Flexible multibody dynamics using polygonal elements

Arturo Cubas[1] and Ivan F. M. Menezes[2]
[1]*Mechanical Engineering Department, PUC-Rio, arturo.cubas@aluno.puc-rio.br*
[2]*Mechanical Engineering Department, PUC-Rio, ivan@puc-rio.br*

ABSTRACT — *The main goal of this work is to present numerical simulations of flexible multibody systems (MBS) using polygonal finite elements. We study their full dynamic response considering large displacements and rotations but small deformations. The use of polygonal elements has presented some advantages for modeling complex geometries and solving the corresponding elasticity problems, when compared to traditional simplex elements (e.g., triangles and quads). The proposed MATLAB implementation is compared with the ANSYS software [1] using two representative examples of flexible MBS.*

1 Introduction

A flexible multibody system is defined as a set of flexible or rigid elements, constrained by flexible or rigid joints and force elements such as dampers or springs ([2]). The system may experience large displacements and rotations that lead to a geometrically nonlinear problem governed by a set of differential and algebraic equations ([3]). In order to avoid numerical instabilities, an appropriate solver must be selected. In this work, we use the generalized-α method ([4]).

According to the characteristics of the strain tensor ϵ, three types of multibody systems can be obtained ([3]): the system is said to be rigid if $\epsilon = 0$. The constraints can be rigid or flexible, representing localized deformations. The system is *linearly elastic* if $\epsilon \ll 1$, in this case the strain–displacement relationship is assumed to be linear. Usually the displacement field is simplified using modal synthesis for enhancing the computational efficiency. If the strain–displacement relation is non-linear or the strains are large, the system is *non-linearly elastic*. For large strains the constitutive equation is non-linear.

An isolated body may be modeled by a number of methods. The *finite element (FE) method* uses the absolute nodal coordinates with respect to a unique inertial frame. These *generalized coordinates* express the bodies motion and rotation as well as the joints constraints. In the *corrotational frame* approach, the motion for each element can be decomposed into large displacements and assumed small deformations. Therefore, linear elasticity is employed to calculate the elastic forces. Moreover, in the *floating frame of reference* approach, an intermediate frame is assigned to each body. The motion is described for this frame and the elastic deformations are characterized by additional flexible coordinates.

Modal expansions and linear assumptions are not always suitable for describing the dynamics of multibody systems because they cannot express accurately the nonlinear behavior of the system that arises when the displacements and velocities are not small. The nonlinear FE method can accurately capture these nonlinear effects, but at the cost of generating large and sparse matrices. Therefore, the efficiency of the numerical method relies on the sparse linear solver being used. Furthermore, owing to the nonlinear nature of the system, the large and sparse matrices need to be updated, and the associated linear system of equations need to be solved several times for each time step.

This work presents a computational environment for dynamic simulations of flexible multibody systems (MBS) using polygonal FE meshes. Recent studies on topology optimization ([5], [6], [7]) have shown that polygonal elements can be very useful in numerical analyses as they provide great flexibility in discretizing complex domains and are computationally very efficient when compared to conventional simplex elements (e.g., triangles and quads).

Moreover, the use of polygonal elements in topology optimization problems has proven to be more efficient when compared to conventional elements because they prevent the formation of numerical anomalies such as checkerboard patterns or single node connections ([5], [6], [7]). As the next step of the present research is to apply topology optimization techniques to minimize the overall weight of MBS, we decided to use polygonal elements in this study.

2 Equations of motion and finite element discretization

The methodology used to solve these problems follows the variational approach for a constrained MBS as stated in ([2]), ([8]) and ([9]). The flexible MBS can be modeled as a set of discretized bodies represented by a set of n generalized coordinates q, satisfying the following set of m holonomic constraints:

$$\boldsymbol{\phi}(\boldsymbol{q},t)=\mathbf{0}. \tag{1}$$

The Lagrangian function $\mathscr{L}$ that relates the kinetic $\mathscr{K}$, potential $\mathscr{V}$ and the deformation $\mathscr{W}$ energies can be written as

$$\mathscr{L}(\boldsymbol{q},\dot{\boldsymbol{q}})=\mathscr{K}(\boldsymbol{q},\dot{\boldsymbol{q}})-\mathscr{V}(\boldsymbol{q})-\mathscr{W}(\boldsymbol{q}). \tag{2}$$

