RJ, 22451-900
Federal University of Bahia - R. Prof. Aristídes Novis, 2, Salvador - BA, 40210-630
diegoduarte@aluno.puc-rio.br, diegoduarte@ufba.br

Ivan F. M. Menezes
Pontifical Catholic University of Rio de Janeiro - R. Marques de São Vicente, 225, 22451-900, Rio de Janeiro, Brazil
ivan@puc-rio.br

Abstract. *Arbitrary polygonal finite element meshes have been widely used in topology optimization in the past few years. This class of meshes offers great flexibility for treating complex geometries, as required in most real-world engineering problems. For topology optimization purposes, arbitrary polygonal meshes have been proven to present an efficient performance. Due to the edge-to-edge connections between elements, instead of vertice-to-vertice connections, as in conventional triangular and quadrilateral meshes, the polygonal elements demonstrated not to be susceptible to numerical instabilities such as checkerboard patterns and one-node connections. Another issue on either lower-order triangular/quadrilateral or arbitrary polygonal meshes in Reissner-Mindlin plate formulations is the shear locking, which considerably increases the plate stiffness as its thickness decreases. For the aforementioned conventional meshes, various techniques primarily emerged to circumvent the shear locking occurrence, such as reduced and selective integration. Recently, a few locking solutions have been proposed and studied for arbitrary polygonal meshes. In the context of academic research, the literature available provides codes for topology optimization using polygonal meshes mainly applied to membrane elements. Motivated by the same academic purposes and inspired by the latest scientific contributions, this work aims to provide an educational, concise, and user-friendly MATLAB® code for topology optimization of plates using polygonal isoparametric plate elements under the Reissner-Mindlin theory. The shear locking is naturally solved by imposing a generalization of an assumed strain field over the polygonal edges under Timoshenko's beam assumption. The novel extracts of the proposed code are explained, and possible variations for further user exploration are discussed. To properly illustrate and validate the computational efficiency of the proposed code, some plate benchmark problems are solved in the context of compliance topology optimization.*

Keywords: *Topology Optimization, Polygonal Elements, Plates, Reissner-Mindlin, Compliance Minimization.*

1 INTRODUCTION

Arbitrary polygonal meshes are composed by polygons that best fit into the domain geometry in terms of an unstructured mesh, but with an attempt to standardize the element characteristic sizes. These meshes have been emerging in several contexts using numerical approximation methods, such as in the finite element method, due to their ability to be adjusted into complex domain boundaries. For membrane elements, there exist uncountable applications and educational codes available for both arbitrary polygonal and regular meshes (linear quadrilateral and triangular). Topology optimization is a widely known application that can use the finite element method in order to compute objective functions, sensitivities and constraints. In the work of Talischi *et al.* (2012b), the combination of SIMP topology optimization method (Bendsøe and Sigmund, 2002) with an arbitrary polygonal mesh (Talischi *et al.*, 2012a) is proposed for membrane finite elements through an educational code. Afterwards, many other contributions were developed in related areas using this code as a basis, such as for fluid flow (Pereira *et al.*, 2016), multi-material bodies (Sanders *et al.*, 2018), dynamic loads (Giraldo-Londoño and Paulino, 2021a) and stress-constrained optimization (Giraldo-Londoño and Paulino, 2021b), showing advantageous evidences in disclosing accessible codes.

Considering plate element meshes, the aforementioned conventional quadrilaterals and triangles have been widely studied along with finite element method applications – which includes, again, the topology optimization of plates and flat-element shells. For these conventional elements, in which the three nodal degrees of freedom are stated as the rotations around x and y axes plus a deflection in z axis, the phenomenon of shear locking is encountered. Shear locking is an underestimated displacement response of the plate when decreasing its thickness, and it arises from the integration of the shear terms. Even if a software has its formulation based on a single plate theory (generally, Reissner-Mindlin for thick plates or Kirchhoff for thin plates (Zienkiewicz *et al.*, 2013)), there is an effort in the literature to build robust tools that

are accurately able to address both thickness cases. For the conventional elements, several solutions were proposed, for instance, the reduced and selective integration (Zienkiewicz *et al.*, 1971; Hughes *et al.*, 1978), heterosis element (Hughes and Cohen, 1978), second-order elements, just to mention a few (for more details, see the book of Zienkiewicz *et al.* (2013)). However, none of these classical techniques solve the shear locking in arbitrary polygonal meshes.

