Fast and robust Matlab-based finite element model used in the layup optimization of composite laminates

A Şerban[1]

[1]Mechanical Engineering, "Gheorghe Asachi" Technical University of Iasi, Iasi, Romania

E-mail: ser.alexandru@gmail.com

Abstract. The layup optimization of the laminated composites is a very complex topic which involves a convoluted solution space usually explored using heuristic computational techniques. Due to the solution space complexity a lot of layup configurations are evaluated during the optimization process. This fact leads to the mandatory requirement that the configuration evaluation should be fast enough to ensure the convergence of the optimization procedure without sacrificing the accuracy. In this work, we propose a robust, accurate and very fast finite element model based on the first-order shear deformation theory (FSDT). The model is structured in three main parts: preprocessing, processing and post processing. The main strategy is to transfer as much as possible operations in the preprocessing phase which is executed only once and to subsequently reuse the results in the processing and post processing phases which are executed for each layup configuration. Using this strategy, the execution time of the processing and post processing phases is drastically reduced and almost consists of regenerating and solving the global linear system – more that 95%. The proposed procedure is relatively easy to implement in Matlab which holds a very powerful linear system solver for sparse matrices. Also, the accuracy of the model was demonstrated by comparison with Ansys and with some closed form solutions.

Introduction

The physical properties of composite materials are highly influenced by the fibers orientation. This characteristic becomes critical when we bring into question composite laminates materials built from unidirectional or bidirectional layers, which exhibit an orthotropic behaviour.

The layup optimization is a very complex problem involving a huge solution space usually explored using heuristic computational techniques of which the most reliable are the genetic algorithms [1-8]. During the optimization process a lot of layup configurations are evaluated in order to converge to the global optimum solution. This fact leads to the mandatory requirement that the evaluation of a single layup configuration should be fast enough to ensure convergence in an acceptable time range.

On the other hand, the mechanical behaviour of the layered composite laminates is very complex due to the orthotropic particularity and its analysis involves numerical tools from which the most performant is the finite element method (FEM). A lot of optimization studies are using commercial software packages [9-15] for mechanical behaviour simulation which may be time consuming.

Because FEM is a time expensive analysis, a big part of the optimization algorithms are based on analytical closed form models applicable only for simple geometries (like rectangle) and simple stress distributions [16-19].

In this respect we developed a very fast and robust finite element model which can be used either in the optimization of stacking sequence or in the mechanical behaviour analysis of a certain layup configuration. The model in based on the first-order shear deformation theory (FSDT) and it is structured such that most as possible computations are executed only once in a preprocessing phase and the results are reused in the processing and post processing phases which are individually executed for all the layup configurations evaluated during the optimization process.

The accuracy of the model is proved by comparison with Ansys and some theoretical results. Also, a detailed execution time analysis is provided in order to show that for any substantial speed improvement it should be addressed complex computation techniques and numerical methods which exceed our purpose.

Mathematical model

The mathematical model is based on *first-order shear deformation theory* (FSDT). This is a two-dimensional theory derived from three-dimensional elasticity theory by making two important assumptions regarding the variation of displacements through the thickness of the laminate. According to these assumptions – coming from Kirchhoff-Love hypothesis – straight lines perpendicular to the midplane before deformation remain (a) *straight* and (b) *inextensible*. It is also assumed that the strains are continuous through the thickness which allows the replacement of a multilayered laminate with an equivalent single layer.

Even if the mathematical model is based on FSDT, the strategies used for speeding up the FEA of laminated composites easily extend to any two-dimensional theory.

According to FSDT the displacements $(\Delta_x, \Delta_y, \Delta_z)$ have the form:

$$\begin{cases}\Delta_x(x,y,z,t) = \Delta_x^m(x,y,t) + z \cdot \phi_y(x,y,t)\\ \Delta_y(x,y,z,t) = \Delta_y^m(x,y,t) + z \cdot \phi_x(x,y,t)\\ \Delta_z(x,y,z,t) = \Delta_z^m(x,y,t)\end{cases} \tag{1}$$

Where $(\Delta_x^m, \Delta_y^m, \Delta_z^m, \phi_x, \phi_y)$ represents the midplane correspondent point displacements and rotations around O_x and O_y axes.

The finite element model based on FSDT has the compact form:

$$[K^e]\{\Delta^e\} - \{F^e\} = \{0\} \tag{2}$$

Where (K^e) represents the element stiffness matrix, (Δ^e) represents the element nodes displacements and rotation (degrees of freedom – *the unknowns*) and (F^e) represents the element force vector defined by Neumann boundary condition.

The expanded form of (2) is:

$$\sum_{\beta=1}^{5}\sum_{j=1}^{n} K_{ij}^{\alpha\beta}\Delta_j^\beta - F_i^\alpha = 0, \quad \left(\alpha = \overline{1,5}\ and\ i = \overline{1,n}\right) \tag{3}$$

where (n) is the number of element nodes.

Stiffness matrix blocks can be computed as follow:

$$K_{ij}^{1\alpha} = \int_{\Omega^e}\left(\frac{\partial\psi_i}{\partial x}N_{1j}^\alpha + \frac{\partial\psi_i}{\partial y}N_{6j}^\alpha\right)dx\,dy$$
$$K_{ij}^{2\alpha} = \int_{\Omega^e}\left(\frac{\partial\psi_i}{\partial x}N_{6j}^\alpha + \frac{\partial\psi_i}{\partial y}N_{2j}^\alpha\right)dx\,dy$$

$$\begin{aligned}
K_{ij}^{3\alpha} &= \int_{\Omega^e} \left(\frac{\partial \psi_i}{\partial x} Q_{1j}^{\alpha} + \frac{\partial \psi_i}{\partial y} Q_{2j}^{\alpha} \right) dx\, dy \\
K_{ij}^{4\alpha} &= \int_{\Omega^e} \left(\frac{\partial \psi_i}{\partial x} M_{1j}^{\alpha} + \frac{\partial \psi_i}{\partial y} M_{6j}^{\alpha} + \psi_i Q_{1j}^{\alpha} \right) dx\, dy \\
K_{ij}^{5\alpha} &= \int_{\Omega^e} \left(\frac{\partial \psi_i}{\partial x} M_{6j}^{\alpha} + \frac{\partial \psi_i}{\partial y} M_{2j}^{\alpha} + \psi_i Q_{2j}^{\alpha} \right) dx\, dy
\end{aligned} \tag{4}$$

