People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Abdelhamid Ibn Badis University of Mostaganem

Faculty of Science and Technology

Department of Civil Engineering

Order Number: M2 …... /GC/2026

ACADEMIC MASTER'S THESIS

Field of Study: Civil Engineering

Specialization: Structures

By

ZITOUNI FOUZIA

Title

AI-Assisted Numerical Investigation of the In-Plane Flexibility of Reinforced Concrete Floor Diaphragms with Openings

Defended on 22/06/2026 before the jury composed of:

Before the jury panel composed of:

President	Maliki Mustapha	Professor	Univ. of Mostaganem
Examiner	Djilali Nassira	MCB	Univ. of Mostaganem
	Sadki Nizar	MAA	Univ. of Mostaganem
Guest of Honor	Aida Achour	MCA	Univ. of Sidi Bel Abbes
Supervisor	Bahar Sadek	MCA	Univ. of Mostaganem

AcademicYear 2025/2026

F. ZITOUNI & S. BAHAR

"Engineering is the closest thing to magic that exists in the world"

Elon Musk

"Programming gives engineering the power not only to analyze structures, but also to optimize and improve them."

Dedication

To my beloved mother,
whose unconditional love, sacrifices, prayers, and support have been the foundation of all my achievements.
To my dear sisters,
for their constant encouragement, patience, and presence during every stage of this journey.
To my sister-in-law and her children,
for their kindness, positive energy, and continuous support throughout this work.
To my dear friend Karima,
for her sincere help, support, and encouragement during the realization of this project.
To everyone who believed in me, supported me, and saw in me a promising future,
your trust has always been a source of motivation and strength.
Finally, I dedicate this work to every person striving to achieve their dreams through knowledge, perseverance, and hard work.
May this work inspire others to believe in their potential and never stop learning, creating, and innovating.

Acknowledgements

First of all, thanks to Allah for His guidance, strength, and help throughout the completion of this work.

I would like to sincerely thank my family and loved ones for their continuous support, encouragement, and patience during this journey.

My sincere gratitude goes to my supervisor, Mr. Sadek Bahar, for his guidance and valuable advice throughout this work.

I also thank Mr. Ammamra El Aid and Mme. Guerzou for their support and encouragement.

I would also like to express my gratitude to all my professors and teachers who contributed to my academic journey and shared their knowledge and experience with me throughout the years.

Special thanks to everyone who believed in me and motivated me to continue and improve.

I would also like to thank myself for staying strong throughout this journey, for never giving up, and for always continuing despite the difficulties. Believing in myself was one of the greatest motivations behind the completion of this work.

Finally, this dissertation is not only an academic work, but also a reflection of my passion for engineering, programming, and learning

.

AI-Assisted Numerical Investigation of the In-Plane Flexibility of Reinforced Concrete Floor Diaphragms with Openings.

ABSTRACT

Reinforced concrete floor diaphragms contribute significantly to the lateral behavior of buildings by transferring horizontal loads to the vertical resisting elements. When openings are introduced in floor slabs for architectural or functional purposes, the continuity of the diaphragm is disturbed, which may affect its in-plane stiffness, displacement field, and load-transfer mechanism.

This study investigates the influence of opening location and geometry on the in-plane response of reinforced concrete floor diaphragms. A numerical model was developed in Python based on the Mindlin–Reissner plate theory, which allows transverse shear deformation to be considered. The model was used to analyse slabs with different opening shapes, including circular, square, and rectangular openings.

Two support configurations were examined: slabs supported by columns and slabs supported by shear walls. For each configuration, several lateral loading patterns were considered, namely uniform, triangular, and double-triangular distributions. The position of the opening was then optimized using Particle Swarm Optimization (PSO). The objective was to identify both the position that minimizes the maximum displacement and the position that produces the most unfavorable response. A local refinement using the Nelder–Mead method was also applied to improve the final solution.

The results show that the effect of the opening position depends strongly on the support system and the applied loading pattern. Openings located in critical zones may increase diaphragm flexibility and modify the displacement distribution, while more favorable locations reduce the structural disturbance caused by the opening. The Python results were compared with finite element simulations performed in ABAQUS in order to evaluate the consistency of the proposed numerical approach.

This work provides a numerical procedure for assessing the sensitivity of reinforced concrete floor diaphragms to opening position and offers a practical basis for selecting structurally more appropriate opening locations during the design stage.

Keywords: Reinforced Concrete Diaphragms, Floor Openings, Mindlin-Reissner Theory, Artificial Intelligence (AI), Particle Swarm Optimization (PSO), ABAQUS, In-plane Flexibility.

F. ZITOUNI & S. BAHAR

AI-Assisted Numerical Investigation of the In-Plane Flexibility of Reinforced Concrete Floor Diaphragms with Openings.

RÉSUMÉ

La flexibilité dans le plan des diaphragmes de plancher en béton armé (BA) joue un rôle déterminant dans la performance sismique et dans la répartition des charges latérales des bâtiments à plusieurs étages. L'introduction d'ouvertures dans les planchers, pour des raisons fonctionnelles ou architecturales, modifie considérablement la rigidité du diaphragme, générant des concentrations de contraintes complexes et pouvant créer des vulnérabilités structurelles potentielles. Cette recherche présente une étude numérique assistée par intelligence artificielle (IA), visant à évaluer la réponse structurelle des dalles en BA comportant des ouvertures sous différentes conditions de chargement.

La méthodologie adoptée intègre la théorie des plaques de Mindlin-Reissner, qui prend en compte les déformations de cisaillement transversal, dans un cadre numérique spécialisé développé sous Python. Afin de traiter la complexité non linéaire liée à la détermination des emplacements d'ouvertures les plus influents, une approche hybride fondée sur l'IA est mise en œuvre. Un double algorithme d'optimisation par essaim de particules, PSO, est utilisé pour explorer l'espace de conception. Il permet d'identifier, d'une part, la position optimale minimisant les déplacements horizontaux et, d'autre part, la position critique, correspondant au cas le plus défavorable, maximisant la flexibilité structurelle. Cette double analyse est essentielle pour établir des marges de sécurité plus complètes et mieux comprendre les limites de l'intégrité structurelle. Les résultats obtenus par l'approche d'IA sont ensuite affinés à l'aide de la méthode de Nelder-Mead, puis validés par des simulations par éléments finis réalisées avec ABAQUS.

Une étude comparative a été menée sur deux systèmes d'appui distincts : des dalles sur appuis ponctuels, représentés par des poteaux, et des dalles sur appuis linéaires, représentés par des voiles de contreventement. Différentes géométries d'ouvertures, à savoir circulaires, carrées et rectangulaires, de dimensions fixes, ont été considérées. La robustesse de l'optimisation a été évaluée sous plusieurs scénarios de chargement, notamment des distributions uniformes, triangulaires et doublement triangulaires. Les résultats montrent que l'influence de l'emplacement de l'ouverture dépend fortement à la fois de la configuration des appuis et du mode de chargement. Cette recherche met en évidence l'efficacité de l'intégration des métaheuristiques basées sur l'IA aux théories structurales classiques, afin d'automatiser et d'améliorer le processus de conception des diaphragmes complexes en béton armé.

Mots-clés : Diaphragmes en béton armé, Ouvertures de plancher, Théorie de Mindlin-Reissner, Intelligence Artificielle (IA), Optimisation par essaims de particules (PSO), ABAQUS, Flexibilité dans le plan.

F. ZITOUNI & S. BAHAR

AI-Assisted Numerical Investigation of the In-Plane Flexibility of Reinforced Concrete Floor Diaphragms with Openings.

الملخص

تلعب مرونة أغشية أرضيات الخرسانة المسلحة دورًا حاسمًا في الأداء الزلزالي وتوزيع الأحمال الجانبية للمباني متعددة الطوابق. ويؤدي إدخال فتحات في الأرضيات لأغراض وظيفية أو معمارية إلى تغيير كبير في صلابة الغشاء، مما يخلق تركيزات إجهاد معقدة ونقاط ضعف هيكلية محتملة. تقدم هذه الأطروحة دراسة عددية مدعومة بالذكاء الاصطناعي تهدف إلى تقييم الاستجابة الهيكلية لألواح الخرسانة المسلحة التي تحتوي على فتحات تحت ظروف تحميل متنوعة.

تدمج منهجية البحث نظرية ميندلين-رايسنر للألواح، التي تأخذ في الحسبان تشوهات القص العرضي، في إطار عددي متخصص تم تطويره باستخدام لغة بايثون. ولمعالجة التحدي غير الخطي المتمثل في تحديد مواقع الفتحات الأكثر تأثيرًا، تم تطبيق نهج هجين للذكاء الاصطناعي. تم استخدام خوارزمية تحسين سرب الجسيمات المزدوجة (PSO) للتنقل في فضاء التصميم، وتحديد ليس فقط الموضع الأمثل الذي يقلل الإزاحات الأفقية، ولكن أيضًا الموضع الحرج (أسوأ حالة) الذي يزيد من المرونة الهيكلية. يُعدّ هذا التحليل المزدوج أساسيًا لتحديد هوامش أمان شاملة وفهم حدود السلامة الإنشائية. وقد تمّ تحسين نتائج الذكاء الاصطناعي باستخدام طريقة نيلدر-ميد، والتحقق من صحتها من خلال محاكاة العناصر المحدودة عالية الدقة باستخدام برنامج ABAQUS.

أُجريت دراسة مقارنة بين نظامي دعم مختلفين: البلاطات المدعومة بنقاط (الأعمدة) والبلاطات المدعومة بخطوط (جدران القص)، مع مراعاة أشكال هندسية مختلفة للفتحات (دائرية، مربعة، ومستطيلة) بأبعاد ثابتة محددة. علاوة على ذلك، تمّ اختبار متانة التحسين في ظل سيناريوهات تحميل متعددة، بما في ذلك التوزيعات المنتظمة والمثلثية والمثلثية المزدوجة. تُظهر النتائج أن تأثير موقع الفتحة شديد الحساسية لكل من تكوين الدعم ونمط التحميل. يُبرز هذا البحث فعالية دمج الخوارزميات الاستكشافية القائمة على الذكاء الاصطناعي مع النظريات الإنشائية الكلاسيكية لأتمتة وتحسين عملية تصميم الحواجز الخرسانية المسلحة المعقدة.

الكلمات المفتاحية: أغشية الخرسانة المسلحة، فتحات الأرضيات، نظرية ميندلين-رايسنر، الذكاء الاصطناعي (AI)، تحسين سرب الجسيمات (PSO)، ABAQUS، المرونة في المستوى.

F. ZITOUNI & S. BAHAR

AI-Assisted Numerical Investigation of the In-Plane Flexibility of Reinforced Concrete Floor Diaphragms with Openings.

LIST OF FIGURES

F. ZITOUNI & S. BAHAR

LIST OF TABLES

AI-Assisted Numerical Investigation of the In-Plane Flexibility of Reinforced Concrete Floor Diaphragms with Openings.

TABLE OF CONTENTS

F. ZITOUNI & S. BAHAR

GENERAL INTRODUCTION

Reinforced concrete slabs are among the main structural components of buildings. In addition to carrying gravity loads and transferring them to beams, columns, or walls, floor slabs may also act as horizontal diaphragms when the structure is subjected to lateral actions. In this role, the slab contributes to transferring horizontal loads toward the vertical resisting elements, such as columns and shear walls. Therefore, the in-plane stiffness of the floor diaphragm has a direct influence on the distribution of lateral forces and on the global response of the structure.

In practical building design, openings are often introduced in floor slabs to satisfy architectural and functional requirements, such as staircases, elevator shafts, ventilation ducts, and technical services. Although these openings are usually necessary, they create discontinuities in the slab and may locally reduce its stiffness. As a result, the in-plane behavior of the diaphragm can be affected, especially when the opening is located near zones where load transfer is important. The influence of an opening is not only related to its size and shape, but also to its position and to the type of support system used in the structure.

The assessment of floor diaphragms with openings remains a relevant issue in structural analysis because the load-transfer mechanism is influenced by several parameters at the same time. These include the geometry of the opening, the boundary conditions, the stiffness of the supporting elements, and the distribution of the applied lateral loads. For this reason, simplified assumptions may not always be sufficient to describe the actual behavior of the diaphragm. Numerical modelling provides a practical way to study these effects and to compare different configurations under controlled conditions.

In this work, a numerical model was developed using Python to study the in-plane response of reinforced concrete slabs containing a single opening. The formulation is based on the Mindlin–Reissner plate theory, which takes transverse shear deformation into account. Three opening shapes were considered: circular, square, and rectangular. The analysis was carried out for two support configurations: slabs supported by columns and slabs supported by shear walls. In addition, different lateral loading patterns were applied, including uniform, triangular, and double-triangular distributions.

The position of the opening was treated as a design variable. Particle Swarm Optimization (PSO) was used to search for the opening location that leads to the lowest maximum displacement. The same procedure was also used to identify the most unfavorable location, corresponding to the highest displacement response. This dual search makes it possible to distinguish between favorable and critical zones within the slab. A local refinement step based on the Nelder–Mead method was then applied to improve the final position obtained by the optimization process.

To evaluate the consistency of the Python model, selected cases were compared with finite element simulations performed using ABAQUS. The comparison focused mainly on the displacement response obtained for the optimal and critical opening positions. This validation step was used to assess whether the simplified Python-based model can reproduce the main trends observed in a more detailed finite element model.

- The main objectives of this research are as follows:
- To study the influence of opening position and opening shape on the in-plane behavior of reinforced concrete floor diaphragms.
- To compare the response of column-supported and shear-wall-supported slabs.
- To evaluate the effect of different lateral loading patterns on the displacement response.
- To apply PSO and Nelder–Mead optimization for identifying optimal and critical opening locations.
- To compare the Python-based numerical results with ABAQUS simulations.

Thesis Structure

To achieve the research objectives, this thesis is organized into five main chapters as follows:

- ✓ **Chapter I: Generalities on Diaphragms and Openings**

This chapter provides the fundamental structural context, exploring the behavior of reinforced concrete floor diaphragms and the mechanical impact of introducing openings on in-plane stiffness and load distribution.

- ✓ **Chapter II: Optimization Methods and Metaheuristics**

Focuses on the transition from classical optimization to Artificial Intelligence (AI) techniques. It details the principles of swarm intelligence, specifically the Particle Swarm Optimization (PSO) algorithm, and its efficiency in navigating complex structural design spaces.

- ✓ **Chapter III: Python in Civil Engineering**

Dedicated to the computational implementation of the research. It describes the development of the numerical framework in Python, utilizing Mindlin-Reissner Plate Theory to account for transverse shear deformations.

- ✓ **Chapter IV: Numerical Modeling and Finite Element Analysis**

Covers the simulation phase using ABAQUS. It details the finite element discretization, material properties, and the application of loading patterns used to validate the AI-assisted results.

- ✓ **Chapter V: Results and Discussion**

Presents a comparative analysis of the findings. It highlights the differences between column-supported and shear wall-supported slabs, evaluating how the diaphragm's flexibility responds to various opening geometries and locations under the studied load cases.