Consequently, over the past years, some methods started emerging to circumvent locking in polygonal meshes. Nguyen-Xuan (2017) proposed a path-breaking technique based on a generalization of an assumed strain field imposed over the polygonal edges under Timoshenko's beam assumption (Soh *et al.*, 1999), which naturally solves the shear locking drawback. In this work, many shape functions were tested and a piecewise-linear function was proposed. This method has also been applied to laminated composite plates by Nguyen *et al.* (2017). Later, Videla *et al.* (2019) utilized an approach based on the discrete Kirchhoff Mindlin theory and the assumed shear strain fields with Wachspress shape functions (Floater *et al.*, 2014). Meanwhile, Katili *et al.* (2019) proposed a polygonal locking-free plate element for smoothed finite element method, while Wu *et al.* (2021) addressed the problem for polygonal hybrid displacement-function element method. More recently, Nguyen *et al.* (2023) adapted the first proposal technique (Nguyen-Xuan, 2017) by imposing a factor α (whose appropriate value was chosen to be 0.5) onto the assumed rotations and shear strains. Right after, Nguyen and Phan (2023) proposed a readjustment on the method by Nguyen *et al.* (2023) to include a selective element domain interpolation.

Although the polygonal meshes for thick-thin plate elements are still being studied, these techniques are proven to be properly used throughout finite element derivations, such as in topology optimization approaches (Nguyen *et al.*, 2022; Pham and Phan, 2022). Therefore, in this paper, we provide an educational MATLAB® code for topology optimization of thick-thin plates using polygonal isoparametric plate elements under the Reissner-Mindlin theory. The basis code is the `PolyTop`, available in the work of Talischi *et al.* (2012b). The aimed modifications are mainly executed in two major steps: converting the membrane element into a plate element and, then, implementing the assumed strain field on polygonal edges under Timoshenko's beam assumption (Nguyen-Xuan, 2017). For simplicity purposes, the shape functions are chosen to remain the same as in `PolyTop`, i.e. the Wachspress functions. This will represent the PRMn-W element in the work of Nguyen-Xuan (2017) and one can easily verify that converging differences between the present element and PRMn-PL, as proposed in the referred work, will not consistently affect the topology optimization results. Some examples in the context of compliance minimization are performed, guiding the reader throughout the modifications that must be done in the `PolyTop` code. Results show that the technique is efficient and simple to implement.

The present paper is organized as follows. Section 2 defines the mechanical and optimization problems, depicting the selected methods. The code modifications are explained in Section 3, followed by Section 4 with some numerical examples to test and validate the proposed scheme. Finally, conclusion remarks are discussed in Section 5. Acknowledgements and references are available in Sections 6 and 7, respectively.

2 PROBLEM STATEMENT

2.1 Reissner-Mindlin Plate Theory

Let $\Omega \subset \mathbb{R}^2$ be a bounded domain defined as the midplane of an isotropic Reissner-Mindlin plate. The governing equations for this model are:

$$\boldsymbol{\nabla} \cdot \mathbf{D^b} \boldsymbol{\kappa}(\boldsymbol{\theta}) + Gh\boldsymbol{\gamma} = \mathbf{0} \qquad \text{in } \Omega, \tag{1}$$

$$Gh\boldsymbol{\nabla} \cdot \boldsymbol{\gamma} = p \qquad \text{in } \Omega, \tag{2}$$

$$w = \overline{w}, \quad \boldsymbol{\theta} = \overline{\boldsymbol{\theta}} \qquad \text{on } \partial\Omega, \tag{3}$$

where w and $\boldsymbol{\theta}^\top = (\theta_x, \theta_y)$ are the transversal displacement and the rotations about y and x axes, respectively (see Fig. 1). The plate is under the transversal load $p(x, y)$ per unit area, $G = \frac{\mu E}{2(1+\nu)}$ is the shear modulus, with $\mu = 5/6$ being the shear correction factor, E is the Young's modulus and ν is the Poisson ratio. The bending ($\mathbf{D^b}$) and shear ($\mathbf{D^s}$) matrices are defined as:

$$\mathbf{D^b} = \frac{Eh^3}{2(1+\nu)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix}; \qquad \mathbf{D^s} = Gh \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{4}$$

where h is the plate thickness. The bending ($\boldsymbol{\kappa}$) and shear ($\boldsymbol{\gamma}$) strains are:

$$\boldsymbol{\kappa} = \frac{1}{2}\left(\boldsymbol{\nabla}\boldsymbol{\theta} + \boldsymbol{\nabla}\boldsymbol{\theta}^\top\right); \qquad \boldsymbol{\gamma} = \boldsymbol{\nabla} w - \boldsymbol{\theta}, \tag{5}$$