(ψ_i) denotes the element *Lagrange* shape functions.
(N_{ij}^{α}) coefficients are defined as:

$$\begin{aligned}
N_{1j}^1 &= A_{11} \frac{\partial \psi_j}{\partial x} + A_{16} \frac{\partial \psi_j}{\partial y}, \quad N_{1j}^2 = A_{16} \frac{\partial \psi_j}{\partial x} + A_{12} \frac{\partial \psi_j}{\partial y} \\
N_{1j}^4 &= B_{11} \frac{\partial \psi_j}{\partial x} + B_{16} \frac{\partial \psi_j}{\partial y}, \quad N_{1j}^5 = B_{16} \frac{\partial \psi_j}{\partial x} + B_{12} \frac{\partial \psi_j}{\partial y} \\
N_{2j}^1 &= A_{12} \frac{\partial \psi_j}{\partial x} + A_{26} \frac{\partial \psi_j}{\partial y}, \quad N_{2j}^2 = A_{26} \frac{\partial \psi_j}{\partial x} + A_{22} \frac{\partial \psi_j}{\partial y} \\
N_{2j}^4 &= B_{12} \frac{\partial \psi_j}{\partial x} + B_{26} \frac{\partial \psi_j}{\partial y}, \quad N_{2j}^5 = B_{26} \frac{\partial \psi_j}{\partial x} + B_{22} \frac{\partial \psi_j}{\partial y} \\
N_{6j}^1 &= A_{16} \frac{\partial \psi_j}{\partial x} + A_{66} \frac{\partial \psi_j}{\partial y}, \quad N_{6j}^2 = A_{66} \frac{\partial \psi_j}{\partial x} + A_{26} \frac{\partial \psi_j}{\partial y} \\
N_{6j}^4 &= B_{16} \frac{\partial \psi_j}{\partial x} + B_{66} \frac{\partial \psi_j}{\partial y}, \quad N_{6j}^5 = B_{66} \frac{\partial \psi_j}{\partial x} + B_{26} \frac{\partial \psi_j}{\partial y}
\end{aligned} \tag{5}$$

(M_{ij}^{α}) coefficients are defined as:

$$\begin{aligned}
M_{1j}^1 &= B_{11} \frac{\partial \psi_j}{\partial x} + B_{16} \frac{\partial \psi_j}{\partial y}, \quad M_{1j}^2 = B_{16} \frac{\partial \psi_j}{\partial x} + B_{12} \frac{\partial \psi_j}{\partial y} \\
M_{1j}^4 &= D_{11} \frac{\partial \psi_j}{\partial x} + D_{16} \frac{\partial \psi_j}{\partial y}, \quad M_{1j}^5 = D_{16} \frac{\partial \psi_j}{\partial x} + D_{12} \frac{\partial \psi_j}{\partial y} \\
M_{2j}^1 &= B_{12} \frac{\partial \psi_j}{\partial x} + B_{26} \frac{\partial \psi_j}{\partial y}, \quad M_{2j}^2 = B_{26} \frac{\partial \psi_j}{\partial x} + B_{22} \frac{\partial \psi_j}{\partial y} \\
M_{2j}^4 &= D_{12} \frac{\partial \psi_j}{\partial x} + D_{26} \frac{\partial \psi_j}{\partial y}, \quad M_{2j}^5 = D_{26} \frac{\partial \psi_j}{\partial x} + D_{22} \frac{\partial \psi_j}{\partial y} \\
M_{6j}^1 &= B_{16} \frac{\partial \psi_j}{\partial x} + B_{66} \frac{\partial \psi_j}{\partial y}, \quad M_{6j}^2 = B_{66} \frac{\partial \psi_j}{\partial x} + B_{26} \frac{\partial \psi_j}{\partial y} \\
M_{6j}^4 &= D_{16} \frac{\partial \psi_j}{\partial x} + D_{66} \frac{\partial \psi_j}{\partial y}, \quad M_{6j}^5 = D_{66} \frac{\partial \psi_j}{\partial x} + D_{26} \frac{\partial \psi_j}{\partial y}
\end{aligned} \tag{6}$$

(Q_{ij}^{α}) coefficients are defined as:

$$\begin{aligned}
Q_{1j}^3 &= A_{55} \frac{\partial \psi_j}{\partial x} + A_{45} \frac{\partial \psi_j}{\partial y}, \quad Q_{2j}^3 = A_{45} \frac{\partial \psi_j}{\partial x} + A_{44} \frac{\partial \psi_j}{\partial y} \\
Q_{1j}^4 &= A_{55} \psi_j, \quad Q_{2j}^4 = A_{45} \psi_j \\
Q_{1j}^5 &= A_{45} \psi_j, \quad Q_{2j}^5 = A_{44} \psi_j
\end{aligned} \tag{7}$$

(A_{ij}, D_{ij}, B_{ij}) denotes *extensional stiffnesses, bending stiffnesses* and *bending-extensional coupling stiffnesses.*

Force vector blocks can be computed as follow:

$$\begin{aligned}
F_i^1 &= \int_{\Gamma^e} \psi_i\, N_n ds, \quad F_i^2 = \int_{\Gamma^e} \psi_i\, N_{ns} ds \\
F_i^3 &= \int_{\Gamma^e} \psi_i\, Q_n ds + \int_{\Omega^e} \psi_i\, q\, dx\, dy \\
F_i^4 &= \int_{\Gamma^e} \psi_i\, M_n ds, \quad F_i^5 = \int_{\Gamma^e} \psi_i\, M_{ns} ds
\end{aligned} \tag{8}$$

Where $(N_n, N_{ns}, M_n, M_{ns}, Q_n, q)$ represent the corresponding edge normal and tangential forces and moments, transverse force and transverse distributed load, respectively.

