26th International Congress of Mechanical Engineering
COBEM 2021
ABCM Associação Brasileira de Engenharia e Ciências Mecânicas
November 22-26, 2021, Virtual Congress, Brazil

COB-2021-0089
SOFTWARE DEVELOPMENT TO THE DESIGN OF FOUR-BAR LINKAGES WITH PYTHON

Cádmo Augusto Rodrigues Dias
Claysson Bruno Santos Vimieiro
Jánes Landre Júnior
Pontifícia Universidade Católica de Minas Gerais (PUC Minas). R. Dom José Gaspar, 500 - Coração Eucarístico, Belo Horizonte - MG, 30535-901.
diascadmo@gmail.com
claysson@pucminas.br
janes@pucminas.br
Abstract. *The four-bar linkage is a relatively old and still widely used solution for several engineering applications, such as windshield wipers, pumpjacks and medical beds. Basically, this mechanism is composed by four rigid links, connected to each other by means of pin joints, where the length and mass of these rigid links, for example, define the behavior of the mechanism. Parallel to this, the use of simulation by engineering has become a fundamental resource during the development of new solutions and, therefore, engineers with skills in programming languages and development of algorithms have become common in the present times. In this context, the present work seeks to develop a software (4bar Solver), using Python computer programming language, to assist in the development of four-bar linkage designs. As a result, the user obtains the torque and forces on the bars as well as the angular displacement, velocity and acceleration plots of the output bar during the complete movement cycle. Seeking to validate the 4bar Solver, problems of consolidated literature are simulated and the same results are found, thus validating the software developed by the present work.*

Keywords: *four-bar linkage, kinematics and dynamics of machinery, software development, python*

1. INTRODUCTION

The four-bar linkage consists of four rigid links connected by four-pin joints, which consequently results in a great adaptability and variety of possible output motions (Khan *et al.*, 2020). Consequently, the mechanism approached by the present work is applied in many areas: Landing gear systems (Son; Huda, 2019), control and robotics (Salah *et al.*, 2019; Zhang *et al.*, 2018), bioengineering (Kittisares *et al.*, 2020; Olsen; Camp; Brainerd, 2017) and onshore wave energy converters (Albert *et al.*, 2017) are examples.

Thus, despite being a simple and relatively old mechanism (Mabie; Reinholtz, 1987), several techniques regarding its development and analysis are researched till the present day, which corroborates the importance of this mechanism. While Wu *et al*. (2021) developed a fully analytical method for coupler-curve synthesis of planar four-bar linkages, Zarkandi (2021) approached the same mechanism to find multiple solutions for path synthesis with the aid of an optimization-based method. Still regarding the path generation, Bureerat and Sleesongsom (2021) used self-adaptive teaching–learning-based optimization while Li and Hao (2018) used a four-bar linkage aiming the kinematic singularities existing in rigid-body linkages to design a mechanism with nonlinear behaviors.

There are a few methods to the development and analysis of four-bar linkages, such as the graphic, iterative and analytical. While the graphical method is less accurate in comparison with the others, the equations of motion can easily be defined using Lagrangian’s formulation (Khan *et al.*, 2020). Among the parameters to be analyzed by the designer in each one of these methods, the relationship between the velocities and accelerations of the input and output bars is one of the main definitions and, through the aid of algorithms, different solutions can be tested and optimized in terms of time spend during the mechanism design.

Meanwhile, the use of computational methods in engineering is increasing once most of its problems can be characterized by mathematics solutions and equations (Elgohary *et al.*, 2014). In tandem with this, problems that would have taken years to work out with the computational methods from 50 years ago can be solved in a few seconds in these days (Pletcher; Tannehill; Anderson, 2013), resulting in greater use of simulation models to solve problems and to aid in decision-making.

In this scenario, with four-bar linkage models established and with the aim to develop a software to simulate these models, the present work uses Python programming language to developed a software with a graphical user interface (GUI) in order to be user friendly, where the user must provide data such as size, mass and inertia of the bars, in addition to the possibility of applying external forces and torques in the system. Subsequently, the user is also able to change the previously assigned parameters and run the simulation again with the updated data, which results in a dynamic way to simulate and optimize the mechanism.

2. FOUR-BAR LINKAGES

The four-bar linkage is one of the simplest and most useful mechanisms in mechanical designs (Mabie; Reinholtz, 1987) and its analysis can be divided, basically, in terms of its kinematics and dynamic parameters.