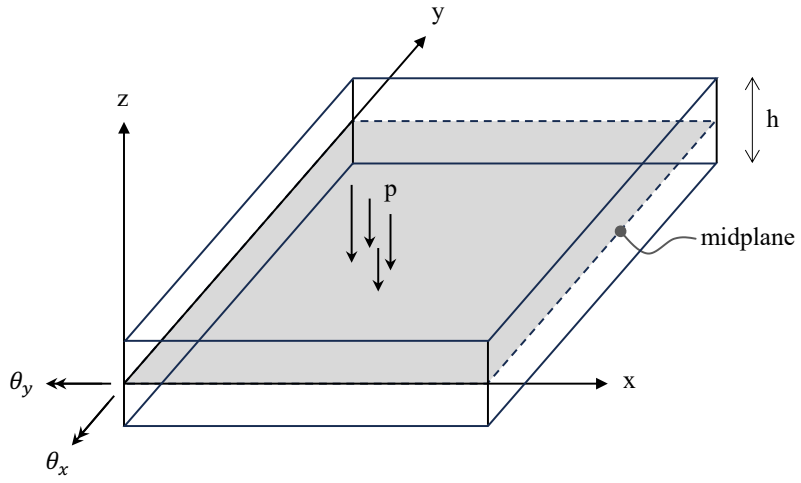


Figure 1. Reissner-Mindlin plate model.

where $\boldsymbol{\nabla} = (\partial/\partial x, \partial/\partial y)^\top$ is the gradient operator. Let us define V and V_0 as:

$$V = \{(w, \boldsymbol{\theta}) : w \in H^1(\Omega), \boldsymbol{\theta} \in H^1(\Omega^2); w = \overline{w}, \boldsymbol{\theta} = \overline{\boldsymbol{\theta}} \text{ on } \partial\Omega\}, \tag{6}$$

$$V_0 = \{(\partial w, \partial\boldsymbol{\theta}) \in V; \partial w = 0, \partial\boldsymbol{\theta} = \mathbf{0} \text{ on } \partial\Omega\}, \tag{7}$$

where $H^1(\Omega)$ is a Hilbert space. The discrete weak form can be obtained as: Find a discrete solution $(w^h, \boldsymbol{\theta}^h) \in V^h$, such that:

$$\int_\Omega \partial\boldsymbol{\kappa}^{h\top} \mathbf{D}^{\mathbf{b}} \boldsymbol{\kappa}^h \, d\Omega + \int_\Omega \partial\boldsymbol{\gamma}^{h\top} \mathbf{D}^{\mathbf{s}} \boldsymbol{\gamma}^h \, d\Omega = \int_\Omega \partial w p \, d\Omega, \quad \forall(\partial w^h, \partial\boldsymbol{\theta}^h) \in V_0^h, \tag{8}$$

in which $V^h \subset V$ and $V_0^h \subset V_0$ are the finite element approximation spaces. Finally, we define $\mathbf{u}^\top = (w, \boldsymbol{\theta})$ as the nodal displacement vector.

2.2 Finite Element Formulation and Locking-Free Technique

The displacement field within each finite element can be stated as:

$$\mathbf{u}^{\mathbf{e}}(x, y) = \sum_i \mathbf{N}_{\mathbf{i}}^{\mathbf{e}}(x, y) \mathbf{u}_{\mathbf{i}}^{\mathbf{e}}, \tag{9}$$

where i is the index for the number of polygonal vertices, $\mathbf{u}^{\mathbf{e}}(x, y)$ is the displacement vector at (x, y), $\mathbf{N}_{\mathbf{i}}^{\mathbf{e}}(x, y) = \mathbf{I}_{3\times3} N_i^e(x, y)$ are the shape functions and $\mathbf{u}_{\mathbf{i}}^{\mathbf{e}} = (w_i^e, \theta_{xi}^e, \theta_{yi}^e)^\top$ is the i^{th} nodal displacement vector of element e. Therefore, we will be able to write:

$$\boldsymbol{\kappa}^e = \sum_i \mathbf{B}_{\mathbf{i}}^{\mathbf{b},\mathbf{e}} \mathbf{u}_{\mathbf{i}}^{\mathbf{e}}; \qquad \boldsymbol{\gamma}^e = \sum_i \mathbf{B}_{\mathbf{i}}^{\mathbf{s},\mathbf{e}} \mathbf{u}_{\mathbf{i}}^{\mathbf{e}}, \tag{10}$$

where:

$$\mathbf{B}_{\mathbf{i}}^{\mathbf{b},\mathbf{e}} = \begin{bmatrix} 0 & \frac{\partial N_i}{\partial x} & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial y} \\ 0 & \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} \end{bmatrix}; \qquad \mathbf{B}_{\mathbf{i}}^{\mathbf{s},\mathbf{e}} = \begin{bmatrix} \frac{\partial N_i}{\partial x} & N_i & 0 \\ \frac{\partial N_i}{\partial x} & 0 & N_i \end{bmatrix}. \tag{11}$$

Finally, we have:

$$\mathbf{KU} = \mathbf{F}, \tag{12}$$

where:

$$\mathbf{K} = \sum_{e=1}^{n_e} \left[\int_{\Omega^e} \mathbf{B}^{\mathbf{b},\mathbf{e}\top} \mathbf{D}^{\mathbf{b}} \mathbf{B}^{\mathbf{b},\mathbf{e}} \, d\Omega^e + \int_{\Omega^e} \mathbf{B}^{\mathbf{s},\mathbf{e}\top} \mathbf{D}^{\mathbf{s}} \mathbf{B}^{\mathbf{s},\mathbf{e}} \, d\Omega^e \right] \tag{13}$$

$$\mathbf{F} = \sum_{e=1}^{n_e} \int_{\Omega^e} pN^e \, d\Omega^e + \mathbf{F}_{\mathbf{b}}, \tag{14}$$

where $\sum_{e=1}^{n_e}$ denotes the assembly procedure and $\mathbf{F}_{\mathbf{b}}$ is the prescribed boundary load vector.