Finally, the General Conclusion summarizes the key findings and provides recommendations for integrating AI-driven optimization into practical structural design.

CHAPTER I: GENERALITIES ON DIAPHRAGMS AND OPENINGS

I.1 Definition and Role of Floor Diaphragms

A floor diaphragm is a horizontal structural element that ensures the transfers of lateral forces to the vertical load-resisting elements such as columns, shear walls, and load-bearing walls. It acts as a rigid or semi-rigid system capable of distributing horizontal loads generated by wind, seismic actions, or other lateral forces. Through this mechanism, the diaphragm contributes significantly to the overall stability and structural integrity of buildings.

Floor diaphragms may take different structural forms depending on the design requirements and construction techniques. They can consist of solid reinforced concrete slabs, ribbed slabs, or composite floor systems. Regardless of the structural configuration, their fundamental role is to provide an efficient path for the transmission of horizontal forces throughout the structural system.

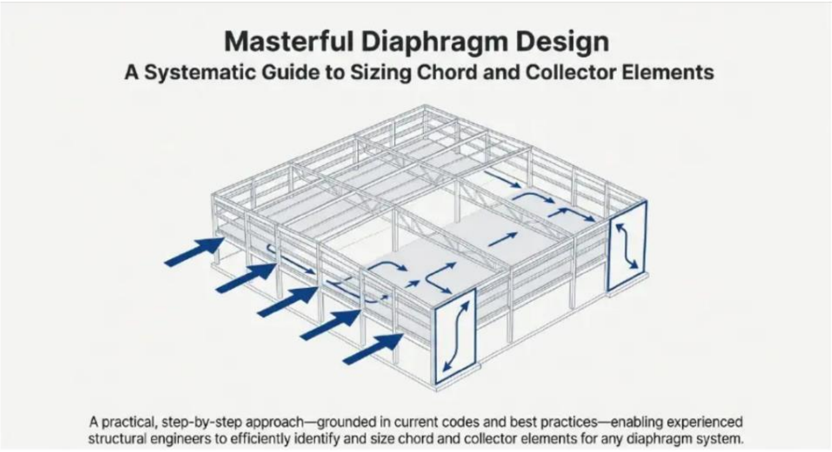


Figure 1: Load transfer mechanism in floor diaphragms.

Role of the floor diaphragm

The floor diaphragm performs several essential functions within a structural system:

- Distribution of horizontal forces among the different vertical lateral load-resisting elements.
- Ensuring the global structural integrity under seismic loading.
- Limiting differential displacements within the plane of the floor in order to maintain an appropriate distribution of internal forces.

1.2 In-Plane Behavior under Horizontal Actions

Beyond providing a horizontal surface within buildings, floor slabs also play an important structural role when subjected to horizontal actions such as wind and seismic loads. Under these conditions, the diaphragm behaves similarly to a very stiff horizontal beam that transfers lateral forces to the vertical resisting elements.

The in-plane behavior of a diaphragm depends on several factors, including its in-plane stiffness, geometric configuration, the distribution of vertical load-resisting elements, and the presence of openings within the slab. When subjected to horizontal forces, the diaphragm

undergoes in-plane deformations that influence the way lateral loads are transmitted to the structural elements responsible for resisting them.
A regular and homogeneous diaphragm behavior promotes a more uniform distribution of lateral forces within the structural system, thereby improving the overall structural performance.

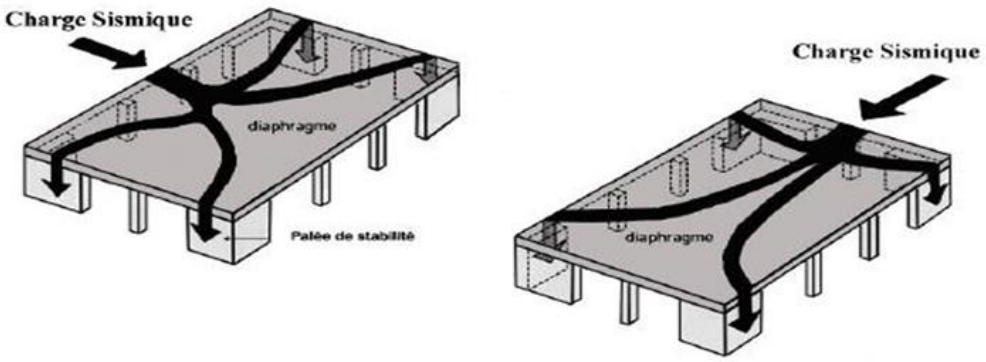


Figure 2: Transmission of horizontal seismic loads to vertical bracing elements. *[1]*

I.3 Hypothesis of Rigid and Flexible Diaphragms

In structural analysis, the actual behavior of diaphragms is generally classified into two main types: **rigid diaphragms** and **flexible diaphragms**.

1. Rigid diaphragm :

- Horizontal in-plane deformations are considered negligible.
- Horizontal forces are distributed almost instantaneously among the load-bearing elements.
- The rigidity assumption simplifies structural analysis, particularly in analytical methods and simplified models.

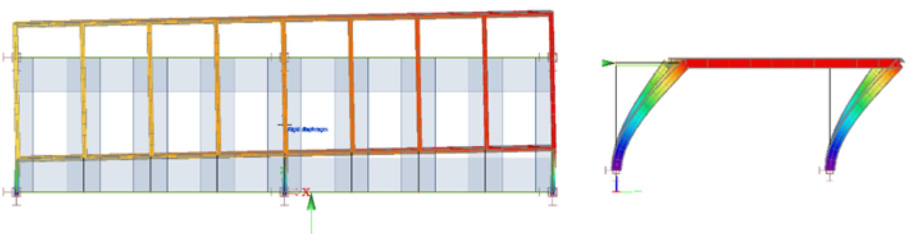


Figure 3: Schematic illustration of the behavior of a rigid diaphragm.

Source : SCIA Engineer – Technical Documentation, Diaphragms.

2. Flexible diaphragm:

- Horizontal deformations are significant and must be considered in the calculation of forces.
- The distribution of forces toward the supports depends on the flexibility of the diaphragm, which may lead to stress concentrations and larger displacements.

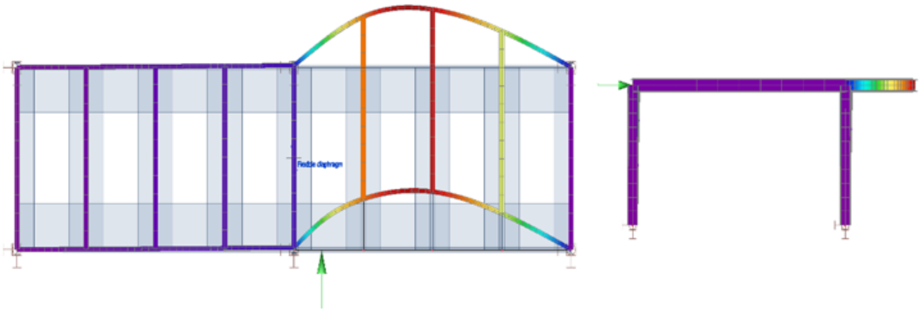


Figure 4: Schematic illustration of the behavior of a flexible diaphragm.

Source : SCIA Engineer – Technical Documentation, Diaphragms.

The choice between these assumptions depends on several factors, including the geometry of the slab, its continuity, the materials used, and the presence of openings.

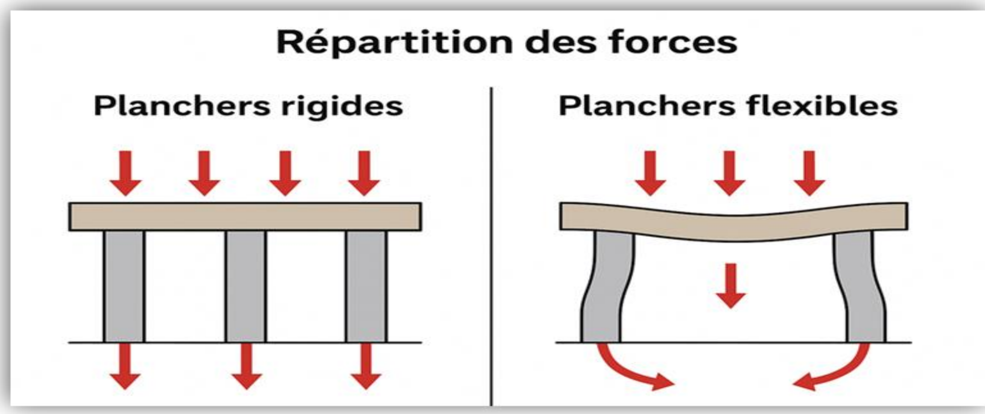


Figure 5: Distribution of forces in rigid and flexible floor diaphragms.

I.4 Openings in Floor Diaphragms

Openings in floor slabs are commonly introduced in building structures to satisfy architectural and functional requirements. There are various reasons for providing openings in floor slabs. Including the provision of space for staircases, elevator shafts, ventilation ducts, and technical service installations. Depending on their function and geometry, slab openings may have different shapes and sizes, ranging from circular, square, to rectangular shapes. While these openings are often necessary in building design, they introduce discontinuities in the slab that significantly influence the structural behavior of the floor diaphragm.

Floor openings are found to reduce the overall stiffness of floor diaphragms. As a result, the distribution of internal forces within the diaphragm may change, causing stress concentrations around these openings.

Large floor openings located near critical structure zones may also increase the in-plane flexibility of the diaphragm.

Therefore, the analysis of slabs with openings requires special attention, particularly in structures subjected to lateral loads such as wind and seismic actions. Proper modeling and evaluation of these openings are essential to ensure adequate structural performance and safety.

I.5 Types of Openings in Floor Slabs

Openings in slabs can be classified according to several criteria:

- **According to shape:** circular, square, rectangular, or irregular openings.
- **According to size:** small opening (for ducts and pipes) and large openings (for stairs or elevators).
- **According to location:** central openings, edge openings, or openings near structural supports.

Each type of opening may influence the structural behavior of the diaphragm differently, depending on its geometry and position within the slab.

I.6 Classification of Openings in Floor Diaphragms According to Seismic Codes

In seismic design, the presence of openings within the floor diaphragms is considered to have a considerable influence on the overall behavior of the structure subjected to lateral forces.

According to the seismic design codes, the openings within the floor diaphragms are classified on the basis of the size and the influence of the openings on the floor diaphragms.

According to seismic design provisions, openings can generally be categorized into small openings and large openings. Small openings, typically used for pipes or service ducts, usually have a negligible influence on the global behavior of the diaphragm and may be ignored in simplified structural models. In contrast, large openings such as those required for staircases, elevators, or large service shafts may significantly reduce the diaphragm stiffness and change the distribution of lateral forces within the structural system.

Large openings can introduce discontinuities in the diaphragm, which may lead to irregular force paths, stress concentrations, and increased in-plane deformations. In some cases, these openings may cause the diaphragm to behave as a **semi-rigid or flexible diaphragm**, especially when the opening size becomes comparable to the slab dimensions.

For this reason, seismic design codes recommend special attention when modeling diaphragms containing significant openings. In many cases, the openings must be explicitly considered in structural analysis to ensure an accurate representation of the load transfer mechanism and the global structural behavior.

I.7 Effect of Openings on In-Plane Stiffness

The presence of openings (such as staircases, elevators, and technical shafts) is considered one of the important factors affecting the in-plane stiffness of floor diaphragms. This modification directly influences the distribution of horizontal forces among the lateral load-resisting elements. In addition, openings in floor slabs may lead to stress concentrations around the discontinuities and create zones of structural weakness.

The effect of openings in floor diaphragms depends mainly on their **size, shape, and location**. Openings located near the center of the slab or having large dimensions may significantly reduce the diaphragm's ability to transfer horizontal forces. In such cases, the structural behavior may become similar to that of a flexible diaphragm.

Therefore, the explicit consideration of slab openings in numerical modeling is essential for a realistic evaluation of the dynamic response of buildings. This issue is particularly important in modern constructions, where architectural requirements often impose large open spaces and significant floor openings.

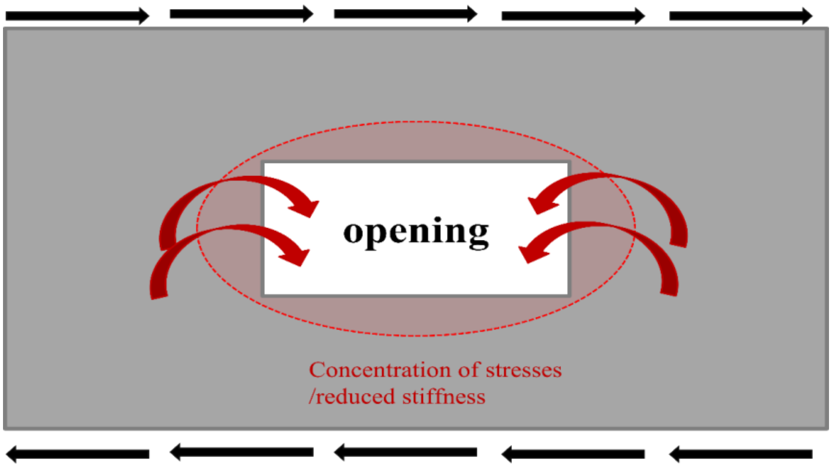


Figure 6: Effect of openings on the in-plane stiffness of the floor diaphragm.

I.8 Influence of the Position of Openings in Floor Diaphragms

The position of openings within a floor diaphragm plays a significant role in the structural behavior of slabs subjected to lateral loads. Depending on their location, openings may have different impacts on the distribution of internal forces and the overall stiffness of the diaphragm. Openings located near the **center of the slab** generally have a smaller influence on the global structural behavior, as the load transfer paths remain relatively uniform. However, when openings are positioned near **structural supports**, such as columns or shear walls, they may significantly disturb the load transfer mechanism and reduce the effectiveness of the diaphragm in distributing lateral forces.

Similarly, openings placed close to the **edges of the slab** may lead to local stress concentrations and increased in-plane deformations. In certain configurations, unfavorable positioning of openings can result in irregular force distribution and increased structural flexibility.

For these reasons, the location of openings should be carefully considered during the design stage. In modern structural engineering practice, numerical modeling and optimization techniques are often employed to determine the most suitable position of slab openings while minimizing their negative impact on diaphragm stiffness and structural performance.

I.9 Research Gap and Motivation

Although numerous studies have investigated the structural behavior of floor diaphragms and the influence of slab openings, most existing research mainly focuses on qualitative assessments or simplified analytical approaches. In many practical cases, the location of openings within slabs is determined primarily based on architectural requirements rather than structural optimization considerations.

However, the position of openings can significantly influence the in-plane stiffness of the diaphragm and the distribution of lateral forces within the structural system. Inappropriate placement of openings may lead to increased displacements, stress concentrations, and reduced structural performance under lateral loading conditions. By defining both optimal and critical zones, this research provides a dual-perspective safety map that assists structural engineers in negotiating opening placements with architects.