2.1 Kinematics

Intuitively, the mechanism approached on in this article is composed of four links and joints, as shown in Figure 1. These links are usually named according to the development of the kinematic chain: Link 1 is the ground, normally stationary; link 2 is the driver, which may rotate completely or may oscillate; and link 3 is connected to output link 4, which closes the kinematic chair and is called the rocker if it oscillates or crank if it can also rotate for 360 degrees (Khan *et al.*, 2020; Mabie; Reinholtz, 1987).

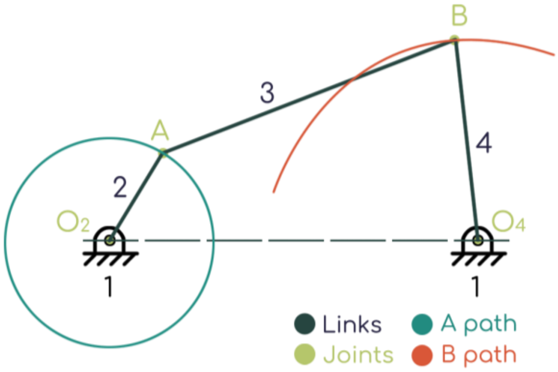


Figure 1. Simplified representation of a four-bar linkage and its motion path

According to Mabie and Reinholtz (1987), link 2 is the driver and if it rotates completely, the mechanism is transforming rotary motion into oscillatory motion, while if the crank oscillates the mechanism only multiplies oscillatory motion. With this, special attention must be taken by the designer when link 2 is only oscillatory, since in this case the mechanism can be locked in extreme positions of the course for reaching the so-called dead points. To identify the type of behavior of the mechanism, the Grashof's law can be applied, as detailed by Asaeikheybari *et al.* (2017).

Another parameter of high interest during the development of four-bar linkage designs, especially when these mechanisms are designed to work close to the dead points, is the transmission angle (γ), defined as the angle between links 3 and 4. Aiming to optimize the force transmission within the mechanism, the ideal situation is that links 3 and 4 should be nearly perpendicular and, if γ devotes from $\pm$ 90 degrees by more than approximately 50 degrees, the linkage tends to bind because of friction in the joints (Mabie; Reinholtz, 1987). Through a purely geometric analysis, the transmission angle can be calculated according to Eq. (1).

$$\gamma = cos^{-1}\left(\frac{O_4A^2 - AB^2 - O_4B^2}{-2.AB.O_4B}\right) \quad (1)$$

In the equation above, AB is the length of link 3, O_4B the length of link 4 and O_4A the shortest distance between points A and O_4. Likewise, during the development of the present paper, O_2A will be used to identify the length of link 2 while O_2O_4 to describe the length of link 1.

As well as the transmission angle, shown in Figure 2, the angles between bars 2, 3 and 4 in relation to the ground (θ) are also fundamental to dictate the behavior of the four-bar linkage. Respectively, the angles θ_2 and θ_4 are defined as the input and output angles of the mechanism, where θ_4 is a function of θ_2. The input angle is usually known to the designer, as well as the length of the links. Consequently, a purely geometric analysis of the assembly of the system allows obtaining the output angle, as shown in Eq. (2).

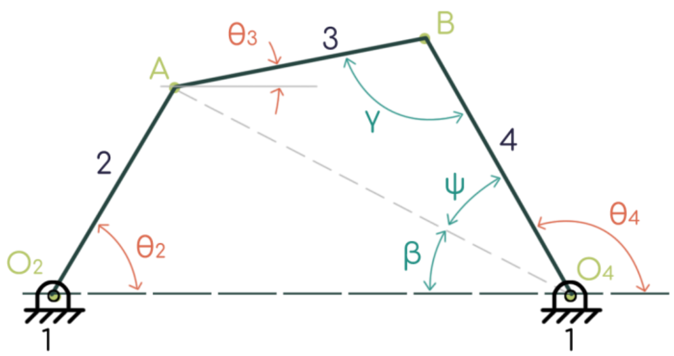


Figure 2. Fundamental angles of the four-bar linkage

$$\theta_4 = 180^o - (\psi + \beta) \tag{2}$$

2.2 Dynamics

In order to analyze the dynamic behavior of a four-bar linkage, it is necessary to define the kinematic parameters such as position of the center of gravity (CG) of the bars and accelerations and velocities of the joints, for example (Norton, 2010). In possession of this information, designers seek to define the torques (T) and forces (F) acting on each of the joints, which can be a somewhat arduous task depending on the geometry and assembly of the bars since these forces vary according to the position of the mechanism. This activity also increases its complexity when external torques and forces are applied on the linkages, shown in Figure 3.

