XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGENEERING

Automated Python Agent for Accelerating Material Point Method Simulations of Hammer-Driven Conductor Installation in Anura3D

Nuno H.A. Pires[1], Mávyla S. C. Tenório[2], Jennifer M. F. Melo[2], Beatriz R. Barboza[2], João P. L. Santos[2], Rafael Dias[3], Fabio Sawada[3]

[1]*Center of Technology, Federal University of Alagoas*
Av. Lourival Melo Mota, 57072-970, Alagoas/Maceió, Brazil
nuno.pires@ctec.ufal.br
[2]*Laboratory of Scientific Computing and Visualization, Federal University of Alagoas*
Av. Lourival Melo Mota, 57072-970, Alagoas/Maceió, Brazil
mavyla.tenorio@lccv.ufal.br, jennifer.ferreira@lccv.ufal.br, beatriz@lccv.ufal.br, jpls@lccv.ufal.br
[3]*Petrobras*
Av. Horácio Macedo, 21941-915, Rio de Janeiro/Rio de Janeiro, Brazil
fabiosawada@petrobras.com.br, rafael_dias@petrobras.com.br

Abstract. The simulation of hammer-driven conductor installation using the Material Point Method (MPM) requires substantial computational effort and manual intervention, making realistic scenarios with 3,000+ impacts computationally prohibitive. This work presents an automated Python-based agent that optimizes the simulation workflow in Anura3D through GUI automation, file management, and process monitoring. The agent automates the three-stage impact cycle (force application, removal, and consolidation) by detecting simulation states, modifying control files, and launching simulations autonomously. Results from 1,000-cycle tests demonstrate 58% reduction in simulation time (from 168 to 70 hours) with zero operational errors compared with zero errors compared to manual operation, eliminating the human error factor. The system enables uninterrupted execution of extensive simulation campaigns, making comprehensive geotechnical analysis of conductor installation feasible for practical engineering applications.

Keywords: Material Point Method, automation, hammer-driven installation, conductor casing, Python agent.

1 Introduction

Hammer-driven conductor installation in offshore oil and gas wells is a complex engineering process involving significant soil structure interaction and large deformations, conditions under which conventional numerical methods, such as the Finite Element Method (FEM), face severe limitations due to mesh distortion [1]. The Material Point Method (MPM) overcomes these challenges by combining Lagrangian and Eulerian descriptions, allowing the simulation of large strains and displacements without mesh entanglement, and has been successfully applied in geotechnical problems such as offshore monopile installation [2] and free-fall penetrometer deployment [3].

The conductor installation process is critical for providing initial structural support and isolating unconsolidated sediments in subsea wells. Numerical simulations of this process enable detailed analysis of penetration mechanisms and soil response under repeated impact loading. Previous studies have shown that key factors such as impact velocity, soil cohesion, and adhesion strongly influence penetration depth and installation efficiency [4]. For instance, higher impact velocities lead to deeper penetration due to greater energy transfer, while increased soil cohesion significantly reduces penetration capacity, particularly at greater depths.

In the Anura3D MPM environment [5], simulating realistic hammer-driven conductor installations typically involves thousands of impact cycles, each comprising three distinct stages: (1) load application, (2) load removal, and (3) consolidation. In conventional workflows, these stages require manual editing of control files and process supervision, resulting in extremely high labor demand often exceeding 200 hours for 3,000+ impacts and introducing a high potential for human error.

This work builds upon the parametric and geomechanical insights gained in previous research [4] by focusing on the development of a Python-based automation framework that fully integrates with Anura3D. The proposed system eliminates manual intervention through graphical user interface (GUI) automation, intelligent file handling, and computer vision–based simulation monitoring. The automation architecture was implemented in Python, us-

ing PyAutoGUI for GUI automation [6], OpenCV for image-based monitoring [7], and Pyperclip for clipboard management [8]. By enabling continuous, unattended execution of extensive simulation campaigns, the approach significantly reduces computational overhead, enhances reproducibility, and facilitates large-scale parametric studies that were previously impractical due to time and cost constraints.

2 Methodology

2.1 Simulation Framework

Simulations were conducted using Anura3D, an open-source MPM platform particularly suited for large-deformation soil-structure interaction problems. The conductor installation process generates two critical files: geometry (.GOM) and control parameter (.CPS) files. Each hammer impact requires three sequential stages: (1) impact force application via SOLID_TRACTION_B parameter modification, (2) force removal, and (3) extended consolidation period (100 load steps) for system equilibrium.