Despite the growing interest in computational methods in structural engineering, limited research has focused on the optimization of opening locations in floor diaphragms using artificial intelligence-based techniques. In particular, the application of metaheuristic optimization algorithms combined with simplified numerical models remains relatively unexplored in this context.

Therefore, the present study aims to address this gap by developing a simplified numerical model implemented in Python and applying metaheuristic optimization techniques, particularly Particle Swarm Optimization (PSO), to determine the optimal location of slab openings while minimizing the in-plane displacement and preserving the diaphragm stiffness.

I.10 Literature Review of Existing Studies

The Research Flow Diagram presents a summary of the relevant studies that focus on reinforced concrete slabs that have openings, flexibility in the diaphragms, and the use of numerical analysis and artificial intelligence. The gap in the research is identified by showing that very few studies have included all these components in an assessment of the in-plane flexibility in floor diaphragms that have openings.

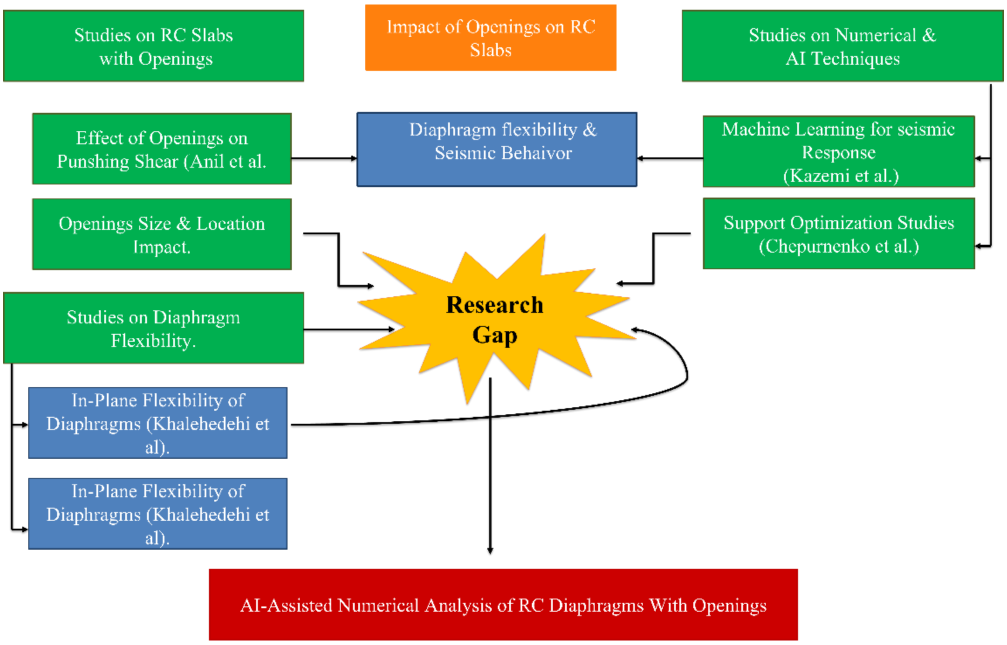


Figure 7: Research Flow Diagram.

The diagram presents the interconnections among studies on reinforced concrete slabs with openings, diaphragm flexibility, and computational or AI-based techniques. It highlights the convergence of these research streams and the gap in literature regarding in- plane flexibility of reinforced concrete floor diaphragms with openings, which is addressed in this investigation.

a. Studies on Reinforced Concrete Slabs with Openings

Several investigations have been carried out on the effects of openings on reinforced concrete slabs. Openings are usually necessary in reinforced concrete slabs to accommodate openings such as staircases, elevator shafts, and ducts.
Punching shear behavior investigations have shown that the size, shape, and location of openings have a significant impact on the load-carrying capacity of the slab.
Openings located close to columns have a significant impact on the stress concentration.
Larger openings have a significant impact on the stiffness of the slab.
This shows the importance of incorporating openings into the analysis of reinforced concrete slabs.

b. Studies on Diaphragm Flexibility

Research on diaphragm behavior has highlighted the importance of in-plane flexibility in distributing lateral loads.
While many analyses traditionally assume rigid diaphragm action, studies indicate that this assumption may lead to inaccuracies, particularly for large or irregular floor systems.
In-plane flexibility is further influenced by the presence of openings, which reduces diaphragm stiffness and changes force distributions.
Comparative analyses between rigid and flexible diaphragms have also revealed significant differences in seismic force transmission within multi-story structures.

c. Studies on Numerical Modeling and AI Techniques

Several investigations have been carried out on the use of numerical models to predict the response of reinforced concrete structures.
Finite element models have been widely used to model reinforced concrete slabs under varying load conditions.
AI techniques have been shown to be effective in predicting the response of reinforced concrete structures.
AI techniques have been used to optimize the placement of the supports to minimize the stress concentration.

Table 1: Summary of Previous Studies on Reinforced Concrete Slabs with Openings, Diaphragm Flexibility, and Support Optimization.

Study	Objective	Method	Main Findings	Limitations
Effect of opening shape, size and location on the punching shear behavior of RC flat slabs.	Investigate the influence of opening shape, size and location on punching shear behavior.	Experimental and Numerical Analysis.	Openings near columns significantly reduce punching shear capacity and increase stress concentration.	Focused on punching shear behavior without considering the overall in-plane diaphragm stiffness or its influence on support locations.
Khajehdehi et al. – In-Plane Flexibility of RC Floor Diaphragms with Openings.	Study the in-plane flexibility of RC floor diaphragms with openings.	Finite Element Analysis.	Openings reduce diaphragm stiffness and affect the distribution of internal forces.	Did not incorporate AI-based analysis or optimization of support locations.
Kazemi et al. – Machine learning-based seismic response assessment.	Evaluate seismic response and performance of RC buildings using machine learning.	Machine Learning Models.	AI methods can accurately predict seismic performance of structures.	Focused on global building response without analyzing the impact of openings on diaphragm behavior.
Anil et al. – Effect of opening size and location on punching shear behavior of two-way RC slabs.	Investigate the effect of opening size and location on punching shear capacity.	Experimental testing.	Larger openings significantly reduce punching shear resistance.	Focused on punching shear only, did not consider in-plane diaphragm stiffness or support optimization.
Hassan – Comparative study between rigid and flexible diaphragms.	Compare rigid and flexible diaphragm assumptions in seismic analysis of buildings.	Numerical Structural Analysis.	Diaphragm flexibility influences the distribution of seismic forces.	Did not study the effects of openings or their influence on optimal support positions
Chepurnenko et al. – Optimizing the location of supports under a monolithic floor slab.	Determine optimal support locations under RC slabs.	Numerical modeling and optimization techniques.	Optimal support locations reduce slab deformation and stresses.	Focused on support optimization without considering the effects of openings on slab stiffness and in-plane diaphragm behavior.

Despite extensive studies on individual aspects of slab openings, diaphragm flexibility, and computational or AI-based analysis, limited research integrates these elements to assess the in-plane flexibility of reinforced concrete floor diaphragms with openings using AI-assisted numerical approaches. This gap highlights the need for integrated investigations combining structural modeling, diaphragm behavior, and artificial intelligence.

I.11 Conclusion

This chapter has established the fundamental role of floor diaphragms in the lateral stability of structures and explored the significant impact of openings on their in-plane stiffness. The review of existing literature and seismic codes highlights a critical research gap: while the presence of openings is known to alter load distribution, their specific placement is rarely optimized for structural efficiency. This motivation sets the stage for the present study, which moves beyond qualitative descriptions toward a numerical optimization approach. To achieve this, the following chapter will detail the mathematical and algorithmic tools required to automate the search for optimal and critical opening locations.

CHAPTER II: OPTIMIZATION METHODS AND METAHEURISTICS.

II.1 Optimization in Civil Engineering

Optimization in civil engineering is a fundamental field focused on improving structural performance, economic efficiency, and safety while ensuring compliance with technical, environmental, and regulatory constraints. It consists of searching for the optimal solution among a set of feasible alternatives based on one or several objective criteria.

In the context of structural engineering, optimization aims to minimize or maximize certain parameters such as structural weight, displacements, stresses, material cost, or energy consumption, while ensuring that requirements related to strength, stability, and durability are satisfied.

Traditionally, optimization methods relied mainly on analytical and deterministic approaches. However, with the emergence of increasingly complex structures characterized by irregular geometries, composite materials, and multiple loading conditions, more advanced numerical and intelligent optimization techniques have been developed.

Recently, artificial intelligence and metaheuristic optimization techniques such as **Genetic Algorithms (GA)**, **Particle Swarm Optimization (PSO)**, and **Artificial Neural Networks (ANN)** have been widely adopted in the field of civil engineering. These methods offer strong exploration capabilities within the search space, allowing optimal or near-optimal solutions to be obtained with high computational efficiency.

Within the context of this study, optimization represents a key aspect for improving the numerical analysis of the in-plane behavior of floor diaphragms with openings, by combining **finite element modeling tools** with **artificial intelligence-based optimization techniques**.

II.2 General Formulation of an Optimization Problem (Variables, Constraints, Objective Function)

In order to solve a structural optimization problem, it is necessary to establish a rigorous formulation representing the physical reality of the system. This formulation enables the optimization algorithm (such as **PSO** or **GA**) to explore the search space and identify the optimal solution.

Three main components are essential in the formulation of an optimization problem:

II.2.1 Decision Variables (Design Vector)

In this work, the variables represent the parameters that the optimization algorithm must explore. These parameters correspond to the design choices that the algorithm is allowed to modify in order to minimize the objective function.

In the framework of this study, the design vector **X** is defined by the coordinates of the center of the opening and is expressed as:

$$\mathbf{X} = \{x, y\}$$

Where:

x: position of the opening along the slab length.

y: position of the opening along the slab width.

Remark: For cases involving variable shapes (rectangle, square, circle), additional variables may be introduced, such as the radius **(R)** or the side lengths **(a, b)**.
Decision variables correspond to the parameters of the system that can be modified in order to optimize the expected results. In civil engineering, decision variables may include geometric dimensions, cost, weight, mechanical properties of materials, and the arrangement of structural components.

II.2.2 Objective Function (Cost Function)

The objective function represents the quantity to be optimized, such as minimizing displacements or stresses, or maximizing stiffness and load-carrying capacity.
In this study, the objective is to preserve the maximum stiffness of the diaphragm despite the presence of an opening. The objective function f (X) is defined as:

$$\textbf{Minimize } f(X) = U_{max}(x, y)$$

Where U_{max} represents the maximum displacement or deformation obtained from the solution of the matrix system:

$$[K]\{U\} = \{F\}$$

Furthermore, the optimization framework is implemented as a dual-search process. Beyond simply minimizing displacements to find the optimal opening position, the algorithm is also tasked with identifying the global maximum. This identifies the "critical zone" or the worst-case location for an opening, providing a comprehensive safety map for the structural diaphragm.

II.2.3 Problem Constraints

Constraints define the search space to ensure that the obtained solution remains physically and technically acceptable.

❖ **Domain constraints (Box constraints):**

The opening must remain within the slab domain:

$$x_{min} < x < x_{max}$$

❖ **Integrity constraints:**

The opening must not overlap with support zones such as columns or shear walls.

❖ **Geometric constraints:**

A minimum distance must be respected between the edge of the opening and the boundary of the slab.
In the Python implementation, a **penalty function** is used to directly reject any particle (position) that does not satisfy these constraints.

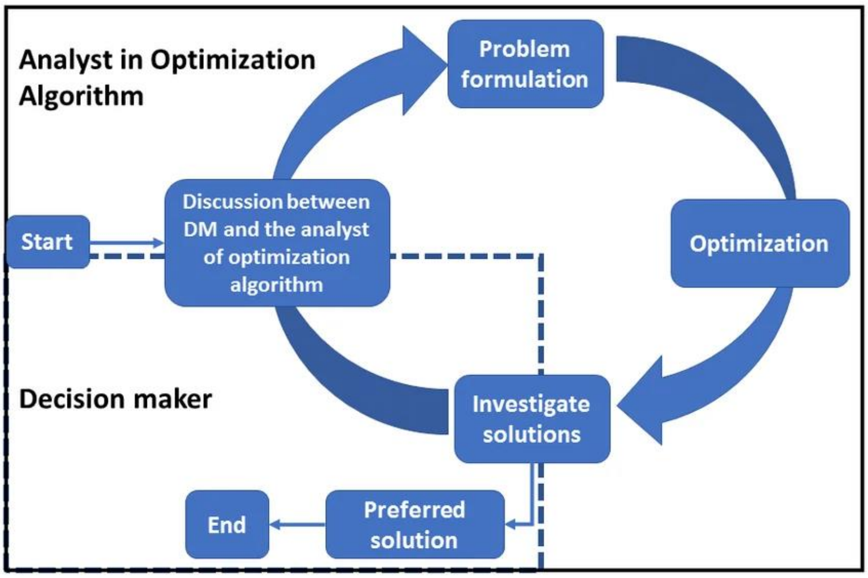


Figure 8: Framework of the optimization problem formulation and the decision-making process.

II.3 Presentation of Metaheuristic Methods

Metaheuristics are approximate optimization methods that use stochastic processes to solve well-defined optimization problems. These methods are particularly effective for complex and nonlinear problems where classical optimization approaches may be difficult to apply. They allow the identification of high-quality solutions within a reasonable computational time. Most metaheuristic algorithms are inspired by natural phenomena and evolutionary processes.
These algorithms enable an efficient exploration of the search space in order to obtain solutions close to the global optimum. Among the metaheuristic methods most commonly used in civil engineering are **Particle Swarm Optimization (PSO)** and **Genetic Algorithms (GA)**, which have demonstrated strong performance in structural optimization problems.

a. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a metaheuristic method developed by **James Kennedy** and **Russell Eberhart** in 1995. This method is inspired by the collective behavior observed in flocks of birds or schools of fish.
The principle of PSO is based on a set of particles representing potential solutions. Each particle moves within the search space by adjusting its position and velocity according to two main pieces of information:

- Its personal best position (**pbest**).
- The global best position found by the entire swarm (**gbest**).

At each iteration, the velocity and position of each particle are updated according to simple mathematical equations, allowing a gradual convergence toward the optimal solution.
The main advantages of PSO include:

- ✓ Simplicity of implementation.
- ✓ Fast convergence speed.

- ✓ A relatively small number of parameters to adjust.
- ✓ High efficiency for continuous and nonlinear optimization problems.

In civil engineering, PSO is widely used for structural optimization, weight minimization, stiffness improvement, displacement reduction, and the identification of mechanical parameters.