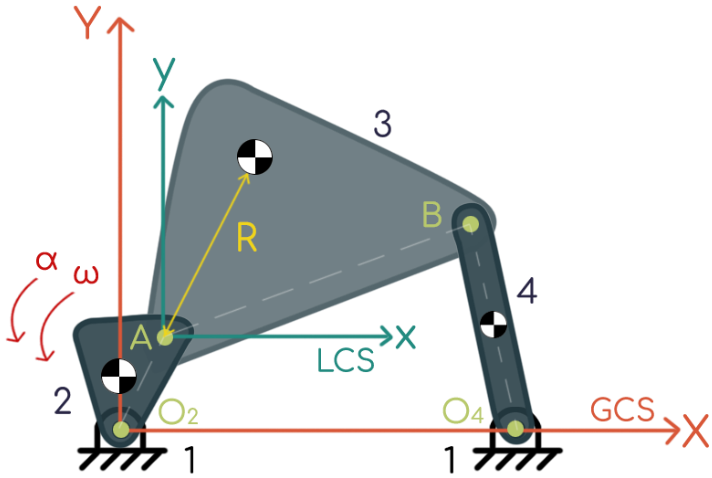


Figure 3. Local and grounded coordinate systems

The first detail to be highlighted in the figure above is the shape of the bars, which have their own angular velocities (ω) and angular accelerations (α). Once the forces are analyzed in the CG of each bar, their geometry has a fundamental role in the task of defining these centers of gravity. Likewise, the mass (m) and the inertia (IG) of each component are parameters that must be known. In this context and among the many approaches to analyzing the mechanism, an analytical methodology widely used is given by the consideration of two coordinate systems: The grounded coordinate system (GCS), normally with the origin in O_2; and the local coordinate system (LCS), defined for each bar along the kinematic chain. It is in this last coordinate system that the distance of the reactions in the joints in relation to the centers of gravity (R) is defined, since these are invariant physical properties of the link. On the other side, the kinematic parameters are defined regarding the GCS (Norton, 2010).

To calculate the dynamic parameters of the four-bar linkage, only the three equations of equilibrium for each of the links are used. As a result, nine equations with nine variables can be represented by a 9x9 matrix. Eq. (3) exemplifies this matrix when a system has a torque applied to bar 4 (T_4) and an external force applied to link 3 (F_P) at a distance R_P from the center of gravity of this same link (Norton, 2010).

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -R_{12_y} & R_{12_X} & -R_{32_y} & R_{32_X} & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & R_{23_y} & -R_{23_X} & -R_{43_y} & R_{43_X} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & R_{34_y} & -R_{34_X} & -R_{14_y} & R_{14_X} & 0 \end{bmatrix} \times \begin{bmatrix} F_{12_X} \\ F_{12_y} \\ F_{32_X} \\ F_{32_y} \\ F_{43_X} \\ F_{43_y} \\ F_{14_X} \\ F_{14_y} \\ T_{12} \end{bmatrix} = \begin{bmatrix} m_2 a_{G_{2x}} \\ m_2 a_{G_{2y}} \\ I_{G_2}\alpha_2 \\ m_3 a_{G_{3x}} - F_{P_x} \\ m_3 a_{G_{3y}} - F_{P_y} \\ I_{G_3}\alpha_3 - R_{P_x}F_{P_y} + R_{P_y}F_{P_x} \\ m_4 a_{G_{4x}} \\ m_4 a_{G_{4y}} \\ I_{G_4}\alpha_4 - T_4 \end{bmatrix} \quad (3)$$

In the matrix above, the subscripts are relative to each joint. For example, F_{12} is relative to the joint between links 1 and 2. Likewise, m_4 is relative to link 4 mass while I_{G3} is relative to link 3 inertia. In parallel, this matrix still has information of the linear acceleration (a_g) of each link.