The locking-free technique proposed by Nguyen-Xuan (2017) turns Eq. (13) into the following:

$$\mathbf{K} = \sum_{e=1}^{n_e} \left[\int_{\Omega^e} \left(\mathbf{B}^{\mathbf{b},\mathbf{e}} + \hat{\mathbf{B}}^{\mathbf{b},\mathbf{e}} \right)^{\top} \mathbf{D}^{\mathbf{b}} \left(\mathbf{B}^{\mathbf{b},\mathbf{e}} + \hat{\mathbf{B}}^{\mathbf{b},\mathbf{e}} \right) d\Omega^e + \int_{\Omega^e} \tilde{\mathbf{B}}^{\mathbf{s},\mathbf{e}\top} \mathbf{D}^{\mathbf{s}} \tilde{\mathbf{B}}^{\mathbf{s},\mathbf{e}} d\Omega^e \right]. \tag{15}$$

The reader is referred to the work of Nguyen-Xuan (2017) for a more detailed description about $\hat{\mathbf{B}}^{\mathbf{b},\mathbf{e}}$ and $\tilde{\mathbf{B}}^{\mathbf{s},\mathbf{e}}$.

2.3 Topology Optimization Problem

Introducing the relative density design variable ρ^e, Eq. (15) becomes:

$$\mathbf{K} = \sum_{e=1}^{n_e} (\rho^e)^{\mathfrak{p}} \left[\int_{\Omega^e} \left(\mathbf{B}^{\mathbf{b},\mathbf{e}} + \hat{\mathbf{B}}^{\mathbf{b},\mathbf{e}} \right)^{\top} \mathbf{D}^{\mathbf{b}} \left(\mathbf{B}^{\mathbf{b},\mathbf{e}} + \hat{\mathbf{B}}^{\mathbf{b},\mathbf{e}} \right) d\Omega^e + \int_{\Omega^e} \tilde{\mathbf{B}}^{\mathbf{s},\mathbf{e}\top} \mathbf{D}^{\mathbf{s}} \tilde{\mathbf{B}}^{\mathbf{s},\mathbf{e}} d\Omega^e \right], \tag{16}$$

where $\mathfrak{p}$ is the material penalization from SIMP (Bendsøe and Sigmund, 2002). The topology optimization problem in the context of compliance minimization is stated as:

$$\min \quad \mathbf{F}^{\top}\mathbf{U}, \tag{17}$$

$$\text{s.t.} \quad \begin{cases} \mathbf{K}\mathbf{U} = \mathbf{F}, \\ \dfrac{\sum_{e=1}^{n_e} \rho^e V^e}{\sum_{e=1}^{n_e} 1^e V^e} - \overline{V} \leq 0, \\ \underline{\rho} \leq \rho^e \leq \overline{\rho}, \; e = 1, \ldots, n_e, \end{cases} \tag{18}$$

where $\underline{\rho}$ and $\overline{\rho}$ are the ρ^e lower and upper bounds, respectively, V^e is the element volume and $\overline{V}$ is the volume constraint value. Other aspects, such as meshing, filtering, sensitivity analysis and optimization method, are the same as in the work of Talischi *et al.* (2012b), except for the continuation method, which will be replaced by a constant penalization value of $\mathfrak{p} = 3$.

3 ALGORITHM & CODE MODIFICATIONS

In this section, we present the main code modifications on the original `PolyTop` code provided by Talischi *et al.* (2012b). (Note: as the MATLAB® software has been updated several times since the publication of the original `PolyTop` article, an updated version of this code has been made available on the website of one of authors (Paulino, 2023), under the name of `PolyTop v1.1`, which will be the software considered hereinafter).

We also note that, since the following modifications will constantly change the line numbers, these numbers will be always updated accordingly to the previous changes already made. Therefore, the reader must strictly follow the sequence proposed below.