The vector holding the nodal displacements $\boldsymbol{q}(t)$ is obtained by solving the variational problem

$$\delta\int_{t_1}^{t_2}\left(\mathscr{L}-\boldsymbol{\lambda}^T\boldsymbol{\phi}\right)dt=0. \tag{3}$$

where $\boldsymbol{\lambda}$ is a vector of Lagrange multipliers that take into account the geometrical constraints amongst the bodies. After integrating Eq. (3), a differential–algebraic system of equations (DAE) is obtained as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}+\boldsymbol{g}^{gyr}(\boldsymbol{q},\dot{\boldsymbol{q}})+\boldsymbol{g}^{int}(\boldsymbol{q})-\boldsymbol{g}^{ext}(\boldsymbol{q})+\boldsymbol{\phi}_{\boldsymbol{q}}^T(\boldsymbol{q},t)\boldsymbol{\lambda}=\mathbf{0}, \tag{4}$$

$$\boldsymbol{\phi}(\boldsymbol{q},t)=\mathbf{0}, \tag{5}$$

where $\boldsymbol{g}^{gyr}=\partial\mathscr{K}/\partial\boldsymbol{q}$, $\boldsymbol{g}^{int}=\partial\mathscr{W}/\partial\boldsymbol{q}$, $\boldsymbol{g}^{ext}=-\partial\mathscr{V}/\partial\boldsymbol{q}$, $\boldsymbol{M}$ is the mass matrix, $\boldsymbol{\phi}_{\boldsymbol{q}}(\boldsymbol{q},t)=\partial\boldsymbol{\phi}/\partial\boldsymbol{q}$. The force vectors are gathered into $\boldsymbol{g}=\boldsymbol{g}^{gyr}+\boldsymbol{g}^{int}-\boldsymbol{g}^{ext}$.

In order to solve this continuous system of equations, Eq. (4) is expressed as a residual $\boldsymbol{r}=\boldsymbol{M}\ddot{\boldsymbol{q}}+\boldsymbol{g}+\boldsymbol{\phi}_{\boldsymbol{q}}^T\boldsymbol{\lambda}$, whose solution is $\boldsymbol{q}(t)$, $\boldsymbol{\lambda}(t)$. The time is discretized into equal steps and we assume that initial conditions $\boldsymbol{q}_0,\dot{\boldsymbol{q}}_0,\ddot{\boldsymbol{q}}_0,\boldsymbol{\lambda}_0$ are given. The solver attempts to minimize both $\boldsymbol{r}$ and $\boldsymbol{\phi}$, according to a prescribed numerical tolerance and this is accomplished by integrating Eq. (4) using the generalized-α method ([4]). Additionally, it should be noticed that high index DAE numerical solutions are polluted by perturbations due to finite arithmetic precision when small time steps are chosen. Thus a preconditioning scheme is necessary. In this work we use the preconditioning method described in ([10]).

2.1 Isoparametric Formulation

The kinematic measure of deformation and the isoparametric formulation for bi-dimensional continuous elements will be discussed in this section according to the FE approach. First, the spatial and material coordinates $\boldsymbol{x}$ and $\boldsymbol{X}$ respectively, are interpolated as a linear function of the shape functions N_i and the generalized coordinates q_i of the element

$$\boldsymbol{x}=\begin{bmatrix}x\\y\end{bmatrix}=\boldsymbol{N}\boldsymbol{q}_e=\begin{bmatrix}N_1\boldsymbol{I} & N_2\boldsymbol{I} & \cdots & N_{ne}\boldsymbol{I}\end{bmatrix}\begin{bmatrix}q_1\\q_2\\\vdots\\q_{nn}\end{bmatrix}, \tag{6}$$

$$\boldsymbol{X}=\begin{bmatrix}X\\Y\end{bmatrix}=\boldsymbol{N}\boldsymbol{q}_{e0}. \tag{7}$$

The Jacobian with respect to the undeformed configuration and the gradient of deformations can be found from the chain rule

$$\boldsymbol{J} = \left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{r}}\right)^T, \tag{8}$$

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{r}} \boldsymbol{J}^{-T}. \tag{9}$$

The derivative $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{r}}$ is used to find the gradient of deformation