3. DEVELOPING A SOFTWARE TO FOUR-BAR LINKAGES DESIGN

Python is a renowned programming language that is being used by well-known corporations through past years due its simple, easy and intuitive syntax (Hussain; Khan, 2018; Šimović *et al.*, 2019), besides the countless open-source libraries available by its community. Because of these characteristics and being a high-level programming language, Python is also gaining popularity in academia to teach not only computer science and engineering novices, but many other knowledge areas too (Marowka, 2018), equipping researches with relevant skills that will be primordial in the next years (Tan *et al.*, 2020).

Alongside Python, other open-source programming languages have been gaining ground in recent years, such as R and GO, in addition to the use of consolidated programming languages such as C, C++, C# and JavaScript, for example. However, due to the greater ease of interpreting the language, comparatively speaking, Python was the chose one programming language to achieve the aim of the present work.

Anaconda Python was used to develop the 4bar Solver software (4bS) (Figure 5). Anaconda is an open-source distribution of Python which ship most of available tools and libraries of the language (Duchesnay; Löfstedt, 2015; Snyman; Wilke, 2018) and, from these libraries, the most used here was Tkinter, a free graphical user interface (GUI) framework that is also licensed under a free software license (David, 2018) and is part of the Python standard libraries (Moore, 2018).

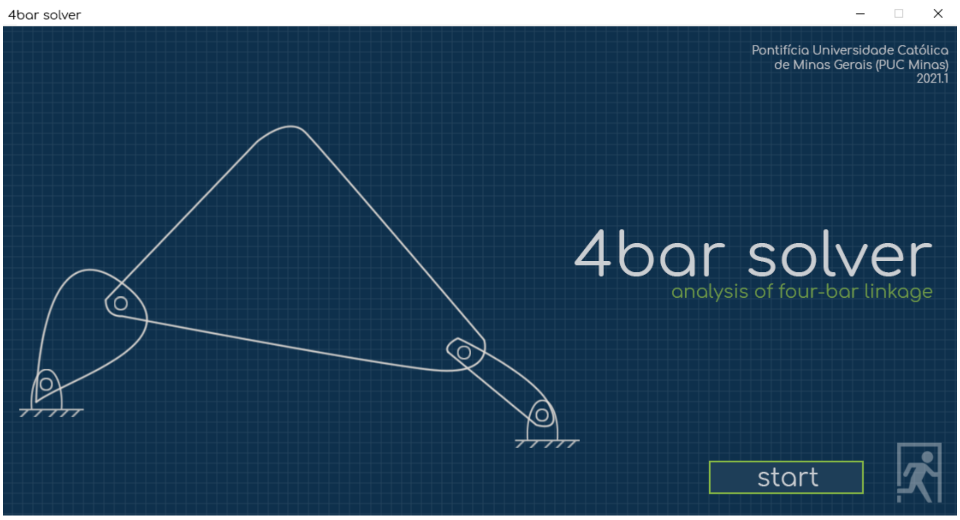
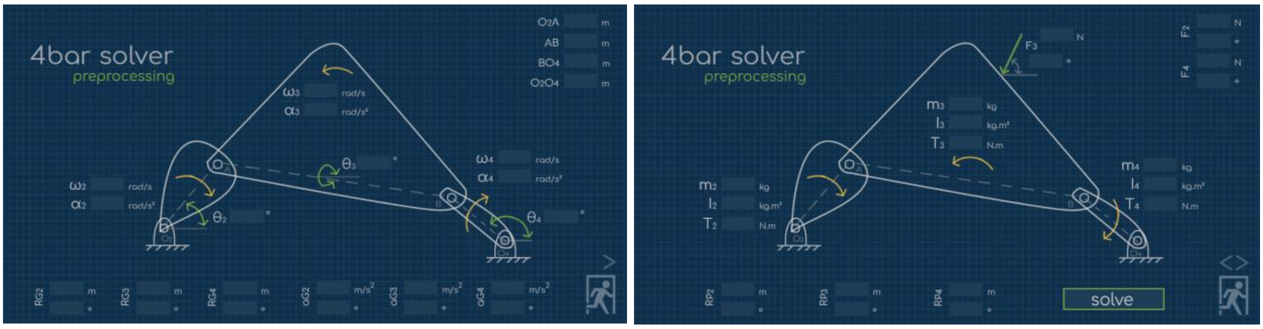