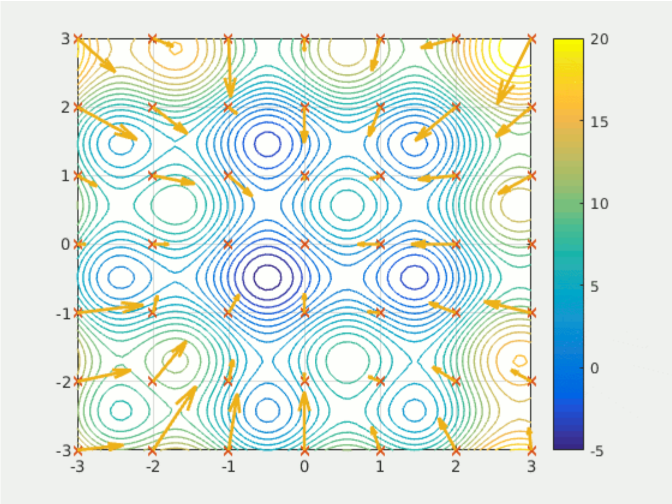


Figure 9: Conceptual illustration of Particle Swarm Optimization.

b. Genetic Algorithms (GA)

Genetic Algorithms (GA) are inspired by the mechanisms of natural evolution such as selection, crossover, and mutation. They were introduced by **John Holland** in the 1970 and are widely used in civil engineering for structural design optimization.

These algorithms are based on **Darwin's theory of evolution**, where the most suitable individuals are more likely to survive and reproduce.

In a GA, each potential solution is represented by a **chromosome**. An initial population of solutions is generated randomly and then progressively improved over several generations.

The general functioning of a GA relies on three main operators:

- ✓ **Selection:** choosing the best solutions to participate in reproduction.
- ✓ **Crossover:** combining chromosomes to generate new solutions.
- ✓ **Mutation:** introducing random modifications to maintain genetic diversity.

Genetic algorithms are particularly suitable for complex, discontinuous, and multimodal optimization problems. They provide strong global exploration capabilities and help avoid convergence toward local minima.

In civil engineering, GA are commonly applied to reinforcement optimization, optimal structural design, shape optimization, and the identification of parameters in numerical models.

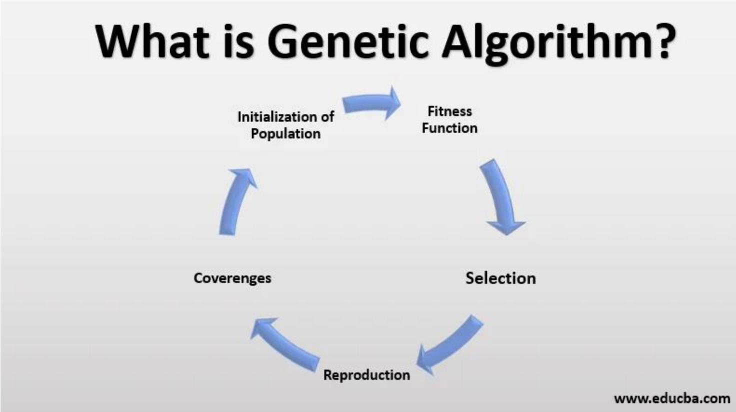


Figure 10: Basic process of the Genetic Algorithm.

II.4 Parameters and Stopping Criteria of the Algorithms

The performance of metaheuristic algorithms strongly depends on the selection of control parameters and stopping criteria. Proper parameter tuning ensures a balance between the accuracy of the obtained results and the computational cost.
Each algorithm has its own specific control parameters.

II.4.1 PSO Parameters and TVAC Strategy

To enhance the performance and convergence characteristics of the algorithm, a Time-Varying Acceleration Coefficients (TVAC) strategy is employed. The main parameters of the PSO algorithm are the following:

- **Number of particles (N):** Represents the size of the swarm. A larger number of particles improves the exploration of the search space but increases the computational cost.
- **Inertia weight (W):** This parameter controls the influence of the previous velocity on the current movement. In the implemented Python code, a linearly decreasing inertia weight is used (typically from 0.9 to 0.4). This ensures a broad global search at the beginning and a fine-tuned local search towards the end of the iterations.
- **Acceleration coefficients (C_1, C_2):** These parameters control the relative influence of the particle's best individual position and the global best position of the swarm.

In this study, these coefficients are dynamically adjusted to optimize the search process: C_1 is programmed to decrease linearly, while C_2 increases linearly as iterations progress. This strategy allows the particles to prioritize individual exploration in the early stages and gradually shift towards the global best position for precise exploitation.

C_1: cognitive factor
C_2: social factor

- **Maximum velocity (V_{max}):** This parameter limits the maximum velocity of the particles to avoid excessively large displacements within the search space. It must be properly defined to ensure stable convergence.

II.4.2 GA Parameters

The main parameters of the Genetic Algorithm are:

- **Population size:** Number of individuals in each generation. A larger population increases diversity but also increases computational time.
- **Crossover rate (Pc):** The probability of recombination between two parent solutions to produce new individuals.
- **Mutation rate (Pm):** The probability of random modification of genes in order to prevent premature convergence.
- **Number of generations:**
  This parameter determines the duration of the evolutionary process.

The appropriate selection of these parameters allows a balance between global exploration of the search space and convergence toward the optimal solution.

II.4.3 Stopping Criteria

Stopping criteria define the point at which the optimization process should terminate. Common stopping criteria include :

- **Maximum number of iterations:** The algorithm stops once a predefined number of iterations is reached.
- **Objective function stagnation:** If no improvement in the objective function is observed over a certain number of iterations.
- **Achieving a predefined precision:** When the variation in the solution falls below a specified threshold.
- **Maximum computation time:** When the allowed computational time is exceeded.

An algorithm is not intended to run indefinitely. In the context of the Python implementation for this study, two types of stopping criteria are typically used:

- **Maximum number of iterations:** A fixed limit is set (e.g: 50 iterations). If the optimum is not found before reaching this limit, the program stops to prevent excessive computation time.
- **Objective function stagnation (convergence):** If the maximum displacement (U_{max}) remains practically unchanged over 10 to 20 consecutive iterations, and the variation is below 10^{-6}, it indicates that the optimal opening position has been reached.

II.5 Hybridization Strategy: PSO and Nelder-Mead

To achieve maximum numerical precision, this research adopts a hybrid approach. While PSO is excellent at global exploration, its stochastic nature can be less precise in the final convergence steps. Therefore, the coordinates identified by the PSO algorithm are used as an initial guess for the Nelder-Mead simplex method. This local optimization technique "polishes" refines the results, ensuring that the identified optimal and critical positions are determined with high mathematical accuracy.

II.6 Advantages, Limitations, and Selection of Methods for the Present Study

Metaheuristic algorithms are powerful tools for solving complex optimization problems in civil engineering. Their effectiveness lies in their ability to explore a wide search space and handle nonlinear problems. However, these algorithms also have limitations that must be considered. This section presents a summary of the main advantages and limitations of the studied algorithms, as well as the rationale for the methods chosen in this study.

Table 2: Advantages and Limitations of PSO and GA Applied in Civil Engineering.

Advantages	Limitations
Ability to handle complex and nonlinear problems	Sometimes high computational cost
Good global exploration of the search space	Sensitivity to control parameters
Independence from gradient information	No guarantee of reaching the global optimum
Ease of implementation	Dependence on initial conditions
Flexibility and adaptability	Requires parameter tuning

- **Selection of Methods for the Present Study**

In the context of this study, which investigates the in-plane flexibility of reinforced concrete floor diaphragms with openings using AI-assisted numerical analysis, **Particle Swarm Optimization (PSO)** was primarily selected. The reasons for this choice include its conceptual simplicity, fast convergence, and robustness when applied to nonlinear problems. PSO is particularly well-suited for parametric optimization problems as well as complex numerical models such as finite element models.

Genetic Algorithms (GA) may also be considered as a reference metaheuristic approach for comparison purposes in future work, in order to further assess the reliability and stability of the proposed optimization strategy.

II.7 Conclusion

In this chapter, the theoretical framework for structural optimization was defined, focusing on the synergy between engineering constraints and metaheuristic algorithms. By selecting Particle Swarm Optimization (PSO) and enhancing it with dynamic parameters and a hybrid coupling with the Nelder-Mead method, we have developed a robust tool capable of navigating complex design spaces. This algorithmic foundation is essential for identifying both the safest and most vulnerable zones within a slab. With the optimization logic now established, the next chapter will focus on the practical implementation of these concepts using the Python programming language and Mindlin-Reissner plate theory.

CHAPTER III: PYTHON IN CIVIL ENGINEERING

Ш.1.Introduction

Ш.1.1. What is Python

Python is a high-level, interpreted, and versatile programing language with clear syntax and excellent readability. Developed and proposed by Guido van Rossum and first released in 1991, Python has become one of the most popular and widely use programing languages around the world due to its philosophy of code simplicity and efficiency.

- *"Python is often called a 'scripting language', but it is actually a robust, object-oriented programing language capable of managing complex software systems while remaining accessible to beginners. "- **Learning Python, Mark Lutz.***

Ш.1.2. Why use Python in Civil Engineering

Python has become an essential tool in civil engineering due to its simplicity, flexibility, and extensive ecosystem of scientific libraries. This programing language allows engineers and researches to develop numerical models, perform simulations, process data, and apply optimization techniques efficiently and quickly.
In the context of structural design and analysis, Python provides the ability to:

- Automate calculation and repetitive procedures, reducing the risk of human errors.
- Develop custom numerical models based of the need of each project.
- Integrate optimization algorithms such as Particle Swarm Optimization and Genetic Algorithms to determinate optimal structures or many other things.
- Interpret the result using 2D or 3D visualizations.

- *"For structural engineering students, Python is no longer just an optional skill, but fundamental tool for transforming theoretical concepts into interactive and precise computational and precise computational models. "- **Python manual for structural Engineering, by Gomez, Kennedy..., 2023.***

Since its inception, Python has evolved through several versions (notably Python 2.x and the current 3.x), consolidating its position in the global scientific community.

Python has gained widespread use in modern civil engineering due to its compatibility with other professional tools in the field, including **ABAQUS**, **OPENSEES**, **ETABS**, and **SAP2000**, as well as its open- source nature and its use in numerous research studies and applications in the field. It's also adopted for structural analysis, finite element modeling **(FEM)**, parametric design, and processing experimental or simulation data.
In this chapter, we will discuss the use of Python in civil engineering, focusing on the main libraries, their roles, and practical applications in structural analysis and optimization.

Ш.2. Python Structure and Syntax

Python is an interpreted, high-level its designed to be intuitive, characterized by simple and readable syntax. Its clear structure facilitates learning and writing complex programs, making it an ideal tool for civil engineers who developed numerical models and perform structural analyses.

Ш.2.1. Readability and indentation

One of the notable features of the Python language is the usage of indentation to define code blocks. This ensures that the code is clean and structured, which is crucial when debugging complex structural models.

- *"Indentation makes your code easier to read for you and for others. Its syntactic rules that ensure the visual structure of the program matches its logical structure. "**-Python crush Course, Eric Matthes.***

Ш.2.2. Variables and data types

Python supports several common data types, such as

- **Int:** integer numbers.
- **Float:** real numbers.
- **Str:** string.
- **Bool:** Boolean values (True/ False).
- **List, tuple, dict:** data structures to store multiple values.

  Example:
- A list can store slab dimensions or the coordinates of an opening:

```
slab_dimensions = [6.0, 3.0] # length and width in meters
opening_coordinates = [1.2, 0.8] # x, y position
```

Ш.2.3. Control Flow

Python allows conditional statements and loops, essential for repetitive calculations or data analysis.

- **if / else:** conditions.
- **For / while:** loops to iterate over data or repeat calculations.

  Example:

```
for opening in openings:
   if opening['type'] == 'circular':
      print("Calculation for circular opening")
```

Ш.2.4. Functions and Modularity

Function allows grouping instruction into reusable blocks, improving code clarity and modularity.

Example:

```
def calculate stiffness (length, width):
   return length * width * 30e9 # simplification: E * area
```

- This function can compute slab stiffness simply by calling it with different parameters.

Ш.2.5. Matrix and Array Handling

In civil engineering matrices are crucial for:

- Representing structural properties.
- Calculating displacements and stress (FEM).

Python use **NumPy** to efficiently manage multidimensional arrays and perform advanced mathematical operation.

Example:

```
import numpy as np

# simplified 2x2 stiffness matrix
K = np.array([[10, -5],
          [-5, 10]])
```

- This matrix can represent a simple system for structural analysis.

Ш.2.6. Comments and Best Practices

- Comments are written with # and are used to explain the code.
- Clarity and readability are essential, especially in research projects.

Example:

```
# Calculate stiffness for the main slab
stiffness = calculate_stiffness(6.0, 3.0)
```

Python's structure and simple syntax allow:

- Rapid understanding of code, even for complex models.
- Creation of modular and reusable functions.
- Efficient handling of matrices and numerical data via NumPy.
- Preparation for using specialized libraries for structural analysis and optimization.

Ш.3. Scientific Python Libraries

Python provides a large number of scientific libraries that are widely used in engineering and scientific research. These libraries offer powerful tools for numerical computation, data analysis, and visualization and results.

Ш.3.1. Numerical Computation Libraries

Numerical computation plays a fundamental role in engineering analysis and scientific modeling. Python provide several powerful libraries designed to perform efficient numerical calculations and matrix-based operations. These libraries enable engineers to implement numerical methods and perform complex computations required in structural analysis and simulation.

- **NumPy**
  It's one of the most popular libraries used for numerical computation. In civil engineering applications, this library is used for matrix manipulation, solving systems of equations, and implementing numerical techniques such as the finite element method.
  NumPy provides efficient data structures for handling multidimensional and matrices, which are essential in numerical modeling. Also includes a large collection of optimized mathematical functions that allow fast numerical computations.

(Source : Numerical Python)

- **SciPy**
  Extends the capabilities of NumPy by providing advanced tools for scientific computing. It includes modules for optimization, numerical integration, linear algebra, and differential equation solving. (Source: Python Programming and Numerical Methods).
- **scipy.sparse (CSR Matrix):** Specifically used to handle the Global Stiffness Matrix. Since structural matrices are sparse, using the Compressed Sparse Row (CSR) format significantly optimizes memory usage and speeds up the solving of **[K] {U} = {F}.**
- **scipy.optimize:** Used for the Nelder-Mead simplex algorithm to refine the optimal coordinates after the initial PSO search.

Example:

```
import numpy as np
import scipy.sparse.linalg as spla
from scipy.sparse import csr_matrix

def solve_structural_system(K_dense, F):
    # Convert the global stiffness matrix to a sparse format for efficiency
    K_sparse = csr_matrix(K_dense)
    # Solving for nodal displacements {U}
    U = spla.spsolve(K_sparse, F)
    return U
```

III.3.2. Data Analysis Libraries

In engineering applications, large amounts of numerical data are often generated during simulations and experimental studies. Efficient tools are therefore required to organize, process, and analyze these datasets. Python offer several libraries that facilitate data manipulation and analysis.