FE model structure

During the layup optimization of the composite laminates, a lot of layup configurations are evaluated in order to find the global optimal solution. Based on some observations the FE model can be structured such that the execution time required for the evaluation of a single configuration decreases dramatically. Our proposal refers to a structure which contains three major phases: *preprocessing, processing* and *post processing*. These phases are discussed in more details in the next pages.

1.1. Preprocessing

Regarding speeding up the FE model, the *preprocessing* phase is the most important. For a specific geometry subjected to layup optimization, the *preprocessing* phase is executed only once at the beginning of the optimization process and the results are reused in the other two phases - *processing* and *post processing,* which are executed for all layup configurations evaluated.

That being said, it is desirable to include as much as possible operations in the *preprocessing* in order to reduce the amount of calculations to be done in the *processing* and *post processing.*

Below are presented the operations included in the *preprocessing* phase.

A1 - Meshing – it is the most obvious operation which can be transferred to the *preprocessing* because the optimization process is carried out for specific (constant) geometry. Meshing involves the geometry discretization using 2D elements. In [20] is presented a robust meshing algorithm with triangular elements based on Delaunay triangulation. The algorithm can be very easily implemented in MATLAB thanks to the built-in function for computing the Delaunay triangulation. Also, it can be extended to quadrilateral elements by using indirect methods such those presented in [21-22]. After meshing the topology is described by a set of points representing the nodes and a connectivity matrix representing the elements of order I.

A2 - Adding auxiliary nodes - if higher order elements are required some additional nodes should be added. For example for a second order triangular element it should be added three auxiliary nodes representing the geometric place of the middle of each triangle edge. These auxiliary nodes are added to the end of the set of points obtained from meshing. Also, the connectivity matrix is updated accordingly with the auxiliary points.

A3 - Stiffness coefficients computation - for all the types of layers at all the orientations used in the optimization process. The stiffness coefficients are used afterwards in the *processing* phase to the computation of the *extensional stiffnesses* (A), *bending stiffnesses* (D) and *bending-extensional coupling stiffnesses* (B). If the layer's orientation is considered a continuous variable in the optimization process than the stiffness coefficients can be computed only for a representative discrete set of orientations and then interpolated in the *processing* phase.

A4 - Shape functions computation – for the nodes of all the elements.

A5 - Elemental force vectors computation. From equations (8) it can be observed that the elemental forces depends only of the elemental shape functions and Neumann boundary conditions which remains constant during the optimization process due to the fixed topology. In other words the elemental forces are independent from the optimization process variables: number of layers, types of layers and orientation of layers.

A6 - Global force vector assembly - based on the above observation (A5) the global force vector remains constant during the optimization process.

A7 - Elemental stiffness coefficients precalculating. For example, if we consider the stiffness matrix block:

$$K_{ij}^{5\alpha} = \int_{\Omega^e} \left(\frac{\partial \psi_i}{\partial x} M_{6j}^{\alpha} + \frac{\partial \psi_i}{\partial y} M_{2j}^{\alpha} + \psi_i Q_{2j}^{\alpha} \right) dx\, dy \tag{9}$$

For $\alpha = 5$:

$$\begin{aligned} K_{ij}^{55} &= \int_{\Omega^e} \left(\frac{\partial \psi_i}{\partial x} M_{6j}^{5} + \frac{\partial \psi_i}{\partial y} M_{2j}^{5} + \psi_i Q_{2j}^{5} \right) dx\, dy \\ M_{6j}^{5} &= D_{66} \frac{\partial \psi_j}{\partial x} + D_{26} \frac{\partial \psi_j}{\partial y} \\ M_{2j}^{5} &= D_{26} \frac{\partial \psi_j}{\partial x} + D_{22} \frac{\partial \psi_j}{\partial y} \\ Q_{2j}^{5} &= A_{44} \psi_j \end{aligned} \tag{10}$$

The integral associated with stiffness matrix block (K_{ij}^{55}) expands to:

$$I = K_{ij}^{55} = \int_{\Omega^e} (D_{26} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial y} + D_{66} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} + D_{26} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial x} + D_{22} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} + A_{44} \psi_i \psi_j)\, dx\, dy \tag{11}$$

Using the linearity property of the integration operation, the expanded integral (I) can be written as:

$$\begin{aligned} I = K_{ij}^{55} = {} & D_{26} \int_{\Omega^e} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial y}\, dx\, dy + D_{66} \int_{\Omega^e} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x}\, dx\, dy + \\ & D_{26} \int_{\Omega^e} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial x}\, dx\, dy + D_{22} \int_{\Omega^e} \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y}\, dx\, dy + \\ & A_{44} \int_{\Omega^e} \psi_i \psi_j\, dx\, dy \end{aligned}$$

$$I = K_{ij}^{55} = D_{26} \cdot I_1 + D_{66} \cdot I_2 + D_{26} \cdot I_3 + D_{22} \cdot I_4 + A_{44} \cdot I_5 \tag{12}$$

Integral (10) written in the form (12) is a linear combination between some *A* and *D* matrices coefficients and integrals (I_i) defined over the elemental subdomain with integrands dependent only of shape functions. *A*, *B* and *D* matrices are computed in the *processing* phase based on the number of layers, types and orientations, being related to the optimization process. Otherwise, integrals (I_i) are independent from the optimization process due to the constant topology and it can be computed only once and reused to the reconstruction (by linear combination) of the integral (12) in the *processing* phase - after *A*, *B* and *D* matrices computation.

