DOI: 10.2478/aucts-2025-0011

ACTA UNIVERSITATIS CIBINIENSIS – TECHNICAL SERIES
Vol. 77 2025

AUTOMATION OF AXISYMMETRIC CRIMPING SIMULATIONS FOR PNEUMATIC SUSPENSION COMPONENTS USING PYTHON IN ABAQUS/CAE

POPP Gabriela-Petruța
Faculty of Engineering, Department of Industrial Machines and Equipment, "Lucian Blaga" University, Sibiu, Romania, gabriela.popp@ulbsibiu.ro

POPP Mihai-Octavian
Department of Industrial Machines and Equipment, "Lucian Blaga" University, Sibiu, Romania, mihai.popp@ulbsibiu.ro

CHICEA Anca Lucia
Ph.D., Faculty of Engineering, Department of Industrial Machines and Equipment, "Lucian Blaga" University, Sibiu, Romania, anca.chicea@ulbsibiu.ro

OLEKSIK Mihaela
Faculty of Engineering, Department of Industrial Engineering and Management, "Lucian Blaga" University, Sibiu, Romania, mihaela.oleksik@ulbsibiu.ro

Abstract: *The paper presents the development of an axisymmetric numerical model for simulating the crimping process of pneumatic suspension components and the implementation of a Python script integrated into the Abaqus/CAE environment. Automation through the Abaqus Scripting Interface (ASI) significantly reduces preprocessing time and eliminates repetitive manual operations. The axisymmetric model describes the deformable behavior of metallic and elastomeric components, using CAX4R and CAX4H elements to ensure numerical stability and result accuracy. The script automatically generates geometry, material properties, contact definitions, boundary conditions, and analysis steps based on data provided in a parameterized Excel file. The proposed solution ensures consistency between simulations, allows rapid execution of parametric studies, and facilitates integration into industrial design workflows. The results demonstrate the efficiency of Python scripting in automating FEA-based analyses for complex industrial applications.*

Key words: Abaqus/CAE; finite element analysis; Python scripting; automation; crimping process.

1. Introduction

The automotive industry is continuously evolving, driven by increasingly demanding market requirements and intense competition among manufacturers. The main objectives pursued are the improvement of vehicle dynamics, enhanced safety, and increased comfort during operation. A key contribution to achieving these performances is provided by the suspension system, whose optimization directly influences the vehicle's dynamic behavior and stability [1]. At the same time, both manufacturers and users are concerned with reducing maintenance and operating costs up to the point of major repairs. Reliability and proper operating conditions largely depend on usage conditions and the quality of the components that make up the pneumatic suspension.

Compared to conventional suspension systems, the pneumatic suspension offers multiple advantages due to its ability to adjust stiffness and ride height according to driving conditions and vehicle load. By varying the air pressure in the elastic elements, a nearly constant natural frequency, optimal wheel alignment, and improved handling are achieved. Moreover, pneumatic suspension ensures superior comfort, protects the vehicle structure from vibrations, and facilitates operations such as luggage loading or passenger access, owing to its self-leveling capability [2].

The analysis of pneumatic suspension behavior and the optimization of its design parameters require advanced simulation tools capable of capturing complex interactions between mechanical, pneumatic, and thermodynamic components. For this purpose, Finite Element Analysis (FEA) represents an essential numerical modeling method used to evaluate the distribution of stresses, strains, and vibrations within the suspension components. In modern automotive engineering, FEA simulations are applied not only to study the dynamic behavior of the suspension but also to technological processes such as the

DOI: 10.2478/aucts-2025-0011

crimping of metallic and elastomeric components, with the goal of optimizing process parameters, preventing defects, and improving assembly durability.

Since the design of pneumatic suspensions involves a wide variety of geometries and working conditions, the simulation process often becomes repetitive. In this context, using Python scripts to automate preprocessing, analysis, and postprocessing stages in the Abaqus/CAE environment provides an efficient solution, reducing working time, eliminating human errors, and ensuring result consistency [3]. This paper presents a dedicated Python script developed for automating the simulation of the crimping process in pneumatic suspension components, enabling full automation of the analysis workflow.

2. Model used for the simulation of the crimping process

The analyzed model consisted of four main components (Figure 1): the piston, the bellow, the crimping ring, and the jaw. The bellow, ring, and piston were modeled as deformable parts, while the jaw was defined as an analytical rigid body responsible for transmitting the crimping motion. This approach enables the analysis of the deformable behavior of the assembly in the critical contact region between the ring, bellow, and piston.

The analyzed model consisted of four main components (Figure 1): the piston, the bellow, the crimping ring, and the jaw. The bellow, ring, and piston were modeled as deformable parts, while the jaw was defined as an analytical rigid body responsible for transmitting the crimping motion. This approach enables the analysis of the deformable behavior of the assembly in the critical contact region between the ring, bellow, and piston.