- **Pandas**
  Is a powerful library for data manipulation and analysis. It provides flexible data structures such as data frames, that allow structured data to be stored and manipulated efficiently. It also offers a wide range of functions for data filtering, statistical analysis, and data transformation.

In civil engineering, Pandas can be used to manage simulation results, analyze experimental datasets, and organize structural analysis outputs. The ability to easily import and export data from different file formats also makes Pandas particularly useful for integrating data from various sources.

(Source: Python for Data Analysis)

III.3.3. Visualization Libraries

Visualization plays an essential role in engineering analysis because it allows researches and engineers to interpret numerical results more effectively. Graphical representation of data helps in identifying patterns, trends, and relationships that may not be immediately visible in numerical form.

- **Matplotlib**

  This library provides a wide range of tools for creating high-quality scientific plots and graphical representations. Matplotlib allows users to generate line plots, bar charts, scatters plots, and other types of visualizations that are commonly used in scientific and engineering research.

  In structural engineering applications, it can be used to visualize structural displacements, stress distributions, or the evolution of optimization results. (Source: Python for Data Analysis.)

  - *"Matplotlib is a mathematical plotting library used to generate plots, charts, and other visualizations... it's particularly useful for understanding the data you've collected".*- ***Matthes, E, Python Crush Course.***

III.3.4. Optimization Libraries

Optimization techniques are used in engineering design problems to find the best solution from a variety of alternatives. Python has a number of libraries that make it easier to implement an optimization technique. One such libraries is:

- **PySwarms:**

  It's designed to implement Particle Swarm Optimization algorithms. Which is a metaheuristic technique inspired by the collective behavior of natural systems, such as bird flocks or fish schools.

  This technique can be used in engineering problems to find the optimal solution, improve structural performance, or minimize objective functions related to structural response. Libraries such as PySwarms provide robust tools for implementing these optimization methods in Python-based numerical models.

"The implemented algorithm in Python is designed to perform a dual-search, identifying both the optimal location to maximize stiffness and the critical zones to ensure structural safety."

III.4. Python Application in Civil Engineering

Python is becoming an important application for civil engineers due to its compatibility with modeling, analysis, and optimization of structural systems. Python is compatible with the analysis of structural systems due to its wide range of libraries for solving complex problems.

(Python for Civil and Structural Engineers; Computational Methods for Civil Engineering Using Python).

Ш.4.1. Structural Analysis

In the analysis of structures, Python applies numerical analysis for the computation if internal forces, stresses, and displacements. Engineers can apply Python to create computer models for the analysis of various structural systems. Engineers can also apply Python to analyze the behavior of various elements in the structure, and how different elements respond to applied loads.
Libraries such as NumPy and SciPy provide tools for matrix computations and solving systems of linear equations, which are essential for structural calculations. These tools also facilitate parametric studies, allowing the evaluation of how different design parameters affect structural behavior (Python for civil and structural engineers).

Ш.4.2. Finite Element Models (FEM)

The finite element method (FEM) is one of the most widely used approaches for structural analysis. Python serves as an ideal environment for discretizing continuous structures (like plates and shears walls) into finite elements, assembling global stiffness matrices, and solving for nodal displacements. Also, to control specialized software such as ABAQUS through scripting. This allows for automated generation of models, application loads, and extraction of results.

- *"The integration of Python in numerical methods allows for the simulation of complex physical phenomena through discretization, providing researchers with a flexible platform to implement custom constitutive laws and boundary conditions."-* ***Mansour & El Boukili, Méthodes Numériques & Simulations avec Python.***

Using Python in FEM reduces the time required for model setup and enables the investigation of multiple scenarios, like the effect of openings in slabs, beams, or shear walls. (Computational Methods for Civil Engineering Using Python).

Ш.4.3. Optimization (PSO/GA)

Modern engineering challenges require finding the most efficient designs under strict constraints. Python scientific libraries make it a leading choice for implementing metaheuristic optimization algorithms, such as Particle Swarm Optimization (PSO) and Genetic Algorithms (GA).
These algorithms are used to identify optimal design parameters, including the placement and geometry of openings in slabs, aiming to reduce structural flexibility or displacements while satisfying constraints. Libraries such as PySwarms and DEAP facilitate the integration of these optimization algorithms within computational models. (Python for Civil and Structural Engineers.)

- *"Optimization algorithms implemented in Python provide a robust framework for iterative design, allowing engineers to converge on optimal solutions that* *satisfy both safety standards and economic efficiency."*- ***Python for Civil and Structural Engineers.***

Ш.5 Conclusion

This chapter has demonstrated that Python is a powerful programming language for scientific computing and engineering applications; it is a powerful computational engine for modern civil engineering. By leveraging libraries such as NumPy for matrix operations, Matplotlib for visualization, and PySwarms for optimization, a numerical framework was implemented based on Mindlin-Reissner theory. This environment allows for the automated analysis of slab diaphragms and the identification of optimal opening positions. With the programming infrastructure now fully established, the following chapter will focus on the practical application of this tool, presenting the numerical results and validating them against Abaqus finite element results.

CHAPTER IV: NUMERICAL MODELING AND FINITE ELEMENT ANALYSIS

IV.1 Introduction

This chapter presents the numerical modeling and finite element analysis (FEA) of the investigated structures. The primary objective is to evaluate the structural behavior of reinforced concrete floor diaphragms with openings by employing a dual-verification approach: a customized Python-based FEA tool and the Abaqus finite element software.

A finite element model is first developed in Python to analyze the structural response and capture the essential behavior of the diaphragm under various loading conditions. This model is built upon rigorous mechanical assumptions and numerical formulations tailored for slab analysis.

Subsequently, a detailed finite element analysis is conducted using Abaqus to obtain reference finite element results and validate the Python-based model. The comparison between the Python results and Abaqus benchmarks serves to verify the reliability, accuracy, and robustness of the proposed computational approach.

IV.2 Structural Models and Geometry

To investigate the influence of support conditions on the diaphragm behavior, two primary structural configurations were modeled. These models represent the extremes of lateral and vertical stiffness distribution typically found in reinforced concrete structures.

IV.2.1 Case A: Slab Supported by Shear Walls

In this configuration, the slab is continuous and supported by longitudinal shear walls at the edges. This model evaluates the "diaphragm action" and force distribution when stiffness is provided by a continuous line support.

- **Support condition:** The walls are modeled as rigid supports to capture the boundary effects on the slab.
- **Visual representation:** As shown in **Figure 11**, the 3D view illustrates the integration of the horizontal diaphragm with the vertical wall system.

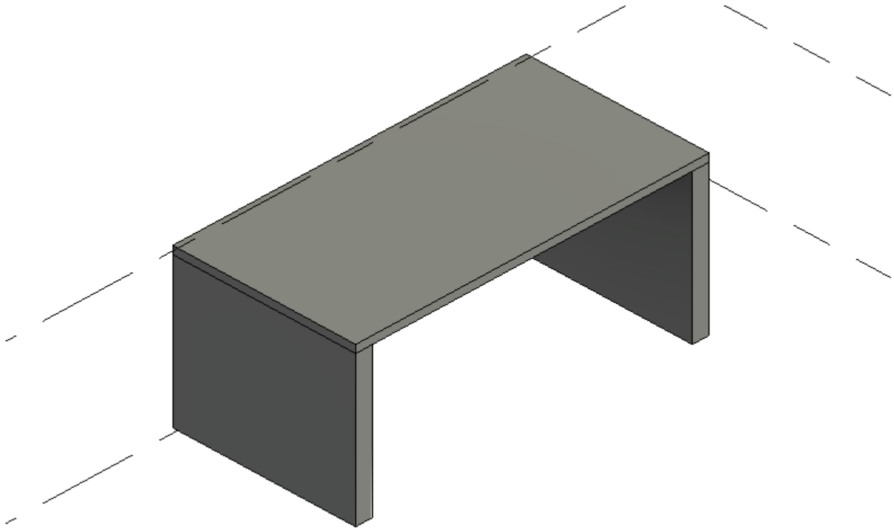


Figure 11:3D isometric view of the slab-shear wall system.

IV.2.2 Case B: Slab Supported by Columns

This model represents a more flexible system where the slab is supported by discrete point supports (columns). This case is critical for analyzing local stress concentrations and the impact of opening placement near discrete supports.

- **Support Condition:** The columns are modeled as square discrete supports.
- **Visual Representation: Figure 12** provides the 3D view of the slab with point supports located at the corners, reflecting the geometry used in the Python solver.

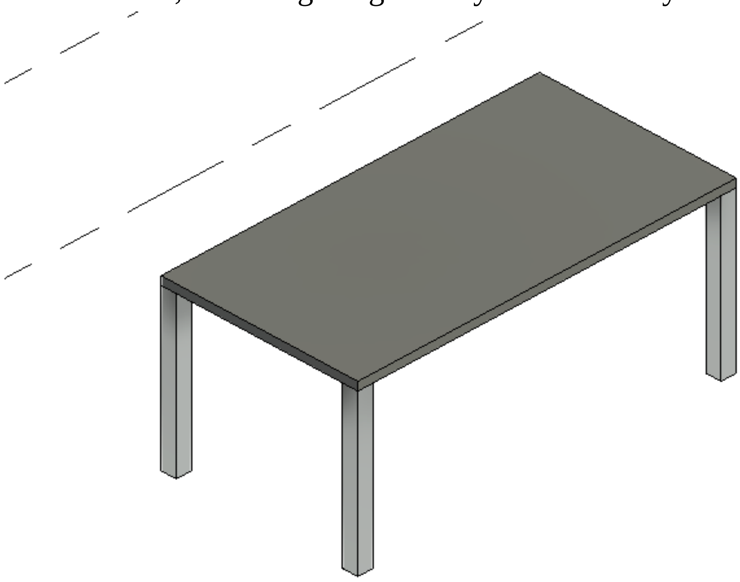


Figure 12:3D isometric view of the slab supported by columns.

IV.2.3 Geometrical Dimensions and Properties

To ensure a consistent comparison between the customized Python tool and the Abaqus finite element benchmark, the following geometrical parameters were strictly defined:

- **Slab Dimensions:** The investigation is conducted on a rectangular reinforced concrete slab with a clear span of **3.0 m × 6.0 m** and a constant thickness of **15 cm**. **(Figure 13)**

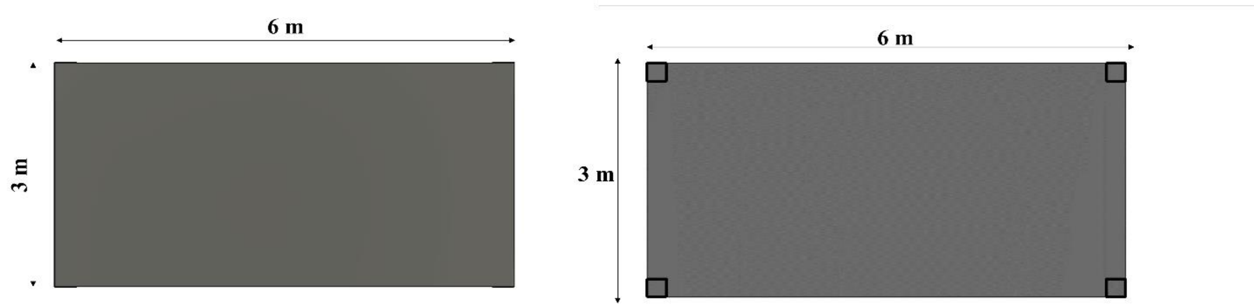


Figure 13: Top view of a 3.0 m × 6.0 m slab for both cases.

- **Vertical Supports Height**: For both shear wall and column scenarios, the supporting elements have a constant height of **3.0 m**, representing a typical story height in residential buildings. This height is verified in the elevation views **(Figure 14)** showing the floor levels.

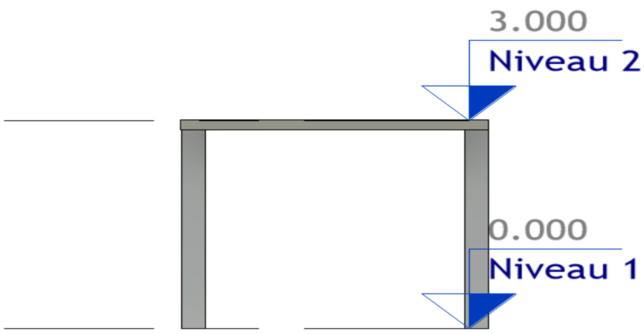


Figure 14: Side elevation (XZ-Plane) showing level heights.

- **Shear Wall Dimensions:** The supporting walls are modeled with a thickness of **30 cm**, ensuring high lateral stiffness along the supported edges.
- **Column Dimensions:** In the point-supported model, the columns feature a square cross-section of **25 cm × 25 cm,** providing precise contact areas for the finite element mesh.

IV.3 Modeling Assumptions

The finite element model developed in this study is founded on several mechanical and numerical assumptions designed to simplify the structural behavior while maintaining high computational accuracy.

- **Plate Theory:** The slab is modeled using the **Mindlin–Reissner plate theory**, which accounts for transverse shear deformation. This assumption is particularly suitable for moderately thick plates and provides a more realistic representation of the structural response, especially in the presence of openings where shear effects are significant.
- **Material Behavior:** The material is assumed to be linear elastic, homogeneous, and isotropic. Consequently, the material properties are independent of direction and position, and the stress–strain relationship remains linear throughout the considered loading range.
- **Geometric and Kinematic Assumptions:** Small displacement theory is adopted, meaning that geometric nonlinearities are neglected. A linear static analysis is performed, which is valid for service load conditions where deformations remain relatively small compared to the slab thickness.
- **Loading Conditions:** The loads are assumed to be static and gradually applied (monotincally increasing), neglecting dynamic effects such as impact, vibration, or time-dependent behaviors like creep and shrinkage.
- **Connectivity:** All structural components are assumed to be perfectly connected, ensuring full continuity of displacements and efficient force transfer between the finite elements.

These assumptions ensure an optimal balance between computational efficiency and the reliability of the model. They are integrated into the Python-based finite element code to simulate the structural response of the system accurately.

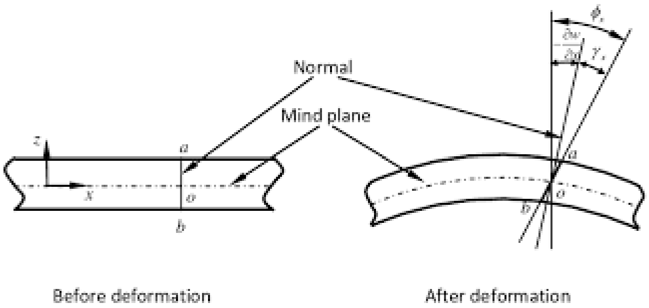


Figure 15: Geometry and kinematics of the Mindlin-Reissner plate theory showing the transverse shear deformation.

IV.4 Discretization of the Structure

The discretization of the structure is a fundamental phase in the finite element formulation, as it transforms the continuous physical domain into a finite number of discrete elements suitable for numerical analysis.