The same observation sticks for all the stiffness matrix blocks. In [23] is presented an accurate evaluation scheme for triangular domain integrals.

A8 - Compute elemental stiffness matrices coefficients positions in the global stiffness matrix. This operation is done due to the observation that the global stiffness matrix is a sparse matrix. Matlab has very powerful numerical tools dedicated to sparse matrices, like the linear system solver which is used to solve the global system. A sparse matrix is defined in Matlab by matrix dimensions and by three vectors (*i, j, v*) where the elements of *i* and *j* represent the matrix positions (row, column) for the non-zero elements and the elements of *v* represents the values at those positions. In *preprocessing* phase *i* and *j* vectors representing the positions of the elemental stiffness matrices coefficients are computed. The vector *v* is afterwards reconstructed in the *processing* phase from *A*, *B* and *D* matrices coefficients and the precalculated integrals for elemental stiffness coefficients (A7).

Another important observation here is related to the case when vectors *i* and *j* contains multiple items for the same matrix position. The Matlab behaviour in this situation is to sum all the values from vector *v* corresponding to the same matrix position. This behaviour is equivalent to the *assembly process* – for global stiffness matrix.

A9 - Dirichlet boundary conditions – are directly applied for the global force vector. For the global stiffness matrix there are computed the correspondent positions in the *v* vector. After *v* vector is computed in the *processing* phase, the values from the correspondent positions are replaced accordingly with Dirichlet boundary conditions.

1.2. Processing

The *processing* phase is executed for all layup configurations generated during the optimization process. For obtaining a globally optimal solution a lot of layup configurations are evaluated in order to explore the complex solution space. To achieve this goal it is mandatory that the *processing* phase of the layup configuration to be very fast. Transferring a lot of operations to the *preprocessing* phase the amount of computations done in the *processing* phase is dramatically reduced to few which are presented below.

B1 - Computation of the *extensional stiffness* matrix (*A*), *bending stiffness* matrix (*D*) and *bending-extensional coupling stiffness* matrix (*B*) – using precomputed stiffness coefficients correspondent to the layers type and orientation (A3).

B2 - Computation of the *v* vector – by linearly combining specific *A*, *B* and *D* matrix coefficients (B1) with correspondent precomputed integrals (A7).

B3 - Apply Dirichlet boundary conditions for the global stiffness matrix – by replacing the values of the *v* vector at the precomputed positions associated with Dirichlet boundary condition (A9).

B4 - Solve the global linear system – using the Matlab solver for sparse matrices.

1.3. Post processing

Similarly with the *processing* phase, the *post processing* phase is executed for all layup configurations generated during the optimization process. The solutions of the global linear system represent the displacements and rotations of the midplane points associated with the topology nodes. Besides these, other mechanical characteristics are often required, sometimes for in plane positions different from nodes. All these additional operations are executed in the *post processing* phase. Because the requirements differ from a project to another, the *post processing* is a custom phase.

Below are listed some of the most common additional computations which are performed using the global linear system solutions.

C1 - Displacements computation – using relations (1).

C2 - Stresses computation – using *von Karman strain-displacements* relations and *Hooke's* law.

C3 - Failure criteria evaluation - *Tsai-Hill*, *Tsai-Wu*, etc.

2. Model validation

In order to validate the FE model it was developed the following two procedures:

- By comparison between the results obtained with the developed FE model and the results obtained with Ansys for the same specific geometry and boundary condition;
- By comparison between the results obtained with the developed FE model and the theoretical results available for the same specific geometry and boundary condition with available closed form solution.

2.1. Validation by comparison with Ansys results

The comparison procedure consists of the following steps:

- Compute the solution for the same geometry, boundary condition and layup configuration using the developed FE model and Ansys software;
- Export the nodes positions and displacements obtained with Ansys;
- Find the triangular elements which contains the Ansys nodes positions;
- Compute the displacements at Ansys nodes positions using the interpolation functions obtained with FE model;
- Compare the displacements obtained with the developed FE model and Ansys using the following deviation formula for each axis:

$$\varepsilon_{\%} = \left|\frac{\Delta_{Ansys} - \Delta_{Matlab}}{\Delta_{Ansys}}\right| \cdot 100 \tag{13}$$

The geometry and the boundary condition used are illustrated in figure 1(a) and consist of a rectangular plate with one edge clamped and a compression load was applied to the opposite edge. The layup configuration consist of 7 carbon fiber plies with [0º, 15º, 30º, 45º, 30º, 15º, 0º] orientation. The mechanical properties of the othotropic carbon fiber ply are shown in table 1.

Table 1. Mechanical properties of a carbon fiber lamina.

E_1 (GPa)	E_2 (GPa)	G_{12} (GPa)	G_{23} (GPa)	G_{13} (GPa)	ϑ_{12}(-)	ϑ_{21}(-)	h (mm)
147	10.3	7	3.7	7	0.27	0.0189	0.5

In figure 1 are shown (a) the geometry and boundary condition, (b) the mesh, (c-d) the displacements in x and y directions and (e-g) the stresses σ_x, σ_x and τ_{xy}. The displacements in z direction are 0 due to material symmetry and were skipped from plotting. The mesh consists of 1310 nodes and 2476 order I triangular elements. The mesh and the results from figure 1 are obtained with the developed FE model. In Ansys simulation it was used quadrilateral elements of type SHELL 181.