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{r}} = \begin{bmatrix} \boldsymbol{N}_r \boldsymbol{q}_e, & \boldsymbol{N}_s \boldsymbol{q}_e \end{bmatrix}, \tag{10}$$

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{N}_r \boldsymbol{q}_e & \boldsymbol{N}_s \boldsymbol{q}_e \end{bmatrix} \boldsymbol{J}^{-T}, \tag{11}$$

where $\boldsymbol{N}_r$ and $\boldsymbol{N}_s$ stand for the derivative of the $\boldsymbol{N}$ matrix respect to r and s respectively. The GreenLagrange tensor of deformation, necessary for large rotations, is found in its tensor and vector forms

$$\begin{aligned} \hat{\boldsymbol{\epsilon}} &= \frac{1}{2}\left(\boldsymbol{F}^T \boldsymbol{F} - \boldsymbol{I}\right), \\ &= \frac{1}{2}\left(\boldsymbol{J}^{-1} \begin{bmatrix} \boldsymbol{q}_e^T \boldsymbol{N}_r^T \boldsymbol{N}_r \boldsymbol{q}_e & \boldsymbol{q}_e^T \boldsymbol{N}_r^T \boldsymbol{N}_s \boldsymbol{q}_e \\ \boldsymbol{q}_e^T \boldsymbol{N}_s^T \boldsymbol{N}_r \boldsymbol{q}_e & \boldsymbol{q}_e^T \boldsymbol{N}_s^T \boldsymbol{N}_s \boldsymbol{q}_e \end{bmatrix} \boldsymbol{J}^{-T} - \boldsymbol{I}\right), \end{aligned} \tag{12}$$

$$\begin{aligned} \boldsymbol{\epsilon} &= \begin{bmatrix} \hat{\varepsilon}_{11} & \hat{\varepsilon}_{22} & 2\hat{\varepsilon}_{12} \end{bmatrix}^T, \\ &= \begin{bmatrix} \boldsymbol{q}_e^T \boldsymbol{P}_1 \boldsymbol{q}_e \\ \boldsymbol{q}_e^T \boldsymbol{P}_2 \boldsymbol{q}_e \\ \boldsymbol{q}_e^T \boldsymbol{P}_3 \boldsymbol{q}_e \end{bmatrix} - \begin{bmatrix} 1/2 \\ 1/2 \\ 0 \end{bmatrix}, \end{aligned} \tag{13}$$

where the matrices $\boldsymbol{P}_i$ are

$$\begin{aligned} \boldsymbol{P}_1 &= \frac{1}{2J^2}\left(J_{22}^2 \boldsymbol{N}_r^T \boldsymbol{N}_r + J_{12}^2 \boldsymbol{N}_s^T \boldsymbol{N}_s - J_{12} J_{22} \boldsymbol{N}_r^T \boldsymbol{N}_s - J_{12} J_{22} \boldsymbol{N}_s^T \boldsymbol{N}_r\right), \\ \boldsymbol{P}_2 &= \frac{1}{2J^2}\left(J_{21}^2 \boldsymbol{N}_r^T \boldsymbol{N}_r + J_{11}^2 \boldsymbol{N}_s^T \boldsymbol{N}_s - J_{11} J_{21} \boldsymbol{N}_r^T \boldsymbol{N}_s - J_{11} J_{21} \boldsymbol{N}_s^T \boldsymbol{N}_r\right), \\ \boldsymbol{P}_3 &= \frac{1}{J^2}\left(J_{11} J_{22} \boldsymbol{N}_r^T \boldsymbol{N}_s - J_{21} J_{22} \boldsymbol{N}_r^T \boldsymbol{N}_r - J_{11} J_{12} \boldsymbol{N}_s^T \boldsymbol{N}_s + J_{12} J_{21} \boldsymbol{N}_s^T \boldsymbol{N}_r\right). \end{aligned} \tag{14}$$

Next the stress-strain matrix is found from the deformation measure

$$\boldsymbol{B} = \frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{q}_e} = \begin{bmatrix} \boldsymbol{q}_e^T \left(\boldsymbol{P}_1 + \boldsymbol{P}_1^T\right) \\ \boldsymbol{q}_e^T \left(\boldsymbol{P}_2 + \boldsymbol{P}_2^T\right) \\ \boldsymbol{q}_e^T \left(\boldsymbol{P}_3 + \boldsymbol{P}_3^T\right) \end{bmatrix}. \tag{15}$$