3.1 `PolyScript.m`

Replace line 19 by the following lines:

```
'h',0.1,...                 % Plate Thickness
'kappa',5/6,...               % Shear correction factor (rectang. section)
'Reg',0 ...                   % Tag for regular meshes
```

Delete lines 44 and 47, and replace line 43 by:

```
penal = 3;
```

3.2 PolyTop.m

Replace line 63 by:

```
fem.ElemNDof = 3*cellfun(@length,fem.Element); % # of DOFs per element
```

Replace line 71 by:

```
  eDof = reshape([3*fem.Element{el}-2;3*fem.Element{el}-1;3*fem.Element{el}],NDof,1);
```

Replace lines 80 to 82 by the following lines:

```
fem.F = zeros(3*fem.NNode,1);  %external load vector
fem.F(3*fem.Load(1:NLoad,1)-2) = fem.Load(1:NLoad,2);  %w-crdnt
fem.F(3*fem.Load(1:NLoad,1)-1)   = fem.Load(1:NLoad,3);  %thetax-crdnt
fem.F(3*fem.Load(1:NLoad,1))   = fem.Load(1:NLoad,4);  %thetay-crdnt
```

Replace lines 85 and 86 by:

```
fixedDofs = [fem.Supp(1:NSupp,2).*(3*fem.Supp(1:NSupp,1)-2);
             fem.Supp(1:NSupp,3).*(3*fem.Supp(1:NSupp,1)-1);
             fem.Supp(1:NSupp,4).*(3*fem.Supp(1:NSupp,1))];
```

Replace lines 89 and 94 respectively by:

```
AllDofs   = 1:3*fem.NNode;
U = zeros(3*fem.NNode,1);
```

Replace line 99 by:

```
G = fem.E0/(2*(1+fem.Nu0)); Ds = [G 0;0 G];
nn=length(eNode); Ke=zeros(3*nn,3*nn);
```

Insert the following line after line 104:

```
N = fem.ShapeFnc{nn}.N(:,:,q);
```

Replace lines 107 to 112 by:

```
Bb = zeros(3,3*nn);
Bb(1,2:3:3*nn) = dNdx(:,1)';
Bb(2,3:3:3*nn) = dNdx(:,2)';
Bb(3,2:3:3*nn) = dNdx(:,2)';
Bb(3,3:3:3*nn) = dNdx(:,1)';
[BbT,BsT] = TimoAssumed(fem,nn,N,dNdx,fem.Node(eNode,:));
Ke = Ke + fem.h^3/12*(Bb+BbT)'*D*(Bb+BbT)*W(q)*det(J0) + ...
    fem.kappa*fem.h*BsT'*Ds*BsT*W(q)*det(J0);
```

After line 116, insert the following lines to add the locking-free technique by Nguyen-Xuan (2017):

```
%-------------------------------------- TIMOSHENKO LOCKING-FREE MATRICES
function [BbT,BsT] = TimoAssumed(fem,nn,N,dNdx,xy)
jkm = [2:nn 1; 3:nn 1 2; nn 1:nn-1]';
c = xy(jkm(:,2),1)-xy(jkm(:,1),1); b = xy(jkm(:,1),2)-xy(jkm(:,2),2);
l = sqrt(c.^2 + b.^2);
chi = (fem.h./l).^2./(2*(fem.h./l).^2+fem.kappa*(1-fem.Nu0));
Ib = zeros(nn); Is = Ib; G = zeros(nn,3*nn);
Hb = zeros(3,nn); Hs = zeros(2,nn);
for i = 1:nn
    j = jkm(i,1); k = jkm(i,2); m = jkm(i,3);
```

```
    Ib(i,i) = 1-2*chi(i);
    Is(i,i) = chi(i);
    G(i,3*j-2:3*j) = [-2 c(i) -b(i)];
    G(i,3*k-2:3*k) = [2 c(i) -b(i)];
    Hb(:,i) = 3/l(i)^2*[c(i)*(dNdx(j,1)*N(k)+dNdx(k,1)*N(j));
                        -b(i)*(dNdx(j,2)*N(k)+dNdx(k,2)*N(j));
                         (c(i)*dNdx(k,2)-b(i)*dNdx(k,1))*N(j)+(c(i)*dNdx(j,2)-b(i)*dNdx(j,1))*N
                             (k)];
    Hs(:,i) = [b(m)/(c(i)*b(m)-c(m)*b(i))*N(j) - b(j)/(c(j)*b(i)-c(i)*b(j))*N(k);
               c(m)/(c(i)*b(m)-c(m)*b(i))*N(j) - c(j)/(c(j)*b(i)-c(i)*b(j))*N(k)];
end
BbT = Hb*Ib*G; BsT = Hs*Is*G;
```

3.3 New domain examples

The domains for the numerical examples in the next section are defined herein.

3.3 .1 `SquarePlateDomain.m`

Copy the original `MbbDomain.m` content, paste into the new `SquarePlateDomain.m` file and adapt according to the following instructions.