In this study, the structural system is discretized into small, interconnected finite elements to approximate its mechanical behavior with high precision. The mesh is generated using a regular subdivision of the slab and structural components. This ensures an accurate representation of both global structural performance and local effects, particularly in the vicinity of slab openings where stress gradients are high.

Each element is linked to its neighbors through shared nodes, which facilitate the transfer of displacements and internal forces across the entire system. The nodal degrees of freedom (DOFs) are defined based on the selected plate element type, consistent with the Mindlin-Reissner theory previously discussed.

The mesh density is carefully selected to balance computational efficiency and solution accuracy. A finer mesh is strategically adopted in regions of expected stress concentrations specifically around openings and connection zones to capture the complex structural response accurately. This discretized framework serves as the foundation for assembling the global stiffness matrix and solving the governing equilibrium equations via the Python-based finite element solver.

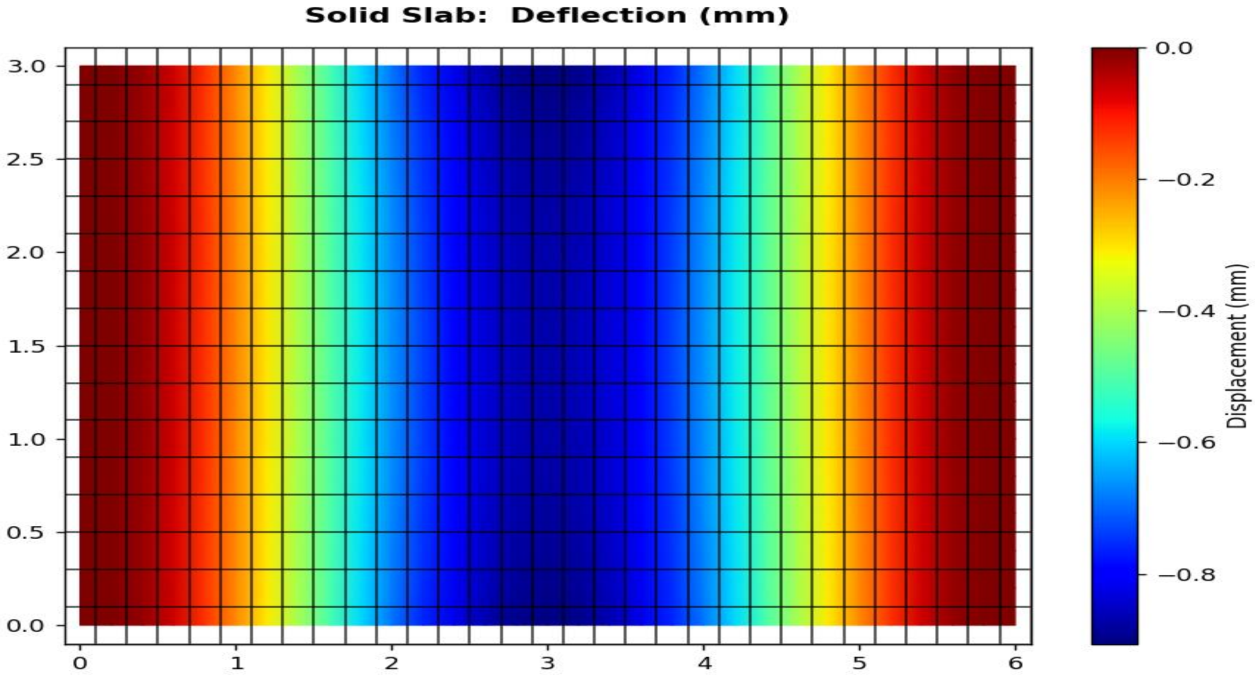


Figure 16: Mesh without openings.

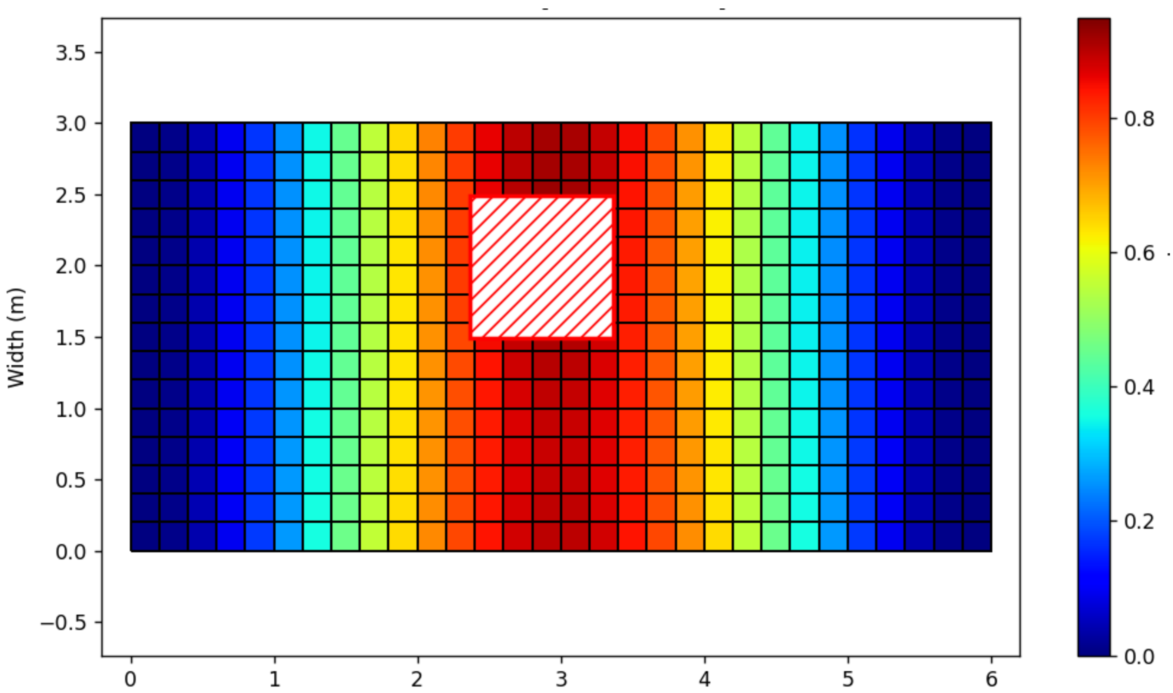


Figure 17: Mesh with openings.

IV.5 Formulation and Assembly of the Stiffness Matrix

In the finite element method, the structural behavior is governed by the relationship between nodal displacements and external forces through the stiffness matrix formulation.

For each finite element, a local stiffness matrix $[k_e]$ is derived based on the Mindlin–Reissner displacement field and the material's mechanical properties. This matrix relates the nodal forces $\{f_e\}$ to the nodal displacements $\{u_e\}$ at the element level.

After defining the element stiffness matrices, a global stiffness matrix [K] is constructed by assembling all individual element contributions. This assembly process ensures kinematic continuity between adjacent elements through shared nodes and guarantees the equilibrium of the entire structure. The resulting global system of equations is expressed as:

$$[K].\{U\} = \{F\}$$

Where:

[K] is the global stiffness matrix.

{U} is the displacement vector.

{F} is the external force vector.

The solution of this system provides the nodal displacements, which are used to compute strains and stresses. In this study, the assembly and solution procedures are numerically implemented using Sparse Matrix techniques in Python to ensure computational efficiency. The complete implementation of the assembly algorithm is provided in Appendix [A].

IV.6 Numerical Implementation in Python

The finite element formulation developed in this study was implemented from scratch using the Python programming language. Python serves as a powerful computational tool to translate the theoretical finite element procedures into a robust numerical algorithm.
The implementation process begins with the discretization of the structural domain. Nodes and elements are programmatically defined, and their connectivity is established to represent the structure's geometry in a numerical format. The core of the algorithm involves formulating the element stiffness matrices based on the adopted mechanical assumptions and material properties. These local matrices are subsequently assembled into a Global Stiffness Matrix using optimized sparse matrix techniques (SciPy's CSR format) to handle the large system of equations efficiently.
A key feature of the developed code is its ability to simulate various lateral loading scenarios. Three distinct loading patterns were numerically implemented to evaluate the slab's response under different stress distributions:

1. **Uniformly Distributed Load:** To represent standard equivalent static loads.
2. **Triangular Load:** To simulate linear seismic force distribution across the slab.
3. **Double-Triangular Load**: To capture complex in-plane force variations and torsional effects across the diaphragm.

Boundary conditions are then applied to reflect the physical constraints, such as fixed or simply supported edges. Following the application of these external loads, the resulting system of linear equations [K]. {U} = {F}is solved to obtain the nodal displacement field.
Finally, a post-processing phase is executed to compute strains and stresses. While this section outlines the fundamental algorithmic logic, the complete Python source code, including the implementation of the loading functions,is provided in Appendix [A].

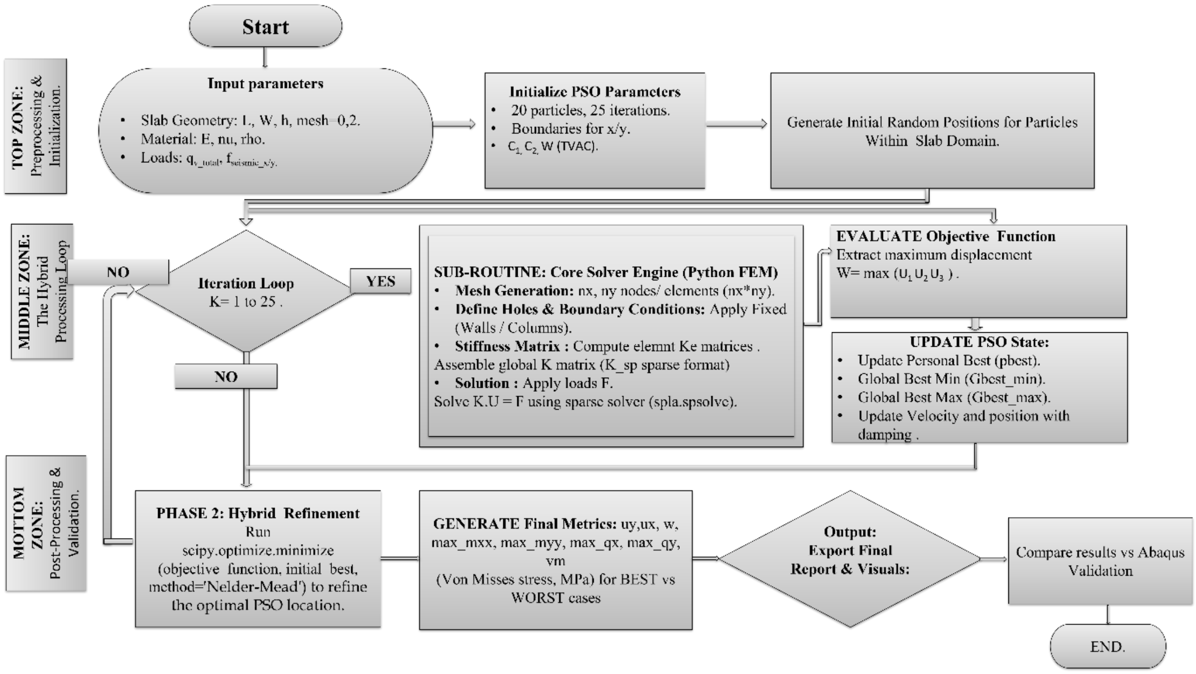


Figure 18: Detailed flowchart of the Python-based numerical solver, illustrating the integration of Finite Element Method (FEM) with Particle Swarm Optimization (PSO).

IV.7 Particle Swarm Optimization (PSO) and Hybrid Refinement

To automate the search for the optimal opening configuration, the developed Python code integrates **the Particle Swarm Optimization (PSO**) algorithm with the Finite Element solver. The optimization process is iterative, where a population of candidate solutions (particles) explores the slab domain. Each particle represents a potential set of opening coordinates.
To enhance the global search capability and avoid premature convergence, a **Time-Varying Acceleration Coefficients (TVAC)** strategy is implemented. This allows the algorithm to explore the entire domain effectively in the early stages and focus on the most promising regions as the iterations progress. The "fitness" of each particle is evaluated based on the maximum displacement (w_{max}) obtained from the FEM solver.
Furthermore, a **hybrid refinement step** using the **Nelder–Mead** simplex method is applied once the PSO identifies the near-optimal region. This local search algorithm performs a fine-tuning of the coordinates to improve the precision of the optimal solution. The integration of these techniques ensures a robust and efficient framework for identifying the most favorable structural locations for slab openings.

IV.8 Finite Element Modeling in Abaqus

A reference finite element model is developed in **Abaqus/CAE** to validate the results of the Python-based solver. In this model, the slab is explicitly represented using shell elements, while the supporting system is idealized through kinematic boundary conditions. As shown in **Figure 19**, Column locations are modeled as fixed-point constraints, whereas shear walls are represented by line supports to reflect their continuous stiffness.
The material is defined as linear elastic and homogeneous. A structured mesh is adopted for the slab domain, with local mesh refinement applied in the vicinity of openings to capture high stress gradients accurately. The loading protocol includes both gravitational and seismic actions, with the latter applied in two orthogonal horizontal directions to simulate the in-plane flexibility of the diaphragm. This model serves as the benchmark for evaluating the structural performance and confirming the accuracy of the optimization results.

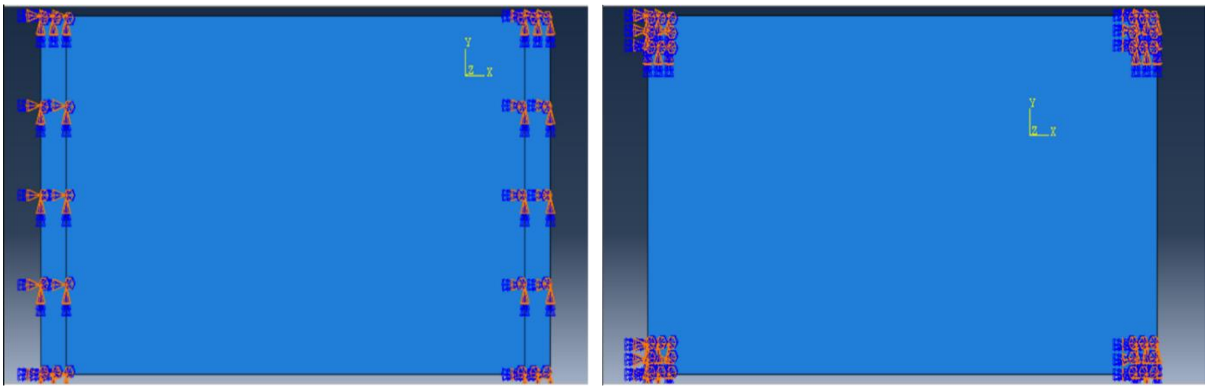


Figure 19: Boundary conditions and support configurations in the Abaqus reference model.