The virtual work of the internal forces and the associated virtual deformation are

$$\delta \mathscr{W}_e = \int_{V_e} \delta \boldsymbol{\epsilon}^T \boldsymbol{S} dV, \tag{16}$$

$$\delta \boldsymbol{\epsilon} = \boldsymbol{B} \delta \boldsymbol{q}_e. \tag{17}$$

By replacing the virtual deformation in the virtual work equation, we obtain the vector of internal forces

$$\delta \mathscr{W}_e = \delta \boldsymbol{q}_e^T \int_{V_e} \boldsymbol{B}^T \boldsymbol{S} dV, \tag{18}$$

$$\boldsymbol{g}_e^{int} = \int_{V_e} \boldsymbol{B}^T \boldsymbol{S} dV. \tag{19}$$

The second PiolaKirchoff tensor of stress is obtained from the constitutive law

$$\boldsymbol{S} = \boldsymbol{C}\boldsymbol{\epsilon}. \tag{20}$$

Then, the stiffness matrix is obtained by taking the derivative of the internal force, resulting in two well-known terms: the elastic and the geometric stiffness matrices. The last one can be found from the $\boldsymbol{P}_i$ matrices, i.e.,

$$\boldsymbol{K}_e = \frac{\partial \boldsymbol{g}_e^{int}}{\boldsymbol{q}_e} = \boldsymbol{K}_e^m + \boldsymbol{K}_e^g, \tag{21}$$

$$\boldsymbol{K}_e^m = \int_{V_e} \boldsymbol{B}^T \boldsymbol{C} \boldsymbol{B} dV, \tag{22}$$

$$\boldsymbol{K}_e^g = \int_{V_e} \left(\left(\boldsymbol{P}_1 + \boldsymbol{P}_1^T \right) S_1 + \left(\boldsymbol{P}_2 + \boldsymbol{P}_2^T \right) S_2 + \left(\boldsymbol{P}_3 + \boldsymbol{P}_3^T \right) S_3 \right) dV. \tag{23}$$

Note that we reuse the $\boldsymbol{P}_i$ matrices in order to achieve numeric performance. To obtain the vector and matrix of external forces, we derive the potential energy expression, i.e.,

$$\mathscr{V}_e = m_e g \bar{y}_e, \tag{24}$$

$$\boldsymbol{g}_e^{ext} = -\frac{\partial \mathscr{V}_e}{\partial \boldsymbol{q}_e} = -m_e g \frac{\partial \bar{y}_e}{\partial \boldsymbol{q}_e}, \tag{25}$$

$$\boldsymbol{K}_e^{ext} = -m_e g \frac{\partial}{\partial \boldsymbol{q}_e} \left(\frac{\partial \bar{y}_e}{\partial \boldsymbol{q}_e} \right). \tag{26}$$

$$\tag{27}$$

2.2 Polygonal elements remarks

The polygonal FE meshes were obtained using the Polymesher code ([5]). It uses centroidal Voronoi diagrams for discretization and a signed distance function to construct domain geometries with Boolean operations. For integration purposes, each polygon that has n nodes is decomposed into n triangles by connecting its nodes with its centroid. The integration procedure is carried out inside these triangles by well-known integration rules ([6]).

3 Applications

We present two problems that comprise the analysis of flexible multibody systems composed of plates in plane stress state, subjected to large displacements but small deformations. For the sake of simplicity, the material behavior is considered linear. The 2D elements are subject to stretching and bending, and the point mass has no rotational effects. In the proposed FEM code, developed in MATLAB, we use a fixed time step of 0.01 s. In this work, the generalized-α method is used to solve the nonlinear dynamic problem, due to its unconditional stability. Proper joint modeling and parameter settings are crucial to obtain accurate solutions. To validate the proposed implementation, the results obtained here are compared with the ones obtained with the ANSYS software.