Figure 5. 4bar Solver main page

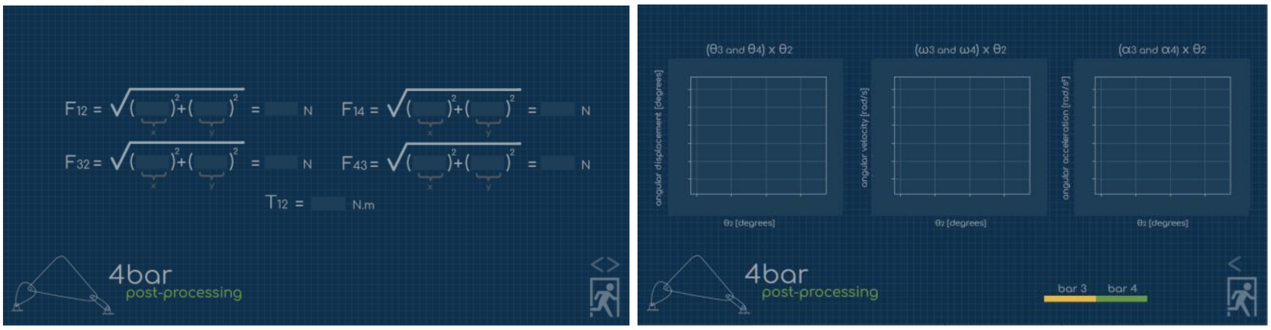
With almost one-hundred code lines, the software presented in the figure above has three levels, basically: The first one where the user can input the data regarding the system, the second one where the forces applied in which linkage are calculated and the third one where the angular displacement, velocity and acceleration plots of the output bar during the complete movement cycle are shown. Also, according to the figure above, after the user selects the start option, data such as length, inertia and mass of the link should be imputed. In addition, it is possible to apply external forces to each of the members of the system. These actions are performed in the preprocessing windows, as shown in Figure 6, where all parameters have been previously presented in this paper.



(a) (b)

Figure 6. 4bar Solver preprocessing. Data related to kinematics (a) and the dynamics of the links (b) are imputed by the user in the blank boxes.

After the preprocessing steps, the user obtains information about the mechanism's behavior on the postprocessing windows (Figure 7). In these, initially the horizontal and vertical components as well as their composition are presented for each joint in the system (Figure 7a). Subsequently, the angular displacement, velocity and acceleration are plotted (Figure 7.b).



(a) (b)

Figure 7. 4bar Solver post-processing. Resulting forces (a) and angular displacement, velocity and acceleration plots (b).

As analyzed in the figures above, the transition by the user in the software windows is possible by buttons located in the lower right corner of the screen. Likewise, there is also an option to return to the main page of the software on all pages. With this, the user has the facility to return to preprocessing steps and modify some input parameter, quickly returning to post-processing to examine whether the results obtained after this modification are more satisfactory than in the initial conditions.

4. VALIDATING THE SOFTWARE BY COMPARING ITS RESULTS WITH ANALYTICAL METHODS

In order to validate the 4bar Solver software, a comparison is made between the software results and solved problems from consolidated bibliography. In problem 11-6.a from Norton (2010), page 601, a four-bar linkage with the configurations and characteristics presented in Figure 8 is proposed.

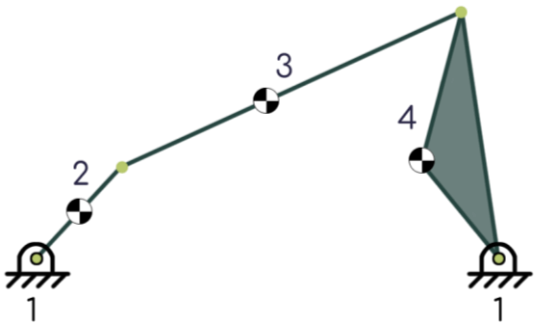


Figure 8. Example 11-6.a from Norton (2010).

To solve this mechanism, the forces, velocities, accelerations and links length are decomposed considering the cartesian plan at the origin of GCS. Subsequently, using the equations of equilibrium and according to what was developed in more detail in Norton (2010), the components can be adjusted in the matrix form previously presented by Eq. (3). Replacing the values provided by the problem 11-6.a, the torque T_{12} is 19.89 N.m, while F_{12} = 616.39 N, F_{32} = 609.53 N, F_{43} = 459.26 and F_{14} = 63.37 N.