IV.9 Validation of the Python Model

The validation of the developed Python solver is performed by comparing its numerical results with the Abaqus reference model. The comparison is carried out in terms of nodal displacements and stress distributions for identical structural configurations and loading conditions (Uniform, Triangular, and Double-Triangular).

The results demonstrate a strong correlation and good agreement between the two platforms, confirming the reliability of the formulated stiffness matrices and the load distribution algorithms. Minor discrepancies are within acceptable engineering tolerances, primarily arising from differences in numerical integration schemes and mesh refinement algorithms. This successful validation demonstrates that the implemented Python model is a robust tool, capable of accurately predicting structural behavior for further parametric studies.

IV.10 Conclusion

This chapter presented the numerical modeling and finite element implementation of the studied structures using both a Python-based code and the Abaqus software. A finite element model was first developed in Python from scratch based on the Mindlin–Reissner plate theory and integrated with a PSO-TVAC optimization algorithm.

The model was subsequently validated by comparison with Abaqus results, showing high accuracy in predicting displacement and stress fields. This coupled numerical framework, enhanced by a hybrid refinement step, forms a reliable foundation for the parametric and optimization studies presented in the next chapter, where the influence of opening locations and shapes will be analyzed in detail.

"The complete Python source code used for FEM analysis, lateral loading, and PSO optimization is provided in Appendix [A]."

CHAPTER V: RESULTS AND DISCUSSION

V.1 Introduction

This chapter investigates the numerical response of reinforced concrete (RC) slab systems under simulated lateral loading conditions. The primary objective is to evaluate and contrast the structural performance of slabs integrated into two distinct support systems: shear walls and discrete columns.

The analysis emphasizes the sensitivity of the slab's global response specifically displacement, stress distribution, and shear behavior to the geometry and spatial positioning of openings. Utilizing a Finite Element (FE) framework developed in Python and executed via Abaqus, the study facilitates a seamless transition of optimal coordinates from the optimization algorithm to the numerical environment for rigorous evaluation.

A Particle Swarm Optimization (PSO) algorithm is employed to identify both optimal and critical opening configurations within the slab domain. Furthermore, a comparative study is conducted to highlight the significant impact of boundary conditions (linear vs. point supports) on the structural integrity. Finally, the PSO-derived results are validated against Abaqus FE simulations to ensure the reliability and precision of the proposed optimization procedure.

"The overall computational framework and the interaction between the optimization loop and the structural solver are illustrated in the systematic workflow shown in **(Figure 18)** of the previous chapter. This study follows this hybrid processing approach to ensure both global exploration of the slab domain and local refinement of the optimal opening coordinates."

V.2 Effect of Structural Support System (Shear Walls vs Columns)

This section investigates the influence of the structural support configuration on the global response of RC slabs under lateral loading. Two distinct boundary conditions are evaluated: linear support provided by shear walls and point support provided by discrete columns.

Numerical results indicate that the support mechanism dictates the slab's performance. Slabs supported by shear walls demonstrate significantly higher inherent rigidity, resulting in minimal peak displacements. Conversely, column-supported systems exhibit increased flexibility; this is primarily attributed to the reduction in lateral bracing compared to continuous wall supports.

Furthermore, boundary conditions significantly alter the internal load paths. The column-supported system is characterized by localized stress concentrations near the point supports, leading to a non-uniform distribution. In contrast, wall-supported slabs facilitate a more distributed load transfer mechanism, enhancing the overall structural stability.

V.2.1 Quantitative Comparison of Support Systems

To evaluate the influence of the structural support system, a comparative analysis was conducted using a **rectangular opening (2.0×0.5 m)** as a reference case. This specific geometry was chosen to highlight how different boundary conditions respond to elongated discontinuities within the slab domain.

The numerical findings, summarized in **Table 3**, validate the technical superiority of the wall-supported system over the column-supported configuration. By providing a continuous linear boundary condition, the shear walls effectively minimize the slab's effective span, limiting the peak displacement to **0.858 mm**. In contrast, the point-support mechanism of the discrete columns results in a more flexible structural response, with a maximum recorded displacement of **1.612** mm for the same opening geometry.

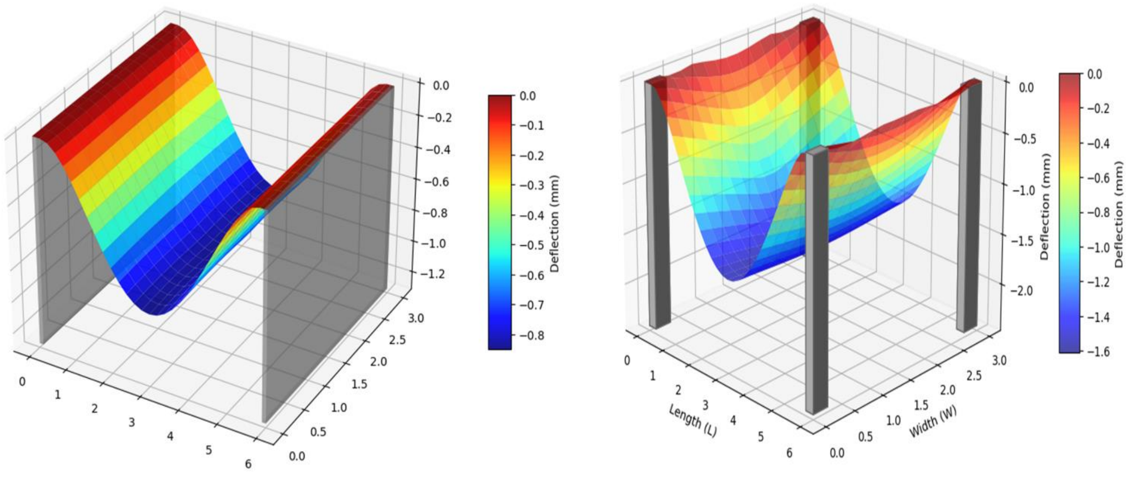


Figure 20: Comparative 3D deformation contours of wall-supported and column-supported slabs with a rectangular opening (2.0×0.5m).

Table 3: Comparison of maximum displacement for wall-supported and column-supported slabs (Rectangular Opening Case).

Structural System	**Maximum Displacement (mm)**
Wall-supported slab	0.85842
Column-supported slab	1.61240

- **Analysis of Structural Efficiency:**

Based on the results presented in **Table 3**, the transition from a column-supported system to a wall-supported system resulted in a **46.76%** reduction in maximum displacement. This significant improvement is primarily attributed to the increased lateral stiffness and superior load distribution path inherent in shear wall structures.

As illustrated in the 3D deformation contours in **Figure 20**, the wall-supported slab maintains a more uniform profile due to the restricted rotation and translation at the edges. Conversely, the column system exhibits localized stress concentrations and higher deflection gradients, which underscores the sensitivity of flat slabs to opening placements when supported by point elements.

V.3 Effect of Opening Shape

This section presents a parametric investigation into the influence of opening shapes on the structural response of the RC slab. Three distinct geometries are evaluated: square **(1×1 m)**, rectangular **(2.0 × 0.5 m)**, and circular **(r=0.5 m)** openings.

Numerical results indicate that opening geometry is a decisive factor in stress redistribution. Circular openings, characterized by their smooth boundary profiles, facilitate a more uniform stress flow and effectively mitigate the stress concentration effects typically observed at sharp corners. In contrast, rectangular openings, due to their high aspect ratio and longitudinal discontinuities, tend to induce more pronounced stress gradients along their major edges.
Square openings demonstrate an intermediate structural behavior. Collectively, these findings confirm that the geometric configuration of the opening significantly dictates both the displacement field and the localized stress response of the slab system.
To visualize this impact, structural sensitivity heatmaps were generated for both wall-supported and column-supported systems.

a- Shear Wall-Supported Slabs

In this configuration, the continuity of the supports helps in distributing the loads more effectively. As shown in **Figure 21**, the safe zones (represented by lighter colors) are more elongated, following the path of the shear walls.

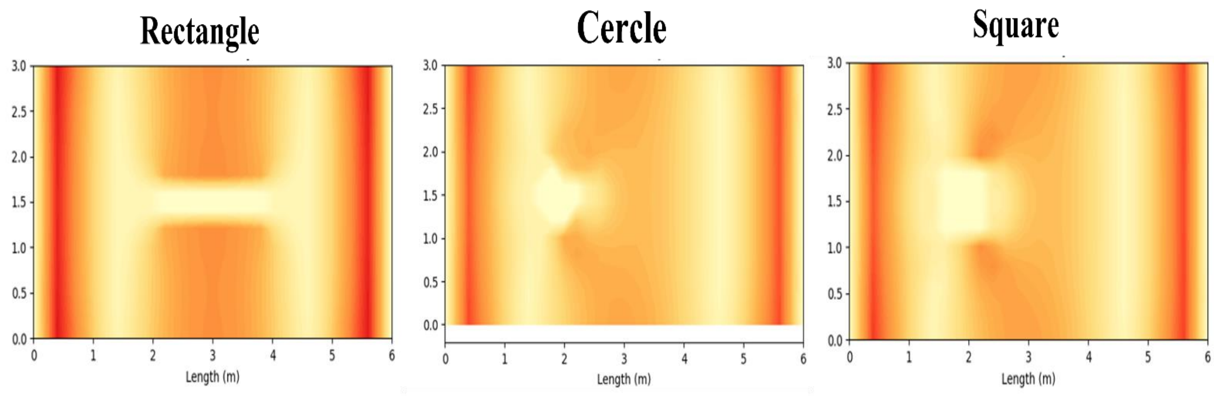


Figure 21: Structural sensitivity maps for rectangular, circular, and square openings in Shear wall-supported slabs.

b- Column-Supported Slabs

In contrast, the column-supported system exhibits a more restricted behavior. The sensitivity maps in **Figure 22** show that maximum displacements (intense red zones) are concentrated at the four corners where the columns are located.

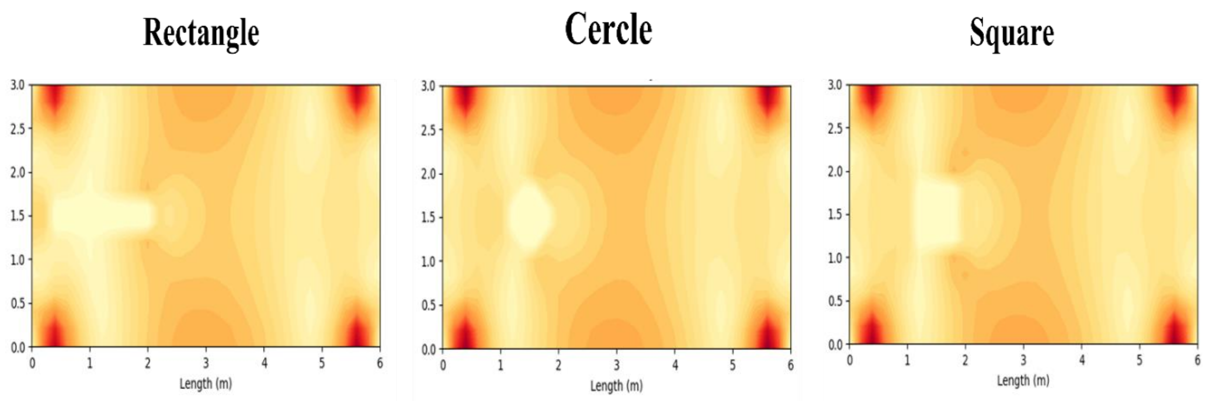


Figure 22: Structural sensitivity maps for rectangular, circular, and square openings in Column -supported slabs.

Table 4: Performance Summary of Opening Geometries under Uniform Loading.

Opening Shape	Area (m^2)	Support System	Max Displacement (mm)	Min Displacement (mm)
Circular	0.785	Shear Walls	1.36951	0.87969
		Columns	1.80449	1.57898
Rectangular	1.0	Shear Walls	0.98240	0.85842
		Columns	1.88946	1.61240
Square	1.0	Shear Walls	1.06459	0.88323
		Columns	1.89644	1.57953

V.3.1 Analysis and Discussion of Geometric Influence

The numerical results presented in **Table 4**, complemented by the structural sensitivity maps, show that the influence of opening geometry depends strongly on the support system and on the opening area. Since the circular opening has a smaller area **(0.785 m^2)** compared with the rectangular and square openings **(1.0 m^2)**, the comparison should be interpreted by considering both geometric shape and opening size.

1. **Superiority of Circular Geometry**

   The circular opening presents a favorable geometric configuration due to its smooth and continuous boundary. Unlike square and rectangular openings, it does not contain sharp corners, which helps reduce localized stress concentration around the opening. In the column-supported system, the circular opening produced the lowest displacement range, with a maximum displacement of **1.80449 mm** and a minimum displacement of **1.57898 mm.** This indicates that circular openings may contribute to a more regular redistribution of internal forces, particularly when the slab is supported by columns.

However, in the shear-wall-supported system, the circular opening did not produce the lowest maximum displacement. This confirms that the structural response is not governed by the opening shape alone, but also by the interaction between the opening geometry, its area, and the global stiffness provided by the support system.

2. **Intermediate Behavior of Square Openings**

The Square opening (1.0×1.0 m) demonstrates an intermediate structural response. While it shares the same area as the rectangular opening, it maintains a better balance of stiffness due to its equal dimensions in both directions. However, the presence of 90º corners induces localized stress gradients, resulting in higher displacement compared to the circular shape. In the column-supported system, the square opening reached a maximum displacement of **1.896 mm**, highlighting its vulnerability when placed near point supports.

3. **Critical Impact of Rectangular Openings**

The Rectangular opening (2.0 × 0.5 m) proved to be the most critical configuration, particularly in the column-supported system where it recorded a maximum displacement of **1.88946 mm** (and a minimum of **1.61240 mm**). The high aspect ratio of the rectangle creates a significant longitudinal discontinuity in the slab's mid-span. This forces the internal forces to deviate over a longer distance, leading to a substantial reduction in the slab’s overall lateral stiffness.

V.3.2 Conclusion on the Combined Effect of Opening Geometry and Support System

In conclusion, the results confirm that the support system plays a major role in controlling the global magnitude of displacement, while the opening geometry strongly affects the local redistribution of stresses and stiffness around the discontinuity.
The circular opening generally offers a more favorable stress redistribution due to the absence of sharp corners, especially in the column-supported system. However, because its area is smaller than that of the square and rectangular openings, its performance should not be attributed to geometry alone. The rectangular and square openings, both having sharp corners, may lead to higher localized stress concentrations and should therefore be carefully positioned within the identified safe zones of the slab to limit excessive deformation.

V.4 Effect of Lateral Load Distribution Across All Parametric Configurations

V.4.1 Introduction and Theoretical Framework

This section investigates the influence of lateral load distribution patterns on the structural behavior of reinforced concrete slabs containing openings. To ensure a rigorous parametric evaluation, three distinct lateral loading distributions are analyzed: Uniformly Distributed Loading (UDL), Linear Triangular Loading, and Double-Triangular Loading. Each loading configuration is systematically applied under identical geometric and boundary conditions, allowing for a consistent cross-comparison between the obtained results.