3.1 First problem

The first problem consists of two plate bodies in a vertical plane x-y with a point mass at the tip and two rotational joints with z-axis degree of freedom as shown in the Fig. 1a. This problem was simulated with ANSYS Workbench, with $E = 2e + 11Pa$, $\rho = 7850kg/m^3$, and a Poissons ratio of 0.3. The point mass has $m = 5kg$, $g = 9.81m/s^2$, and the plate has a thickness of 50 mm. ANSYS uses an HTT integrator with a fixed time step of 0.01 s. Figures 1b and 2 shows the results obtained using the proposed implementation and the one obtained with the ANSYS software, respectively. Figure 3 illustrates the time histories of the tip node displacements x and y obtained with the proposed MATLAB code and the ANSYS software. It can be observed that the results are nearly identical.

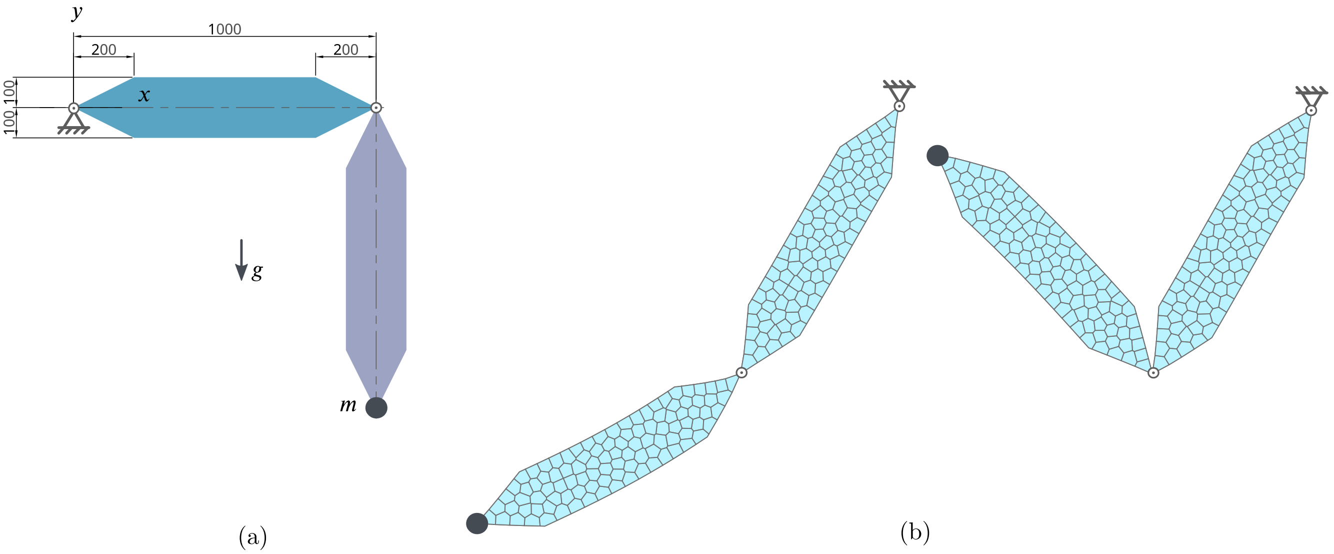
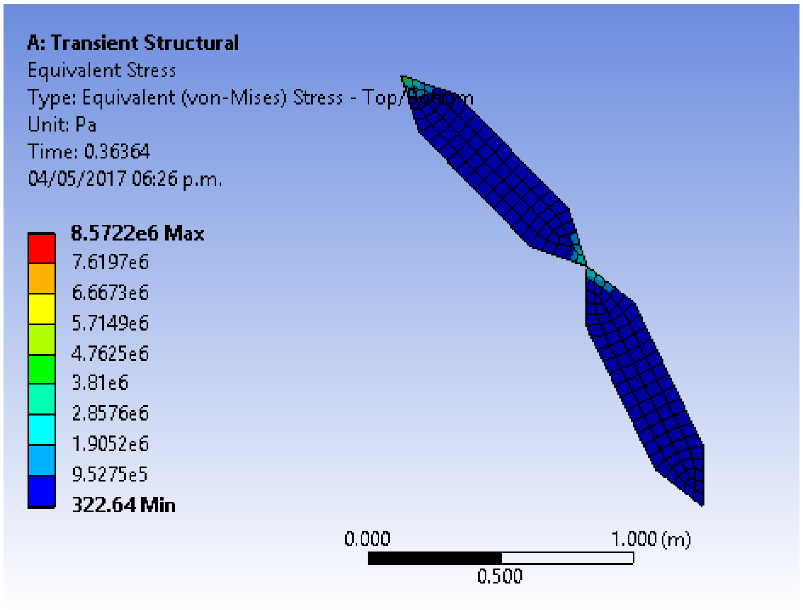


Fig. 1: First problem: (a) two vertical flexible plates with two joints and a mass (units in mm). (b) Deformed configuration for two time steps using the proposed MATLAB code.