Replace lines 6 and 7 by:

```
function [x] = SquarePlateDomain(Demand,Arg)
  BdBox = [0 0.5 0 0.5];
```

For a simply supported plate, replace lines 20 to 28 by:

```
  LeftEdgeNodes = find(abs(Node(:,1)-BdBox(1))<eps);   %x=0 (y-parallel)
  RightEdgeNodes = find(abs(Node(:,1)-BdBox(2))<eps);  %x=Lx (y-parallel)
  BottomEdgeNodes = find(abs(Node(:,2)-BdBox(3))<eps); %y=0 (x-parallel)
  UpperEdgeNodes = find(abs(Node(:,2)-BdBox(4))<eps);  %y=Ly (x-parallel)
  FixedNodes = [LeftEdgeNodes;BottomEdgeNodes;RightEdgeNodes;UpperEdgeNodes];
  Supp = zeros(length(FixedNodes),4); Supp(:,1)=FixedNodes;
  div = length([LeftEdgeNodes;BottomEdgeNodes]);
  Supp(1:div,2)=1;                            % w = 0 at left & bottom edges
  Supp(1:length(LeftEdgeNodes),4)=1;          % at left edge, theta_y = 0
  Supp(length(LeftEdgeNodes)+1:div,3)=1;      % at bottom edge, theta_x = 0
  Supp(div+1:div+length(RightEdgeNodes),3)=1; % in y-parallel sym., theta_x = 0
  Supp(div+length(RightEdgeNodes)+1:end,4)=1; % in x-parallel sym., theta_y = 0
  UpperRightNode = find(abs(Node(:,1)-BdBox(2))<eps & abs(Node(:,2)-BdBox(4))<eps);
  Load = [UpperRightNode,-1/4,0,0];
```

After implementing for simply supported plates, the reader may adapt the domain for a clamped plate by replacing lines 26 to 31 by:

```
  div1 = length([LeftEdgeNodes;BottomEdgeNodes]);
  div2 = length([LeftEdgeNodes;BottomEdgeNodes;RightEdgeNodes]);
  Supp(1:div1,2:4)=1;    % all dof fixed at left & bottom edges
  Supp(div1+1:div2,3)=1; % in y-parallel sym., theta_x = 0
  Supp(div2+1:end,4)=1;  % in x-parallel sym., theta_y = 0
```

3.3 .2 `HookPlateDomain.m`

Similarly, copy the original `HookDomain.m` content, paste into `HookPlateDomain.m` file and then replace lines 33 to 40 by:

```
  UpperCircleNodes = find(abs(sqrt(Node(:,1).^2+(Node(:,2)...
                                -80.6226).^2)-10)<eps);
  Supp = ones(size(UpperCircleNodes,1),4);
```

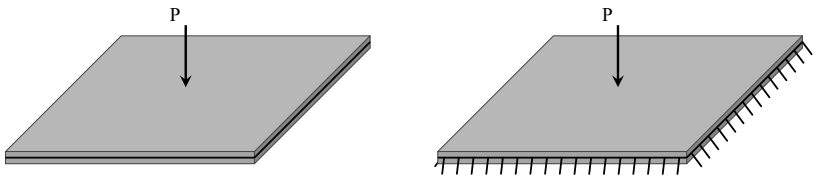
```
Supp(:,1) = UpperCircleNodes;
EndHookNode = find(abs(Node(:,1)+27.0406)<eps & abs(Node(:,2)−8.0406)<eps);
while isempty(EndHookNode)
    eps=eps∗1.1;
    EndHookNode = find(abs(Node(:,1)+27.0406)<5∗eps & abs(Node(:,2)−8.0406)<5∗eps);
end
EndHookNode = EndHookNode(find(Node(EndHookNode,2)==max(Node(EndHookNode,2))));
Load = [EndHookNode,−1,0,0];
```

4 NUMERICAL EXAMPLES

4.1 Square Plates

The square plates are classical benchmark problems in plate bending. In this paper, we will demonstrate the code efficiency by running topology optimization of simply supported and clamped plates, both on thick and thin cases. This example has been studied by Pham and Phan (2022) and Long *et al.* (2009). The 1x1 plate has a unit load acting on its center – see Figs. 2(a) and 2(b). The Young's modulus is $E = 1092000$ and the Poisson ration 0.3. The number of elements is set as 16000, the filter radius, 0.0135, and the constraint volume fraction is 50% of the initial volume. The Dirichlet boundary conditions apply for every plate side, i.e. all sides will be either simply supported or clamped, depending on each case. For thick approach, we set $h = 0.1$, and, for thin case, $h = 0.001$. Since the domain is symmetric, only one quarter of the plate will be considered in the optimization process.

Results are shown in Figs. 3 and 4, where the whole plates are assembled with the four symmetric pieces. As we can see, the optimal topologies become very different when the thickness changes.



(a) Simply-supported plate (b) Clamped plate

Figure 2. Square Plate Boundary Conditions.

4.2 A Hook Plate

The hook example has been studied as a plate problem by Pham and Phan (2022) and is an appropriate geometry to take advantage of the polygonal meshes' power. The domain is illustrated in Fig. 5(a), where the eyelet circle is clamped and a transversal unit force acts on the hook end. The parameters remain the same of the previous examples, except for filter radius (2.0) and volume fraction (27.5%). Thickness for the thick case is $h = 25$, while $h = 0.01$ for thin case.