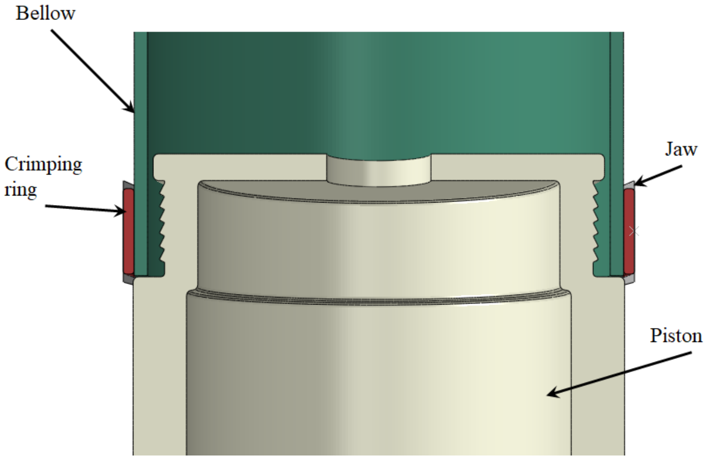

Figure 1 – The assembly used in the simulation

The material properties used in the simulation vary according to the design requirements and are selected to ensure the stiffness, deformability, and sealing capability required for the proper functioning of the system. In general, the metallic components are defined with elastic or elastoplastic behavior, described by Hooke's law in the elastic domain:

$$\sigma = E \cdot \varepsilon \tag{1}$$

where: σ is the stress, E is the Young's modulus and ε is the strain.

When the material enters the plastic domain, the behavior is extended by a hardening law (for example, isotropic), which is generally expressed by the relation:

$$\sigma = \sigma_y + K \cdot \varepsilon_p^n \tag{2}$$

where: σ_y is the yield stress, K is the hardening constant and n is the strain hardening exponent.

The bellow, on the other hand, is modeled as a hyperelastic material capable of reproducing large deformations and the nonlinear behavior specific to elastomers. Its behavior is described by the strain energy function of the Neo-Hookean model:

$$U = C_{10}(\overline{I_1} - 3) + \frac{1}{D_1}(J^{el} - 1)^2 \tag{3}$$

where: C_{10} and D_1 are material coefficients determined experimentally, $\overline{I_1}$ is the first deviatoric invariant of the strain tensor and J^{el} represents the determinant of the elastic strain tensor.

The material parameters are defined in the Property module of Abaqus/CAE and can be adjusted for various combinations of geometries and loading conditions.

DOI: 10.2478/aucts-2025-0011
ACTA UNIVERSITATIS CIBINIENSIS – TECHNICAL SERIES
Vol. 77 2025

The geometry was meshed using axisymmetric elements of type CAX4R (a 4-node axisymmetric quadrilateral element with reduced integration) for the metallic components and CAX4H (a 4-node hybrid axisymmetric quadrilateral element, suitable for incompressible materials) for the bellow, ensuring numerical stability under incompressible material conditions. The global element size was approximately 0.8 mm, with local mesh refinement (bias) applied in the crimping area to obtain a more accurate stress distribution.

The components were positioned so that the bellow was clamped between the crimping ring and the piston, while the jaw applied a radial compression on the ring. The surface interactions were defined using a surface-to-surface contact formulation, with specific friction coefficients for each material pair (metal–metal and metal–rubber), and the finite sliding option enabled to allow large relative displacements between parts.

The boundary conditions were defined as follows:

- The jaw was associated with a Reference Point (RP), allowing only radial displacement, while axial displacements and rotations were constrained.
- The piston was fixed on its lower surface.
- In the crimping step, a controlled radial force was applied to the jaw, calculated based on the geometric dimensions of the assembly and the desired pressure during the process.
- In the uncrimping step, this force was gradually reduced to simulate the retraction of the jaw.

To reduce the setup time for these analyses and ensure consistency between simulations performed for different geometries, the following section presents an automation method using a Python script integrated into the Abaqus/CAE environment.

3. Automation of the analysis process

The simulation of the behavior of pneumatic suspension components involves numerous repetitive stages — defining the geometry, assigning materials, generating the finite element mesh, applying boundary conditions, and running the analysis. To streamline this workflow, the Abaqus/CAE environment provides the possibility of fully automating these operations through the Abaqus Scripting Interface (ASI), which is based on the Python programming language.

The Abaqus/CAE graphical user interface (GUI) is used for model creation and result visualization, while user actions performed within the interface are automatically translated into Python commands. These commands are then transmitted to the Abaqus Kernel, the internal component responsible for processing instructions and building the numerical representation of the model. The Kernel can be accessed either via the GUI or directly through dedicated Python scripts, providing complete control over preprocessing, analysis, and postprocessing stages [4].

The communication architecture between the Abaqus/CAE interface, Python scripts, and the Kernel is shown in Figure 2. Commands may originate from the graphical interface or from a script file and are subsequently interpreted and executed by the Kernel to generate the internal structure of the model.