A: Transient Structural
Equivalent Stress
Type: Equivalent (von-Mises) Stress - Top/Bottom
Unit: Pa
Time: 0.36364
04/05/2017 06:26 p.m.

Fig. 2: Simulated plates at t = 0.36 s using ANSYS.

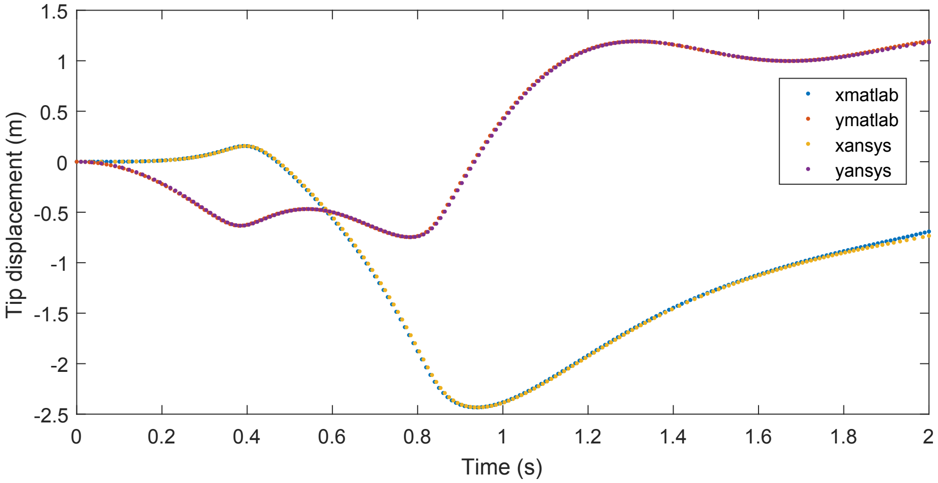


Fig. 3: MATLAB and ANSYS *x* and *y* responses for the tip node in the first problem.

3.2 Second problem

The second problem consists of imposing a prescribed variation in the angle at each rotational joint, as function of time, as shown in Fig. 4. The geometry and input data are the same as in the first example. This problem comprises three stages: two constant acceleration stages and one constant velocity stage. The MBS starts in the position shown in Fig. 5 and no gravity is considered. Figure 6 shows the von Mises stresses in the plates corresponding to the last time step of the dynamic simulation, and Fig. 7 illustrates the time histories of the tip node displacements *x* and *y* obtained with the proposed MATLAB code and the ANSYS software. It can be noticed that the responses are no longer identical. This is due to the different models used to represent the joints. In the MATLAB code, the forces at the joints are applied to the neighbors of the plate tip nodes, and therefore, the applied force is distributed among these nodes. In the ANSYS software, the forces at the joints are applied directly at the tip node, which leads to larger displacements at the tips.

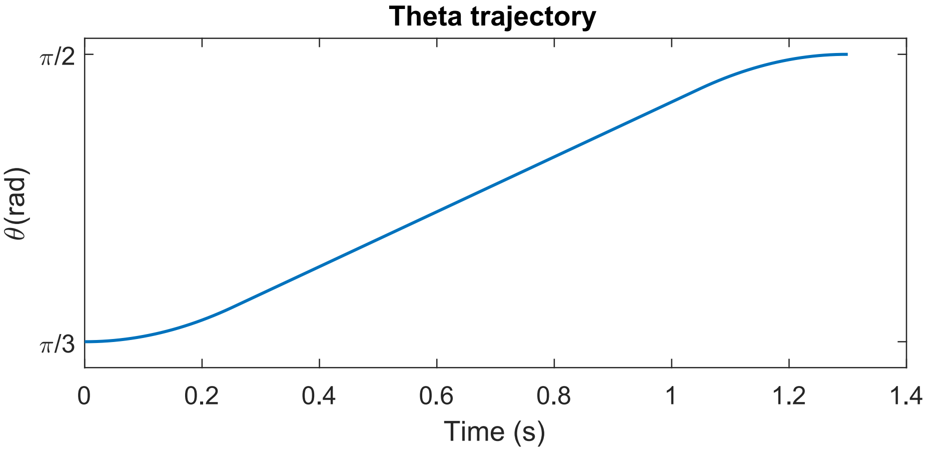


Fig. 4: Angles trajectories as function of time.