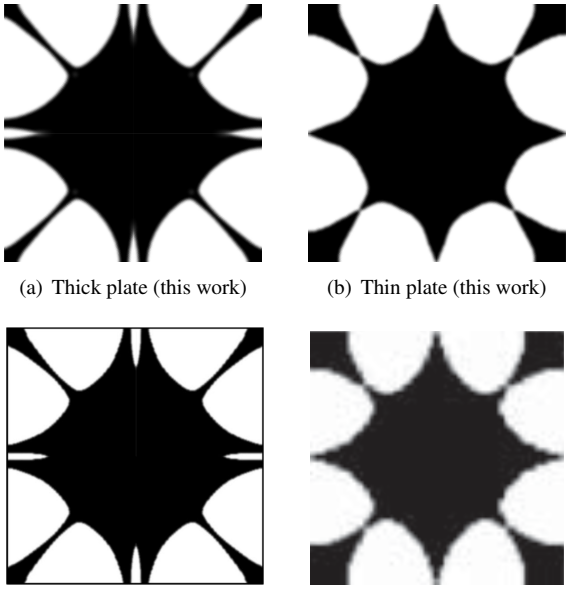
Figure 5 demonstrates the results for thick and thin hook topology optimization. Again, the results differ from each other for distinct values of plate thickness.

5 CONCLUSIONS

This paper provides simple steps to modify a well-known polygonal topology optimization code in order to enable the proper utilization in both thick and thin plate approaches. Since the shear locking problem has also been shown to be present in polygonal meshes, a solution is incorporated into the original code besides the plate model changes. Some examples were generated so as the reader may test and validate if implementations are correctly working. Future works encompass any of the various applications regarding the use of plates in the context of topology optimization using arbitrary polygonal meshes.

6 ACKNOWLEDGEMENTS

The authors acknowledge the financial support provided by CNPq and "Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil" (CAPES).



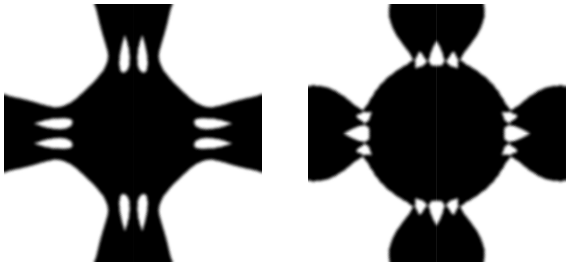
(a) Thick plate (this work)

(b) Thin plate (this work)

(c) Thick plate from Pham and Phan (2022)

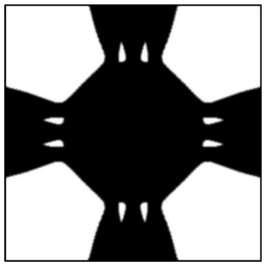
(d) Thin plate from Long *et al.* (2009)

Figure 3. Simply-Supported Square Plate Topology Optimization Results.

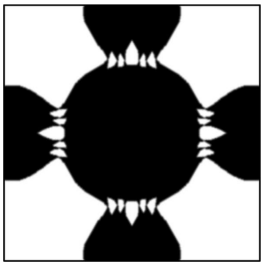


(a) Thick plate (this work)

(b) Thin plate (this work)



(c) Thick plate from Pham and Phan (2022)



(d) Thin plate from Pham and Phan (2022)

Figure 4. Clamped Square Plate Topology Optimization Results.

7 REFERENCES

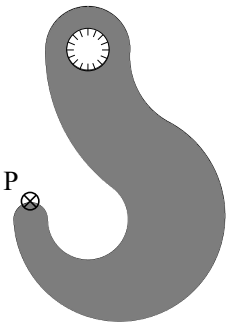
Bendsøe, M.P. and Sigmund, O., 2002. *Topology Optimization - Theory, Methods, and Applications*. Springer Berlin, Heidelberg.

Floater, M., Gillette, A. and Sukumar, N., 2014. "Gradient bounds for wachspress coordinates on polytopes". *SIAM J Numer Anal*, Vol. 52, pp. 515–532.

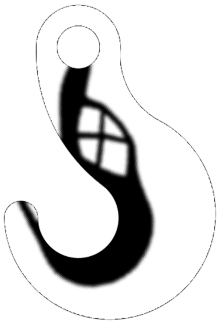
Giraldo-Londoño, O. and Paulino, G.H., 2021a. "`PolyDyna`: a Matlab implementation for topology optimization of structures subjected to dynamic loads". *Struct Multidisc Optim*, Vol. 64, p. 957–990.

Giraldo-Londoño, O. and Paulino, G.H., 2021b. "`PolyStress`: a Matlab implementation for local stress-constrained topology optimization using the augmented Lagrangian method". *Struct Multidisc Optim*, Vol. 63, p. 2065–2097.