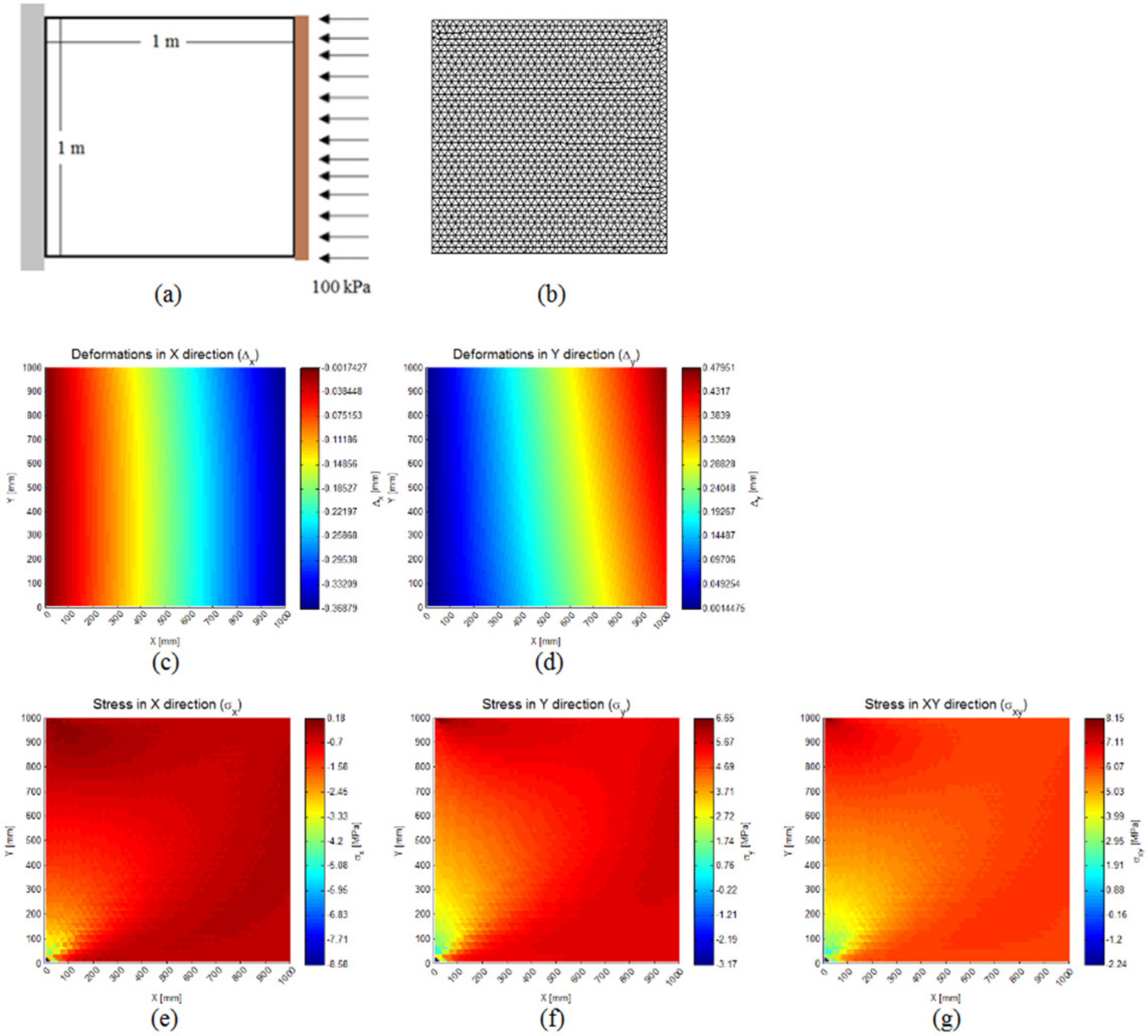

Figure 1. (a) geometry and boundary condition, (b) mesh, (c-d) displacements in *x* and *y* directions (e-g) σ_x, σ_x and τ_{xy}.

The mean and standard deviation for FE model vs. Ansys deviations – in percentage – is show below:

- $\boldsymbol{M^x = 0.1147}$ % $\boldsymbol{D^x = 0.559}$ %
- $\boldsymbol{M^y = 0.1811}$ % $\boldsymbol{D^y = 0.6051}$ %
- $\boldsymbol{M^{total} = 0.1045}$ % $\boldsymbol{D^{total} = 0.2237}$ %

It is obvious that formula (13) isn't applicable for the Ansys nodes with the corresponding displacement equal to 0. For these nodes we will show below that the maximum absolute deviation is very small. From a total of 784 Ansys nodes only 28 are subjected to this exception.

- $\Delta_{max}^{x} = 4.0753 \cdot 10^{-14}\ [m]$
- $\Delta_{max}^{y} = 5.6402 \cdot 10^{-14}\ [m]$
- $\Delta_{max}^{total} = 6.9584 \cdot 10^{-14}\ [m]$

From the above deviations it is obvious that the results obtained with the developed FE model are extremely close to the results obtained with Ansys.

2.2. Validation by comparison with theoretical results

For this procedure it was emulated the mechanical behaviour of an isotropic material for a rectangular plate with a small diameter central hole. The plate has one clamped edge and a traction load σ was applied to the opposite edge - figure 2(a). The theoretical expectation is that the stress at point *P* in the direction of the applied load is 3 times bigger than σ.

The layup configuration consists of a single layer and the material used in this simulation is steel with the following mechanical properties - table 2:

Table 2. Mechanical properties of a steel lamina.

E_1 (GPa)	E_2 (GPa)	G_{12} (GPa)	G_{23} (GPa)	G_{13} (GPa)	ϑ_{12}(-)	ϑ_{21}(-)	*h* (mm)
210	210	79	79	79	3	3	1