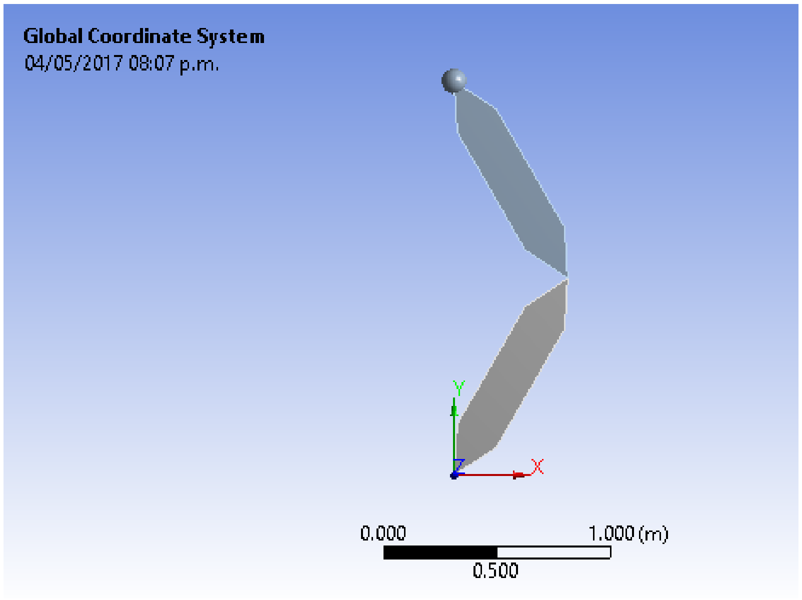


Fig. 5: Initial position for the second problem.

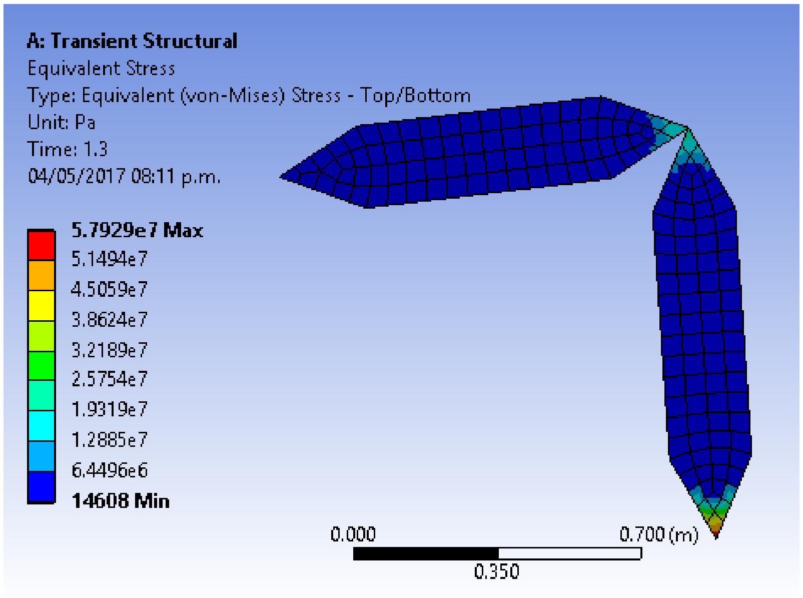


Fig. 6: Simulated plates at t = 1.3 s using ANSYS for the second problem.