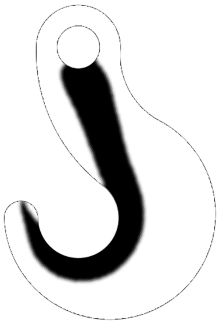
Hughes, T.J.R. and Cohen, M., 1978. "The heterosis finite element for plate bending". *Comput Struct*, Vol. 9, pp. 445–450.



(a) Hook plate model



(b) Thick hook plate (this work)



(c) Thin hook plate (this work)

(d) Thick hook plate from Pham and Phan (2022)

(e) Thin hook plate from Pham and Phan (2022)

Figure 5. Hook Plate Topology Optimization Results.

Hughes, T.J.R., Cohen, M. and Haroun, M., 1978. "Reduced and selective integration techniques in finite element analysis of plates". *Nucl Eng Des*, Vol. 46, pp. 203–222.

Katili, I., Maknun, I.J., Katili, A.M., Bordas, S.P.A. and Natarajan, S., 2019. "A unified polygonal locking-free thin/thick smoothed plate element". *Composite Structures*, Vol. 219, pp. 147–157.

Long, C.S., Loveday, P.W. and Groenwold, A.A., 2009. "Effects of finite element formulation on optimal plate and shell structural topologies". *Finite Elem Anal Des*, Vol. 45, p. 817–825.

Nguyen, N.V., Nguyen, H.X., Phan, D. and Nguyen-Xuan, H., 2017. "A polygonal finite element method for laminated composite plates". *Int J Mech Sci*, Vol. 133, pp. 863–882.

Nguyen, N.V., Nguyen-Xuan, H. and Lee, J., 2022. "Polygonal composite elements for stress-constrained topology optimization of nearly incompressible materials". *European Journal of Mechanics - A/Solids*, Vol. 94, p. 104548.

Nguyen, S.H., Nam, N.N., Hoang, T., Nguyen, T.N. and Nguyen-Thoi, T., 2023. "Alpha (α) assumed rotations and shear strains for spatially isotropic polygonal Reissner-Mindlin plate elements (αARS-poly)". *Computers & Structures*, Vol. 274.

Nguyen, S.H. and Phan, D., 2023. "Selective element domain interpolation technique for assumed rotations and shear strains in polygonal finite element thick/thin plate analysis". *Thin-Walled Structures*, Vol. 186.

Nguyen-Xuan, H., 2017. "A polygonal finite element method for plate analysis". *Computers & Structures*, Vol. 188, pp. 45–62.

Paulino, G.H., 2023. "Glaucio H. Paulino's website, Software". Glaucio H. Paulino's website, Princeton University, http://paulino.princeton.edu/software.html. Accessed 20 May 2023.

Pereira, A., Talischi, C., Paulino, G.H., Menezes, I. and Carvalho, M.S., 2016. "Fluid flow topology optimization in `PolyTop`: stability and computational implementation". *Struct Multidisc Optim*, Vol. 54, p. 1345–1364.

Pham, Q.H. and Phan, D.H., 2022. "Polygonal topology optimization for Reissner–Mindlin plates". *Engineering with Computers*, Vol. 38.

Sanders, E.D., Pereira, A., Aguiló, M.A. and Paulino, G.H., 2018. "`PolyMat`: an efficient Matlab code for multi-material topology optimization". *Struct Multidisc Optim*, Vol. 58, p. 2727–2759.

Soh, A.K., Long, Z.F. and Cen, S., 1999. "A new nine dof triangular element for analysis of thick and thin plates". *Comput Mech*, Vol. 24, pp. 408–417.

Talischi, C., Paulino, G.H., Pereira, A. and Menezes, I., 2012a. "`PolyMesher`: a general-purpose mesh generator for polygonal elements written in Matlab". *Struct Multidisc Optim*, Vol. 45, p. 309–328.

Talischi, C., Paulino, G.H., Pereira, A. and Menezes, I., 2012b. "`PolyTop`: a Matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes". *Struct Multidisc Optim*, Vol. 45, p. 329–357.

Videla, J., Natarajan, S. and Bordas, S.P.A., 2019. "A new locking-free polygonal plate element for thin and thick plates based on Reissner-Mindlin plate theory and assumed shear strain fields". *Computers & Structures*, Vol. 220, pp. 32–42.

Wu, C., Cen, S. and Shang, Y., 2021. "Shape-free polygonal hybrid displacement-function element method for analyses of Mindlin– Reissner plates". *Engineering with Computers*, Vol. 37, p. 1975–1998.

Zienkiewicz, O.C., Taylor, R.L. and Zhu, J.Z., 2013. *The Finite Element Method: Its Basis and Fundamentals*. Butterworth-Heinemann.

Zienkiewicz, O.C., Too, J. and Taylor, R.L., 1971. "Reduced integration technique in general analysis of plates and shells". *Int J Num Methods Eng*, Vol. 3, pp. 275–290.

8 RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.