2.2 Automation Architecture

The Python automation agent was developed using PyAutoGUI for GUI interaction, OpenCV for computer vision-based monitoring, and specialized clipboard management modules. The system architecture comprises five key components:

Initialization Module: Captures simulation parameters (base name, impact count, consolidation time) and locates the simulation directory through automated file system navigation.

File Management System: Implements the `format_number()` function for standardized load step naming (CPS_001, CPS_010, etc.) and extracts current load step numbers through computer vision-based parameter recognition.

Load Parameter Control: The core `martelamento()` function orchestrates complete impact cycles by: (1) validating file format and redirecting to text editors when necessary, (2) incrementing load step numbers, (3) modifying SOLID_TRACTION_B parameters for three distinct states (impact application: `step 0.0 1.0`, removal: `step 1.0 0.0`, consolidation: `off 0.0 0.0`), and (4) launching Anura3D calculations.

Process Monitoring: The `visualizar_cps()` function implements computer vision-based simulation completion detection through continuous GUI element scanning with adaptive waiting periods and error recovery protocols.

Execution Control: The `procurar_arquivos()` function coordinates entire simulation campaigns through nested loops managing both individual impact cycles and three-stage processes within each cycle, incorporating robust error handling and performance monitoring.

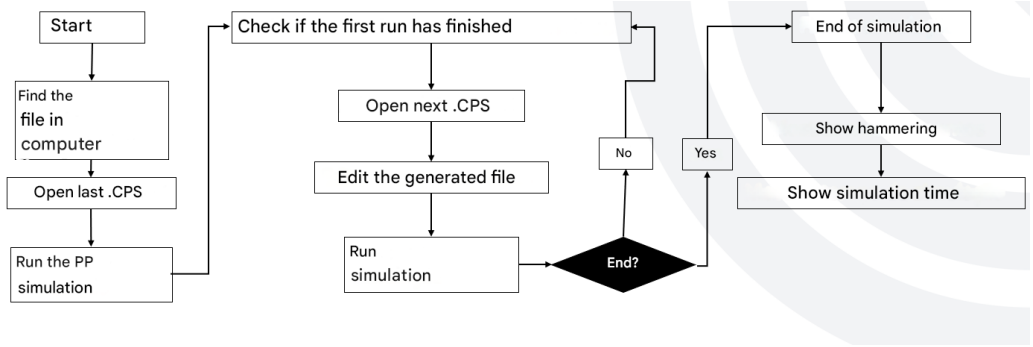


Figure 1. Automated execution workflow showing integration of file management, GUI automation, and simulation control in Anura3D.

3 Results and Discussion

3.1 Performance Evaluation

The automation framework was tested through comprehensive comparison with manual simulation workflows. For 1,000 complete hammer impact cycles, the manual approach required 168 hours and 20 minutes, while the automated system completed identical simulations in 70 hours and 12 minutes, representing a 58.3% reduction in execution time (Table 1). The time savings result from eliminated manual editing delays, removed operator response variability, and continuous 24-hour operation capability.

Table 1. Performance comparison between manual and automated simulation approaches

Methodology	Total Time (h:mm:ss)	Time per Cycle (min) (%)
Manual Process	168:20:00	10.1
Automated Agent	70:12:00	4.2