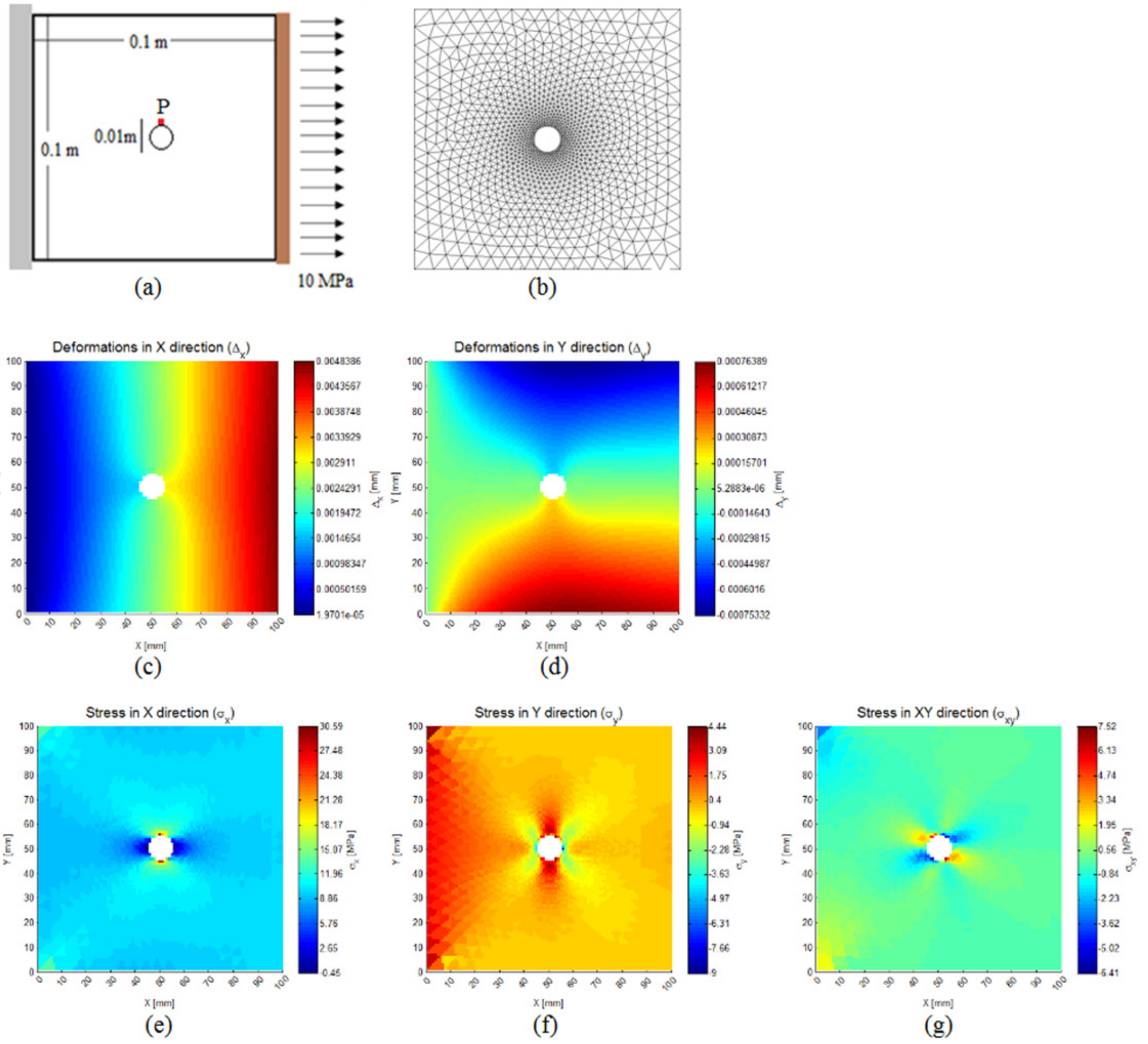

Figure 2. (a) geometry and boundary condition, (b) mesh, (c-d) displacements in *x* and *y* directions (e-g) σ_x, σ_x and τ_{xy}.

In figure 2 are shown (a) the geometry and boundary condition, (b) the mesh, (c-d) the displacements in x and y directions and (e-g) the stresses σ_x, σ_x and τ_{xy}. The displacements in z direction are 0 and were skipped from plotting. The mesh consists of 1240 nodes and 2375 order I triangular elements. The stresses σ_x at point *P* and at diametrical opposing point are 30.2 MPa and 30.1 MPa, respectively. The expected theoretical stress is 30 MPa which means that **the model error in this case is 0.66%.**

3. Execution time analysis

The model speed analysis was conducted on a regular PC with the following configuration: Intel dual CPU T3400 @2.16 GHz, RAM 2.5 GB DDR2. In this section all the time functions values are expressed in seconds and the elements used in simulation are triangular of order I.

The *preprocessing* phase was skipped from the speed analysis due to its purpose to be executed only once and to incorporate as much as possible operations. That being said the model executions time related to a single configuration simulation is defined as:

$$T_s(p,n) = T_{proc}(p,n) + T_{post}(n) \tag{14}$$

Where T_s, T_{proc} and T_{post} represent the total *simulation* time, *processing* time and *post processing* time, *p* represents the plies count and *n* represents the topology nodes count. The *processing* time can be defined as:

$$T_{proc}(p,n) = T_{B1}(p) + T_{B2}(n) + T_{B34}(n) \tag{15}$$

Where T_{B1}, T_{B2} represent the execution time related to B1 and B2 *processing* subphases and T_{B34} represents the execution time related to B3 and B4 *processing* subphases which were coupled here due to the fact that the execution time of the subphase B3 in neglectable.

As the *post processing* phase is a custom phase depending of the project we consider only the displacements computation to simplify our execution time analysis.

From (14) and (15):

$$T_s(p,n) = T_{B1}(p) + T_{B2}(n) + T_{B34}(n) + T_{post}(n) \tag{16}$$

In order to define the function $T_{B1}(p)$ we executed the *processing* subphase $T_{B1}$for $p = \overline{1{,}100}$ with random generated plies orientation in the [0°, 360°] range. It can be observed from figure 3(a) that the function $T_{B1}(p)$ is linear. Using linear regression with the constraint that for 0 plies the execution time is 0 we obtained the following form for $T_{B1}(p)$:

$$T_{B1}(p) = 6.2071 \cdot 10^{-5} \cdot p \tag{17}$$

A different analysis was conducted for $T_{B2}(n), T_{B34}(n)$ and $T_{post}(n)$. Because these functions are dependent of the topology number of nodes and the topology and boundary condition generation is a quite involving process it was considered only a set of 5 topologies with the number of nodes in the [0, 11660] range.