Aiming the same results in the 4bar Solver in order to validate this software, the same data considered by the previously mentioned reference are imputed in the software's preprocessing windows (Figure 9) and, as expected, the results obtained through simulation (Figure 10.a) are very close to those found by the analytical method, with the addition of the angular displacement, velocity and acceleration plots (Figure 10.b).

(a)

(b)

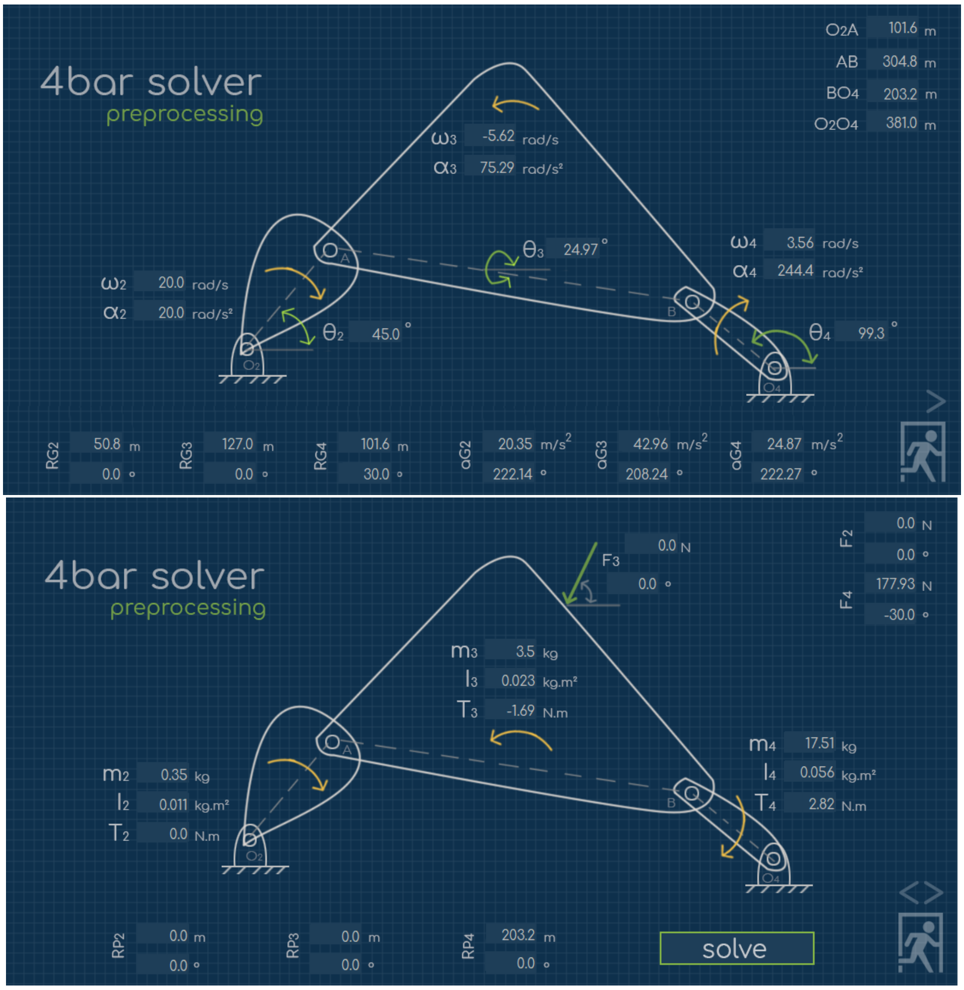
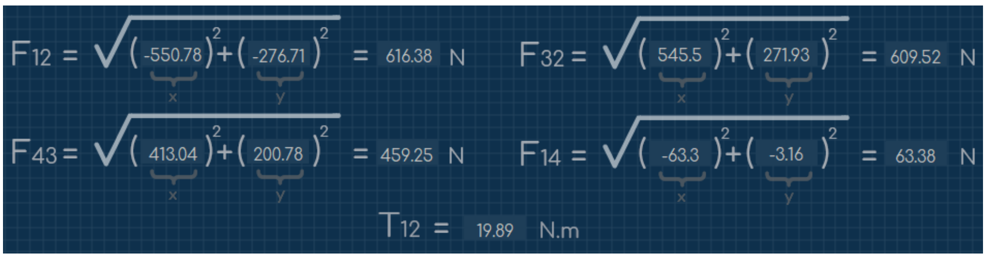


Figure 9. Data imputed in the software for the resolution of the proposed mechanism.