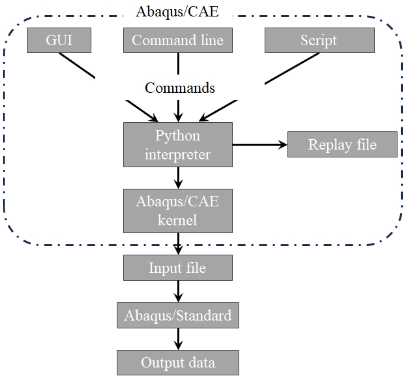

Figure 2 – Interaction scheme between the Abaqus/CAE GUI, Python scripts, and the Kernel [5]

Based on these mechanisms, a custom Python script was developed to automate the preparation process for crimping analyses of pneumatic suspension components. The main goal of this tool is to standardize and accelerate the modeling process, eliminating human errors and ensuring consistency among different simulations.

DOI: 10.2478/aucts-2025-0011
ACTA UNIVERSITATIS CIBINIENSIS – TECHNICAL SERIES
Vol. 77 2025

The input data are provided through a configurable Excel template, which includes the geometrical parameters of the components, the position of the crimping ring (upper or lower), mesh size, material properties, friction coefficients, and loading parameters. Once the spreadsheet is completed, the script automatically generates an axisymmetric preview of the model, allowing the user to verify the input data before running the actual simulation.

The script is launched directly from Excel using a dedicated button ("Create Model"), which triggers a VBA macro. This macro starts a batch file (.bat) that initializes Abaqus/CAE and sends the corresponding Python commands to the Abaqus Kernel, where the complete model is built and configured. As a result, the entire preprocessing workflow is executed automatically, without additional user intervention.

The script generates all components required for the axisymmetric analysis — the bellow, ring, piston, and analytical rigid jaw — defined parametrically according to the values entered in the Excel file. Node sets and surfaces are automatically created for assigning materials, defining boundary conditions, and configuring contact interactions between components. Material properties are assigned based on user selections: the bellow is defined as hyperelastic, while the metallic components are characterized by elastic or elasto-plastic behavior.

The meshing process is also automated, using CAX4H elements for the bellow and CAX4R elements for the metallic components. In the contact area between the ring and bellow, the script applies a local mesh refinement (bias) to achieve a more accurate stress distribution. After meshing, the components are automatically assembled and correctly positioned within the model.

The analysis is divided into three main steps. In the Initial step, contacts and boundary conditions are defined. In the Crimping step, a radial compressive force is applied to the jaws, while in the Uncrimping step, this force is gradually reduced to simulate their retraction. The contacts are defined as surface-to-surface with finite sliding, and friction coefficients are automatically assigned according to the material pairings.

Once the model setup is complete, the script automatically generates the analysis, configures restart file saving, and defines the output data for postprocessing. At the end of execution, Abaqus/CAE saves the configured model and launches the actual simulation.

This automation method significantly reduces the model preparation time, ensures result reproducibility, and enables rapid parametric studies for different geometries, materials, or loading conditions. By integrating Python scripting within the Abaqus/CAE environment, the crimping simulation process becomes more efficient, flexible, and better aligned with modern industrial requirements.

4. Conclusions

The paper presented the development of an axisymmetric numerical model and a Python script dedicated to automating the simulation process of crimping pneumatic suspension components in the Abaqus/CAE environment. The implementation of the Abaqus Scripting Interface (ASI) enabled the elimination of repetitive preprocessing steps, reduced setup time, and improved consistency among analyses performed for different geometries and loading conditions.

The proposed approach facilitates the rapid generation of parameterized models, provides full control over material properties, and ensures the automatic definition of contacts, boundary conditions, and analysis steps. By integrating Python scripting with the Abaqus/CAE graphical interface, the simulation process becomes more efficient and flexible, allowing easy adaptation for various industrial applications.

Furthermore, this methodology can be extended to conduct complex parametric studies, optimize the crimping process, or be integrated into a complete computer-aided design and analysis (CAD–CAE) workflow, offering a robust tool for the analysis and development of pneumatic suspension components.

5. References

1. Kalyan Kumar, R., Suresh Babu, A., "Finite element analysis and experimental study on metal joining by mechanical crimping," Anna University, India, (2012).
2. Xiao, J., Chen, S., Yang, J., Zhang, X., "Mechanical Behavior of Crimping Composite Post Insulator," Buildings, Vol. 13, No. 5, 1275, (2023).
3. NASA Technical Report, Using ABAQUS Scripting Interface for Materials Evaluation and Life Prediction, NASA Center for Aerospace Information, Hanover, MD, (2006).
4. Pletz, M., Efficient FE Modelling Course: Scripting Abaqus CAE using Python, ResearchGate, (2019).
5. ABAQUS Scripting User's Guide, Dassault Systèmes Simulia Corp., Providence, RI, (2023).