In the table 3 are presented the time results obtained at simulation for T_{B2}, T_{34}, T_{post} and also for T_{B234} - (T_{B2}+T_{B34}).

Table 3. Execution time analysis for the subphases dependent of the topology number of nodes.

p (-)	T_{B2} (s)	T_{B34} (s)	T_{234}= T_{B2}+ T_{B34} (s)	T_{post} (s)
128	0.015698	0.010708	0.026406	0.00011758
494	0.064868	0.046115	0.110983	0.00012638
1310	0.178807	0.169404	0.348211	0.00014996
5162	0.725306	0.997104	1.72241	0.00025433
11660	1.6813	3.9465	5.6278	0.00042567

In figure 3(b-c) are presented the results from table 3 together with the regression fits. It can be easily observed that $T_{B2}(n)$ and $T_{post}(n)$ are linear and $T_{B34}(n)$ - and consequently $T_{B234}(n)$ - have a quadratic behaviour.

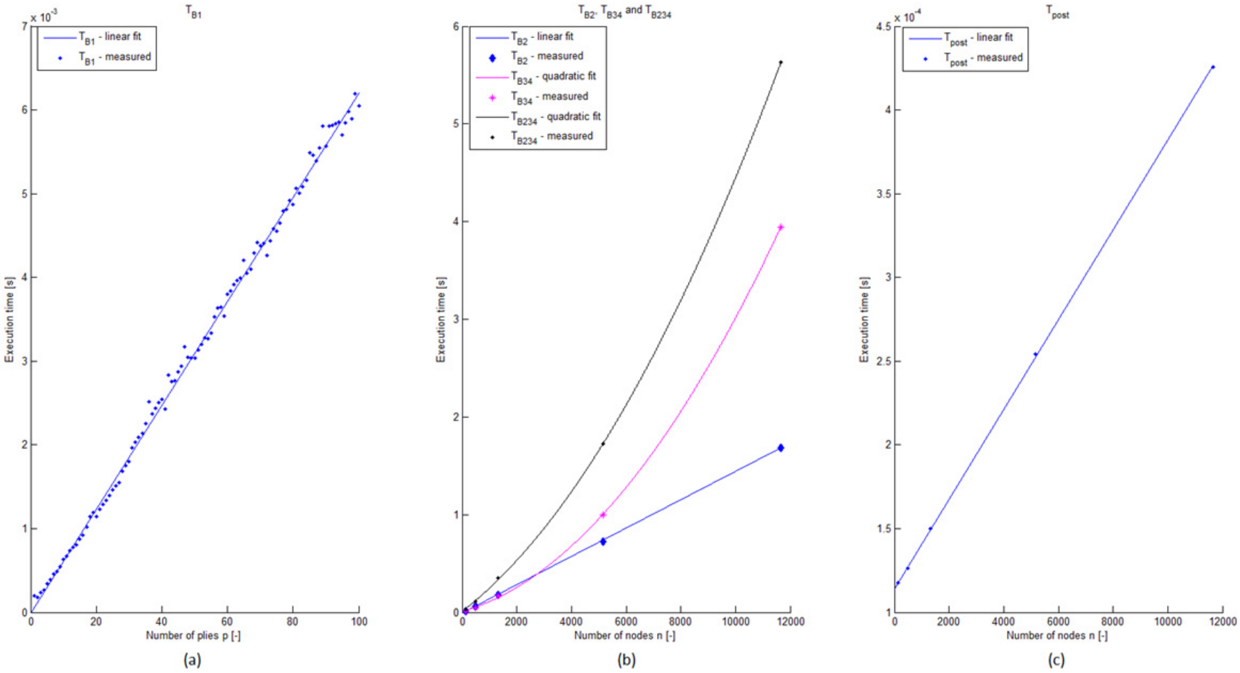

Figure 3. Execution time analysis for **(a)** $T_{B1}(p)$, **(b)** $T_{B2}(n), T_{B34}(n)$ and $T_{B234}(n)$ and **(c)** $T_{post}(n)$.

The regression was used to obtain the following forms for $T_{B2}(n)$, $T_{B34}(n)$ - and consequently $T_{B234}(n)$ – and $T_{post}(n)$:

$$T_{B2}(n) = 1.4419 \cdot 10^{-4} \cdot n \tag{18}$$

$$T_{B34}(n) = 2.22 \cdot 10^{-8} \cdot n^2 + 7.77 \cdot 10^{-5} \cdot n + 0.0087 \tag{19}$$

$$T_{B234}(n) = T_{B2}(n) + T_{B34}(n) = 2.22 \cdot 10^{-8} \cdot n^2 + 2.2189 \cdot 10^{-4} \cdot n + 0.0087 \tag{20}$$

$$T_{post}(n) = 2.6763 \cdot 10^{-8} \cdot n + 1.144 \cdot 10^{-4} \tag{21}$$

The reason that $T_{B34}(n)$ and $T_{post}(n)$ have non-zero intercepts are related to the fact that the computation contains some operations unrelated to the number of nodes.

Of course, the forms of all the time functions are dependent of the computation capability. However, we expect that the proportions between execution times remain relatively constant on different computers.

From the above analysis it can be observed that the biggest execution time is related to linear combination for obtaining the vector v (B2) and to the global linear system solving (B34). The computation complexity of subphase (B2) is very small and the Matlab code is vectorized in order to obtain the smallest execution time. In order to improve the execution time of subphase (B2) some advanced computation techniques should be addressed – parallel computing for example – but this exceeds our purpose. Also, the subphase (B34) consists of solving the global linear system using the Matlab solver for sparse matrices which is a very powerful solver and improving the execution time for subphase (B34) will involve the development of a faster solver which exceeds our purpose. That being said, we can consider that subphases (B2) and (B34) cannot be further optimized without addressing very complex computation techniques and numerical methods and this is the reason we had coupled the subphases as (B234).