4 Conclusions

This work presented numerical simulations of flexible multibody systems using polygonal finite elements. We have shown expressions for computing the internal forces and tangent matrices that arise from the Hamilton principle and the FEM applied to continuous elements. A MATLAB code for dynamic simulation of flexible MBS was proposed and tested by means of two examples corresponding to 2D plates in a plane stress state. The results obtained were in good agreement with the ones obtained using the ANSYS software. The use of the presented methodology for minimizing the overall weight of MBS, in the context of topology optimization, is currently under investigation by the authors. A three-dimensional extension can be made by using polyhedral meshes and

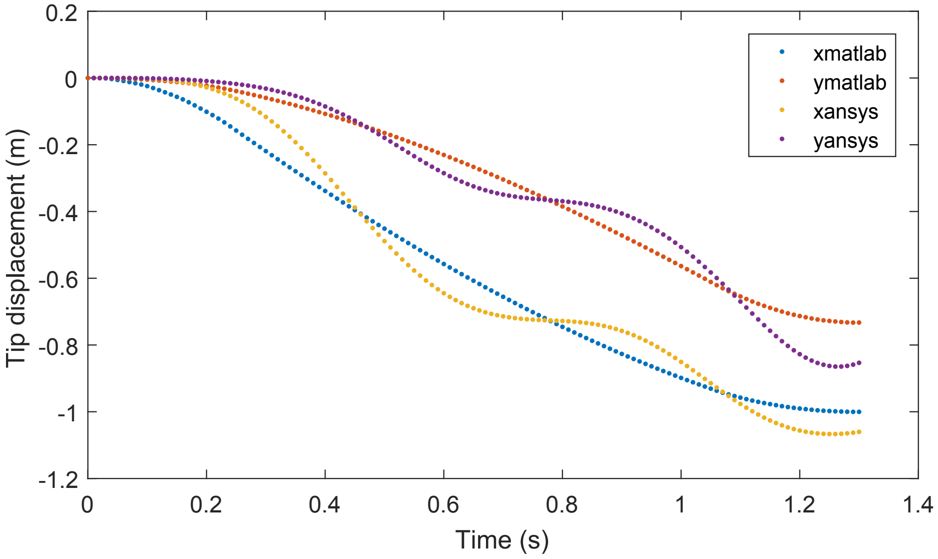


Fig. 7: MATLAB and ANSYS *x* and *y* responses for the tip node in the second problem.

the recently proposed Virtual Element Method ([11]).

Acknowledgements

The first author acknowledges the financial support provided by the Brazilian National Council for Scientific and Technological Development (CNPq).

References

[1] *ANSYS® Academic Research Mechanical, Release 18.1.*

[2] M. Geradin and A. Cardona, *Flexible Multibody Dynamics: A Finite Element Approach.* Wiley, 2001.

[3] O. A. Bauchau, *Flexible Multibody Dynamics.* Springer Netherlands, 2011.

[4] J. Chung and G. Hulbert, "Time integration algorithm for structural dynamics with improved numerical dissipation: the generalized- α method," *Journal of Applied Mechanics, Transactions ASME*, vol. 60, pp. 371–375, 6 1993.

[5] C. Talischi, G. H. Paulino, A. Pereira, and I. F. M. Menezes, "Polymesher: a general-purpose mesh generator for polygonal elements written in matlab," *Structural and Multidisciplinary Optimization, 45(3):309-328*, 2012.

[6] C. Talischi, G. H. Paulino, A. Pereira, and I. F. Menezes, "Polytop: A matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes," *Struct. Multidiscip. Optim.*, vol. 45, pp. 329–357, Feb. 2012.

[7] C. Talischi, G. H. Paulino, A. Pereira, and I. F. M. Menezes, "Polygonal finite elements for topology optimization: A unifying paradigm," *International Journal for Numerical Methods in Engineering*, vol. 82, no. 6, pp. 671–698, 2010.

[8] O. Brüls, A. Cardona, and M. Géradin, *Modelling, simulation and control of flexible multibody systems*, pp. 21–74. Vienna: Springer Vienna, 2009.

[9] O. Brüls, E. Lemaire, P. Duysinx, and P. Eberhard, *Optimization of Multibody Systems and Their Structural Components*, pp. 49–68. Dordrecht: Springer Netherlands, 2011.

[10] C. L. Bottasso, D. Dopico, and L. Trainelli, "On the optimal scaling of index three daes in multibody dynamics," *Multibody System Dynamics*, vol. 19, pp. 3–20, Feb 2008.

[11] A. L. Gain, G. H. Paulino, L. S. Duarte, and I. F. Menezes, "Topology optimization using polytopes," *Computer Methods in Applied Mechanics and Engineering*, vol. 293, pp. 411 – 430, 2015.