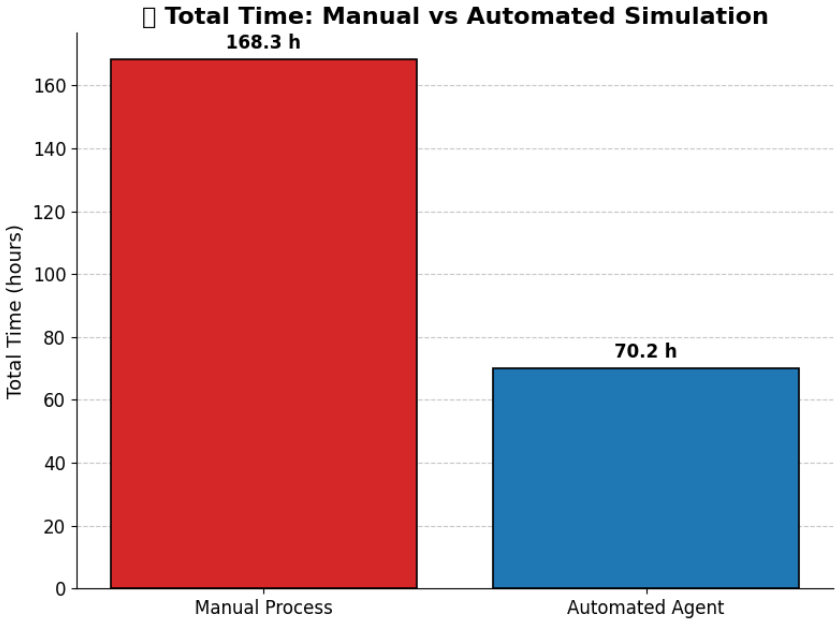


Figure 2. Execution time comparison demonstrating substantial performance improvement through automation.

3.2 Reliability and Error Analysis

The automated system demonstrated superior reliability, completing 1,000 cycles without operational failures. Manual testing recorded three critical errors: incorrect parameter modification, improper file saving, and premature termination due to operator oversight. The automation framework's error handling mechanisms, including GUI validation and process state verification, successfully prevented common failure modes while ensuring simulation continuity under varying system conditions.

4 Conclusions

This work successfully developed and validated a comprehensive Python automation framework that addresses critical limitations in Material Point Method simulations of hammer-driven conductor installation. The automated agent demonstrated exceptional performance improvements, reducing simulation time by 58.3% while eliminating operational errors that commonly occur in manual workflows. The key achievements of this research include: (1) development of a robust automation architecture integrating GUI manipulation, computer vision monitoring, and intelligent file management; (2) successful execution of 1,000-cycle simulation campaigns without operational failures, compared to multiple errors in manual approaches; and (3) enablement of continuous 24-hour operation that makes previously prohibitive large-scale geotechnical analyses economically feasible. The automation framework transforms the practical application of MPM simulations in offshore engineering by eliminating the need for 3,000+ hours of manual supervision in realistic conductor installation scenarios. This breakthrough enables comprehensive parametric studies and optimization analyses that were previously computationally and economically unfeasible. Future work should focus on extending the automation capabilities to handle variable soil conditions, implementing machine learning algorithms for adaptive parameter optimization, and developing cloud-based distributed computing architectures to further accelerate simulation throughput. The integration of real-time monitoring sensors with the automated simulation framework could enable predictive analysis capabilities for field operations. The demonstrated reliability and efficiency improvements establish this automation approach as a valuable tool for advancing computational geotechnics in offshore engineering applications, providing the foundation for more sophisticated decision-support systems in conductor installation planning and execution.

References

[1] K. Abe, K. Soga, and S. Bandara. Material point method for coupled hydromechanical problems. *Journal of Geotechnical and Geoenvironmental Engineering*, 2014.

[2] V. Galavi, M. Martinelli, A. Elkadi, P. Ghasemi, and R. Thijssen. Numerical simulation of impact driven offshore monopiles using the material point method. In *Proceedings of the XVII ECSMGE-2019*, volume 1, pp. 1–8, 2019.

[3] L. Zambrano-Cruzatty and A. Yerro. Numerical simulation of a free fall penetrometer deployment using the material point method. *International Journal of Geomechanics*, vol. 20, n. 5, pp. 1–20, 2020.

[4] N. H. A. Pires, M. S. C. Tenório, J. M. F. Melo, B. R. Barboza, J. P. L. Santos, R. Dias, and F. Sawada. Parametric study of pile installation using a numerical approach with material point method. In *Proceedings of the joint XLV Ibero-Latin-American Congress on Computational Methods in Engineering (CILAMCE-2024)*, Maceió, Brazil, 2024.

[5] A. Yerro, A. Rohe, J. Fern, F. Ceccato, and V. Girardi. *Anura3D MPM Software Tutorial Manual*. Anura3D MPM Research Community. Version 2023, 2023.

[6] A. Sweigart. Pyautogui documentation. Version 0.9.54, Accessed: 2025-08-14, 2024a.

[7] G. Bradski. The opencv library. In *Dr. Dobb's Journal of Software Tools*, 2000.

[8] A. Sweigart. Pyperclip documentation. Version 1.9.0, Accessed: 2025-08-14, 2024b.