(a)



(b)

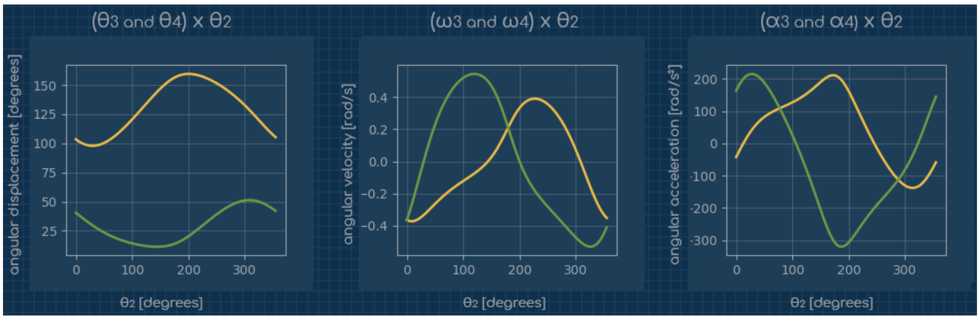


Figure 10. Results obtained after the simulation of the proposed mechanism.

Seeking to corroborate the validation exposed above, other problems of the same bibliography were simulated. As shown in Figure 11 and Table 1, the results via analytical analysis and the software developed by the present work were equivalent, thus validating the 4bar Solver. The step by step of these simulations, however, will not be shown here aiming not to overload the present text, especially since the procedure to achieve the results, either analytically or via software, are very similar to the example presented previously.

(a) (b)

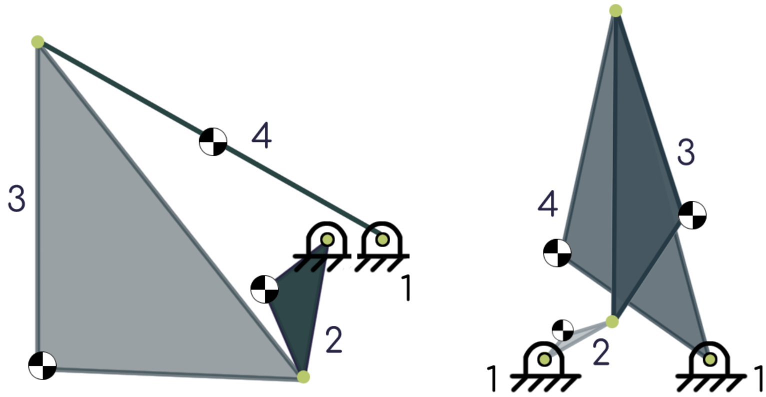


Figure 11. Examples 11-6.c (a) and 11-6.b (b) from Norton (2010). The parameters of the mechanisms can also be consulted in the same reference.

Table 1. Comparison of analytical examples solved with the aid of 4bar Solver.

Parameters	Problem 11-6.c		Problem 11-6.b	
	Analytically	4bar Solver	Analytically	4bar Solver
F_{12} [N]	1317.96	1317.96	2385.35	2385.35
F_{32} [N]	1307.78	1307.78	2366.26	2366.25
F_{43} [N]	1215.62	1215.62	2240.63	2240.63
F_{14} [N]	888.17	888.17	1462.87	1462.88
T_{12} [N.m]	140.63	140.63	143.14	143.14

5. CONCLUSION

Year over year, Computational Science has been having more and more vital role in the Engineering applications. In the present work and regarding this scenario, through Python computer programming language and with the aid of Anaconda, an open-source distribution of Python, a software to four-bar linkages design was developed. With the objective of create a user-friendly graphical interface, the software here developed and named 4bar Solver also used Tkinter, a framework to create graphical user interfaces.

Aiming to validate the results obtained with the 4bS, three examples were simulated with the aid of the software here developed and analytically. In some of the parameters found, there was a small variation of the result in the second decimal place only, which can still be justified by the rounding used in the analytical solution while the software maintains all decimal places of a double type variable. Besides this, inasmuch the results of these two methodologies were equivalent, the software developed by the present paper is considered as validated.

6. ACKNOWLEDGEMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

7. REFERENCES

Albert, Alberto *et al.* Mechanical design and simulation of an onshore four-bar wave energy converter. *Renewable Energy*, v. 114, dec. 2017.
Asaeikheybari, Golnoush *et al.* Dimensional synthesis of a four-bar linkage mechanism via a PSO-based Cooperative Neural Network approach. Maio 2017, [S.l.]: IEEE, maio 2017.
Bureerat, Sujin; SLEESONGSOM, Suwin. Constraint handling technique for four-bar linkage path generation using self-adaptive teaching–learning-based optimization with a diversity archive. *Engineering Optimization*, v. 53, n. 3, 4 mar. 2021.
David, L. *Tkinter GUI Programming by Example*. 1. Ed. Birmingham: Packt Publishing Ltd, 2018.
Duchesnay, e; löfstedt, T. *Statistics and Machine Learning in Python*. 2. Ed. Paris: [s.n.], 2015.
Elgohary, Tarek A. *Et al.* Time Domain Inverse Problems in Nonlinear Systems Using Collocation & Radial Basis Functions. *Computer Modeling in Engineering and Sciences*, jul. 2014.
Hussain, Zahid; Khan, Muhammad Siyab. Introducing Python Programming for Engineering Scholars. *IJCSNS International Journal of Computer Science and Network Security*, v. 18, n. 12, p. 26–33, nov. 2018.
Khan, Sikandar *et al.* Dynamic modeling and analysis of a four-bar mechanism for automobile applications. Jun. 2020, [S.l.]: IEEE, jun. 2020.
Kittisares, Sarin *et al.* Design of knee support device based on four-bar linkage and hydraulic artificial muscle. *ROBOMECH Journal*, v. 7, n. 1, 27 dec. 2020.
Li, Baokun; hao, Guangbo. Nonlinear behaviour design using the kinematic singularity of a general type of double-slider four-bar linkage. *Mechanism and Machine Theory*, v. 129, nov. 2018.
Mabie, Hamilton H.; Reinholtz, Charles F. *Mechanisms and Dynamics of Machinery*. 4. Ed. New Jersey: Wiley, 1987. v. 1.
Marowka, Ami. On parallel software engineering education using python. *Education and Information Technologies*, v. 23, n. 1, 25 jan. 2018.
Moore, A. *Python GUI Programming with Tkinter: Develop responsive and powerful GUI applications with Tkinter*. 1. Ed. Birmingham: Packt Publishing Ltd, 2018.
Norton, Robert L. *Cinemática e Dinâmica dos Mecanismos*. 1. Ed. [S.l.]: AMGH, 2010.
Olsen, Aaron M.; Camp, Ariel L.; Brainerd, Elizabeth L. The opercular mouth-opening mechanism of largemouth bass functions as a 3D four-bar linkage with three degrees of freedom. *Journal of Experimental Biology*, v. 220, n. 24, 15 dez. 2017.
Pletcher, R; Tannehill, J; Anderson, D. *Computational Fluid Mechanics and Heat Transfer*. 3. Ed. Boca Ratón: CRC Press, 2013.
Salah, Mohammad *et al.* Robust Backstepping Control for a Four-Bar Linkage Mechanism Driven by a DC Motor. *Journal of Intelligent & Robotic Systems*, v. 94, n. 2, 27 mai 2019.
Šimović, Vladimir *et al.* Architectural Element of Rose Flower Decorationshown with Python. *Journal of Agronomy, Technology and Engineering Management*, p. 35–42, 2019.
Snyman, Jan A.; Wilke, Daniel N. PRACTICAL COMPUTATIONAL OPTIMIZATION USING PYTHON. [S.l: s.n.], 2018.
Son, L; Huda, S. Impact vibration response attenuation using four-bar linkage landing gear system. *Journal of Physics: Conference Series*, v. 1349, nov. 2019.
Tan, Song Wei Benjamin *et al.* Simple Visual-Aided Automated Titration Using the Python Programming Language. *Journal of Chemical Education*, v. 97, n. 3, 10 mar. 2020.
Wu, Rui; LI, Ruiqin; Bai, Shaoping. A fully analytical method for coupler-curve synthesis of planar four-bar linkages. *Mechanism and Machine Theory*, v. 155, jan. 2021.
Zarkandi, Soheil. A novel optimization-based method to find multiple solutions for path synthesis of planar four-bar and slider-crank mechanisms. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 12 jan. 2021.
Zhang, Yanheng *et al.* Design and Implementation of a Two-Wheel and Hopping Robot with a Linkage Mechanism. *IEEE Access*, v. 6, 2018.

8. RESPONSIBILITY NOTICE

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