Considering (17-21) and replacing in (16) we obtain:

$$T_s(p,n) = 6.2071 \cdot 10^{-5} \cdot p + 2.22 \cdot 10^{-8} \cdot n^2 + 2.21916 \cdot 10^{-4} \cdot n + 8.8144 \cdot 10^{-3} \quad (22)$$

For our layup optimization purpose some typical values for the number of layers and for the number of nodes are p = 30 and n = 500. For these values the total execution time (T_s) obtained with the testing PC is 0.1272 s. The execution time for the linear combining and global system solving (T_{B234}) is 0.1252 s and represents $\mathbf{98.43\%}$ from the total execution time (T_s) which means that very small improvements can be achieved without addressing complex computation techniques.

4. Conclusions

In this paper we presented a fast and accurate FE model which can be used in the mechanical behaviour analysis and layup optimization of the laminated composites. Even the developed model is based on the *first-order shear deformation theory* (FSDT) the strategies used for speeding up the FEA of laminated composites easily extend to any two-dimensional theory.

The model was validated by comparison with Ansys results and by comparison with some theoretical results.

The fastness is obtained by organizing the model in three phases: *preprocessing, processing* and *post processing*. A lot of computations are assigned to the *preprocessing* phase which is executed only once and the smallest amount of operations is assigned to the remaining phases which are executed for each layup configuration. We have shown that more than 95% of the total execution time of the *processing* and *post processing* phases is related to global linear system regeneration (by a low complexity linear combination operation) and solving (using the Matlab solver for sparse matrices). We consider that any further substantial fastness improvement should address complex computation techniques and numerical methods which exceed our purpose.

5. References

[1] Bagheri M, Jafari A A and Sadeghifar M 2011 Multi-objective optimization of ring stiffened cylindrical shells using a genetic algorithm *Journal of Sound and vibration* pp 374-384

[2] Chatzi E N, Hiriyur B, Waisman H and Smyth A W 2011 Experimental application and enhancement of the XFEM-GA algorithm for the detection of flows in structures *Computers and structures* pp 556-570

[3] Ghiasi H, Pasini D and Lessard L 2009 Optimum stacking sequence design of composites materials Part I: Constant stiffness design *Composites structure* pp 1-11

[4] Ghiasi H, Fayazbakhsh K, Pasini D and Lessard L 2010 Optimum stacking sequence desing of composite materials Part II: Variable stiffness design *Composite Structures* pp 1-13

[5] Badallo P, Trias D, Marin L and Mayugo J A 2013 A comparative study of genetic algorithms for the multi-objective optimization of composite stringers under compression load *Composites: Part B* pp 130-136

[6] Montagnier O and Hochard Ch 2013 Optimization of hybrid high-modulus/high-strength carbon

fiber reinforced plastic composite drive shafts *Materials and Design* pp 88-100

[7] Wang W, Guo S, Chang S, Chang N and Yang W 2010 Optimum buckling design of composite stiffened panels using and ant colony algorithm *Composite Structures* pp 712-719

[8] Zein S, Basso P and Grihon S 2014 A constrained satisfaction programming approach for computing manufacturable stacking sequence *Computers and structures* pp 56-63

[9] Yong M, Falzon B G and Iannucci L 2008 On the application of genetic algorithms for optimizing composites against impact loading *International journal of impact engineering* pp 1293-1302

[10] Yong M, Iannucci L and Falzon B G 2010 Efficient modeling and optimization of hybrid multilayered plates subject to ballistic impact *International journal of impact engineering* pp 605-624

[11] Pohlak M, Majak J, Karjust K and Kuttner R 2010 Multi-criteria optimization of large composite parts *Composite Structures* pp 2146-2152

[12] Lee D S, Morillo C, Bugeda G, Oller S and Onate E 2012 Multilayered composite structure design optimization using distributed/parallel multi-objective evolutionary algorithms *Composite Structure* pp 1087-1096

[13] Falzon B G and Faggiani A 2012 The use of a genetic algorithm to improve the postbuckling strength of stiffened composite panel susceptible to secondary instabilities *Composite Structure* pp 883-895

[14] Murugan S, Flores E I S, Adhikari S and Friswell M I 2012 Optimal design of variable fiber spacing composites for morphing aircraft skin *Composite Structures* pp 1626-1633

[15] Sliseris J and Rocens K 2013 Optimal design of composite plates with discrete variable thickness *Composite Structures* pp 15-23

[16] Sharma D S, Patel N P and Trivedi R R 2014 Optimum design of laminates containing an elliptical hole *International journal of mechanical science,* pp 76-87

[17] Park C H, Lee W I, Han W S and Vautrin A 2008 Improved genetic algorithm for multidisciplinary optimization of composites laminates *Computers and Structures,* pp 1894-1903

[18] Kayikci R, Sonmez F O 2012 Design of composite laminates for optimum frequency response *Journal of sound and vibration,* pp 1759-1776

[19] Papadopoulos L and Kassapoglou C 2007 Shear buckling of rectangular composites plates composed of concentric layups *Composites part A* pp 1425-1430

[20] Persson P-O and Strang G 2004 A simple mesh generator in MATLAB, *SIAM Rev.* **46** pp 329–345

[21] Lee K-Y, Kim I-I, Cho D-Y and Kim T 2003 An algorithm for automatic 2D quadrilateral mesh generation with line constraints *Computer-aided design* **35** pp 1055-1068

[22] Remacle J-F, Lambrechts J, Seny B, Marchandise E, Johnen A and Geuzaine C 2012 Blossom-Quad: a non-uniform quadrilateral mesh generator using a minimum cost perfect matching algorithm *International journal for numerical methods in engineering* **89** pp 1102-1119

[23] Farzana H and Karim M S 2012 Accurate Evaluation schemes for triangular domain integrals *Journal of Mechanical and Civil Engineering* **2** pp 38-51