The numerical simulations indicate that while the global displacement field is heavily anchored by the surrounding boundary stiffness, the internal stress gradients and shear flow dynamics vary significantly depending on the applied lateral force distribution profile. Uniform loading produces a relatively regular deformation pattern. In contrast, non-uniform profiles (triangular and butterfly) generate higher localized deformation and more aggressive internal shear gradients that tax the structural integrity of the floor diaphragm.

Opening shape	Lateral load pattern	Configuration (PSO)	Max displacement (mm)	Max shear force (kN)	Max stress (MPa)
Circular	Uniform	Best/Worst	0.8797/1.3695	14.01/35.60	1.52/2.41
	Triangular		0.8797/1.3312	14.48/34.21	1.53/1.78
	Double triangular		0.8797/1.3623	14.48/26.87	1.54/1.64
Square	Uniform		0.8832/1.1421	19.08/20.29	1.51/2.34
	Triangular		0.8832/1.0646	19.08/20.29	1.53/2.35
	Double triangular		0.8832/1.0646	19.08/20.29	1.53/2.33
Rectangular	Uniform		0.8584/0.9824	3.45/15.41	1.47/1.97
	Triangular		0.8584/0.9824	3.45/15.41	1.49/1.98
	Double triangular		0.8584/0.9824	3.45/15.46	1.49/2.00

Table 5: Multi-Parametric Structural Responses for Continuous Shear Wall Systems.

Opening shape	Lateral load pattern	Configuration (PSO)	Max displacement (mm)	Max shear force (kN)	Max stress (MPa)
Circular	Uniform	Best/Worst	1.579/1.9478	83.22/84.04	3.72/3.81
	Triangular		0,8797/1.816	83.22/84.75	3.76/3.88
	Double triangular		1.579/1.8227	83.22/85.02	3.77/3.83
Square	Uniform		1.7953/1.8964	82.94/85.18	3.70/3.87
	Triangular		1.5795/1.8964	82.94/85.18	3.76/3.92
	Double triangular		1.5795/1.8964	82.94/85.18	3.76/3.91
Rectangular	Uniform		1.6124/1.8895	83.77/93.13	3.74/4.08
	Triangular		1.6124/1.8784	83.77/92.83	3.81/4.1
	Double triangular		1.616/1.8895	83.33/93.13	3.73/4.1

Table 6: Multi-Parametric Structural Responses for Discrete Column Systems.

V.4.2 Step-by-Step Parametric Evaluation (Case Study: Rectangular Opening)

Finally, the structural investigation shifts to the Rectangular Opening configuration. Due to its high aspect ratio and sharp 90 internal corner geometry, this configuration represents the most challenging and highly sensitive geometric discontinuity analyzed in this research project.

a) Uniform Lateral Loading (UDL) Response Profiles

a. Global Best (Optimal Case) Layout

Under a basic uniform lateral force distribution, the rectangular discontinuity demands precise spatial control. Supported by continuous shear walls, the global slab behavior remains stable. However, when transitioning to the discrete column system, the loss of continuous boundary bracing causes a significant widening of the out-of-plane deflection profile, with localized stress concentrations beginning to emerge along the longer edges of the rectangular void.

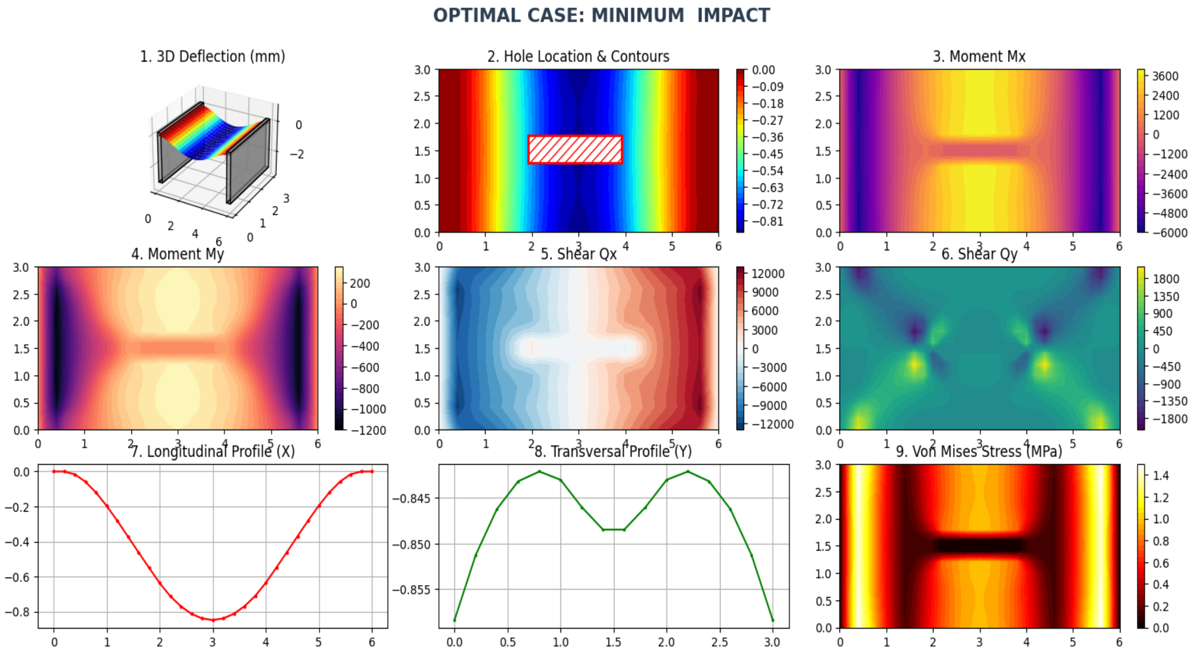


Figure 23: Optimal rectangular opening response profile under uniform loading shear walls system.

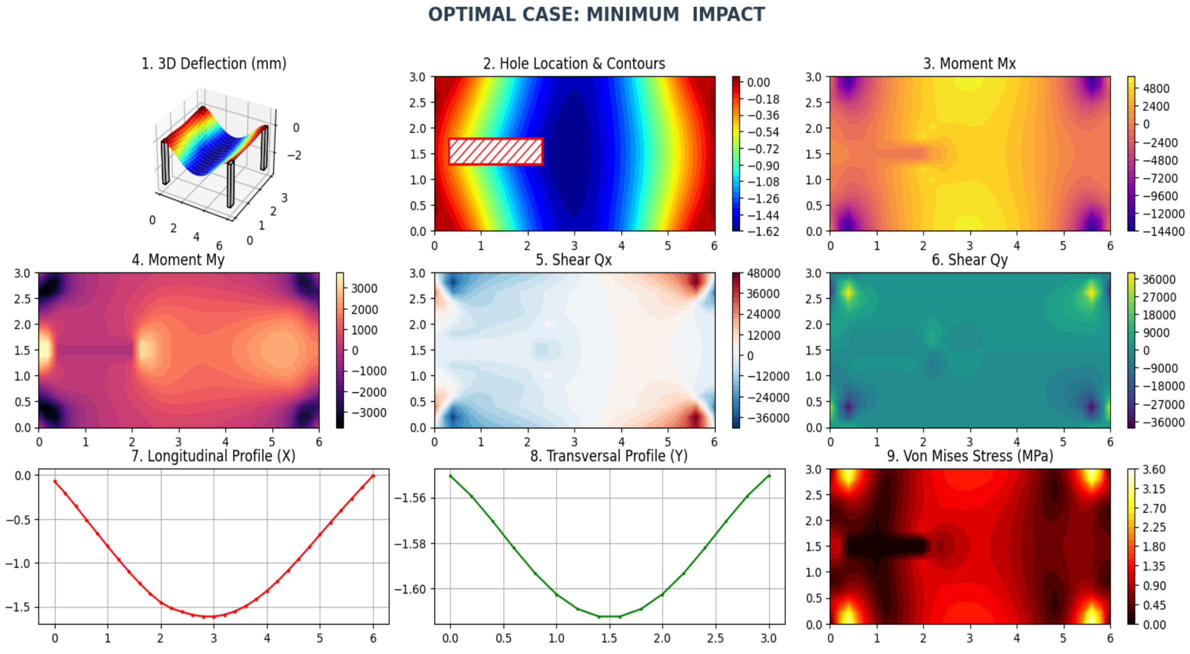


Figure 24: Optimal rectangular opening response profile under uniform loading columns system.

b. Global Worst (Critical Case) Layout

When the rectangular opening is positioned in its worst spatial zone under uniform loading, the structural vulnerability escalates sharply. In the discrete column environment, the global deflection curvature deepens significantly. The Von Mises stress profile highlights severe localized stress fields tracking aggressively at the internal corners, pointing to a high probability of concrete macro-cracking along the opening boundaries.

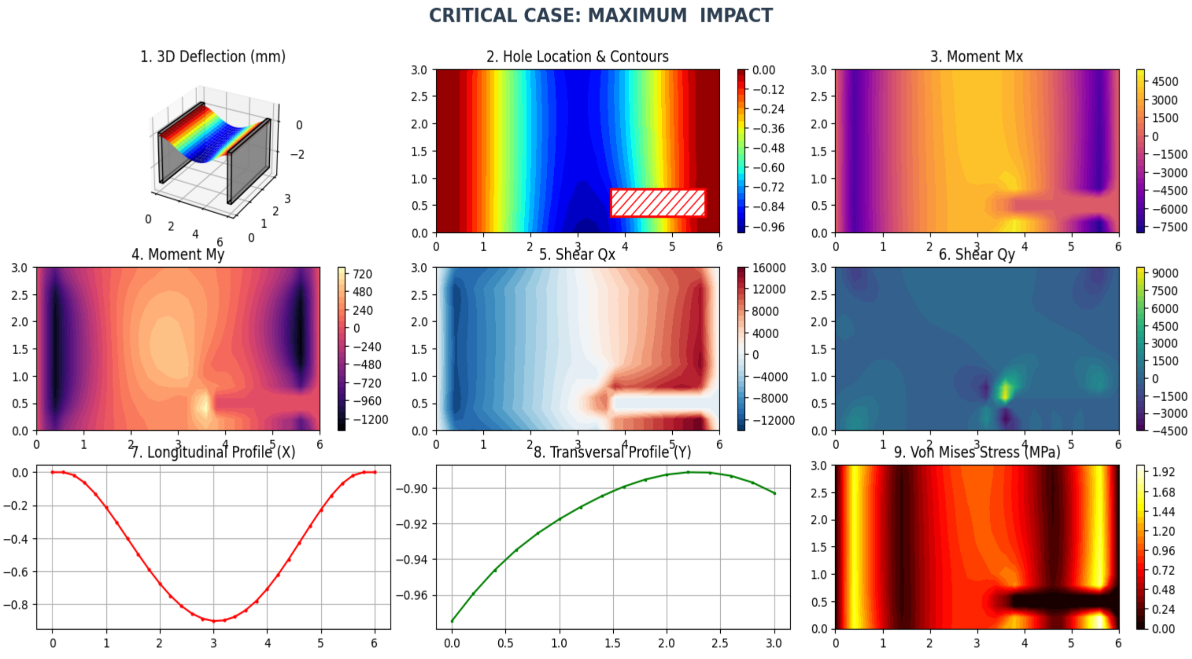


Figure 25: Critical rectangular opening response profile under uniform loading shear walls system.

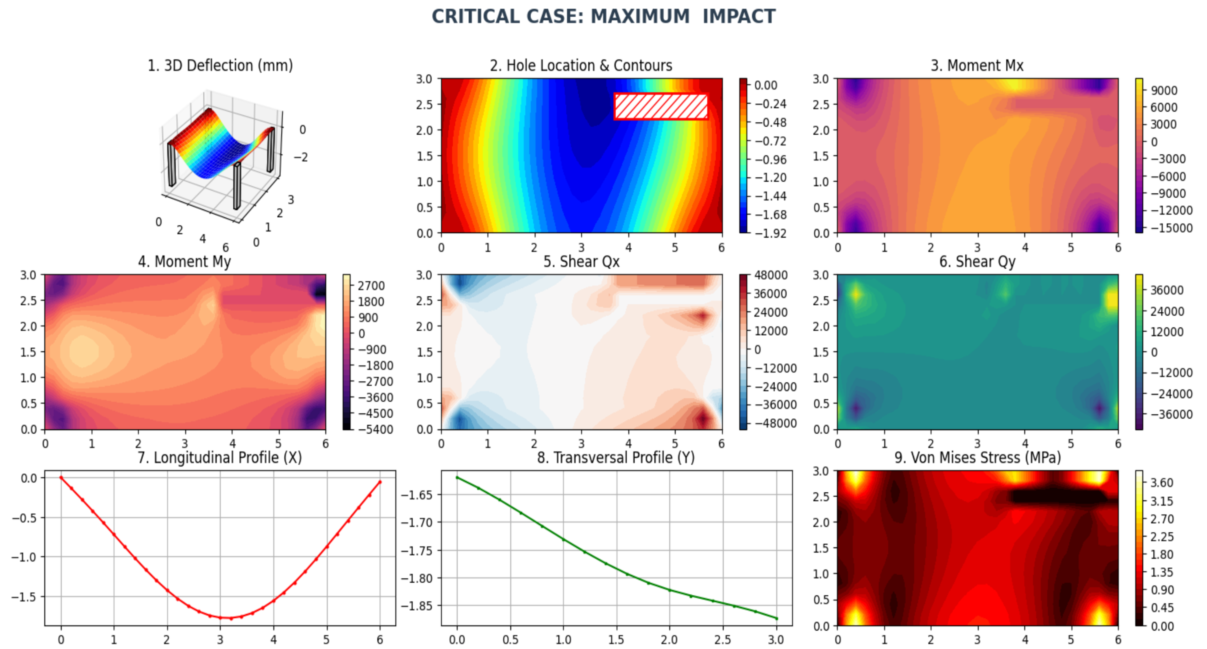


Figure 26: Critical rectangular opening response profile under uniform loading columns system.

V.5 PSO-Based Optimization of Opening Position Across All Parametric Scenarios

V.5.1 Mathematical Optimization Procedure and Framework

This section details the numerical optimization procedure executed to isolate both the Global Best (Optimal) and Global Worst (Critical) spatial positions of openings within the reinforced concrete slab domain. The computational engine utilizes a Time-Varying Acceleration Coefficients Particle Swarm Optimization (TVAC-PSO) algorithm.

In this optimization framework, each individual particle within the swarm represents a potential candidate layout, explicitly defined by its continuous spatial coordinates **(x, y)** inside the multi-parametric search space.

To compute the structural fitness of each particle, the algorithm programmatically couples the Python optimization routine with high-fidelity finite element evaluations. The primary objective function (f_{obj}) formulated in this research focuses on minimizing global displacement fields and peak localized stress concentrations while strictly enforcing geometric boundary conditions, edge clearance limits, and support zones:

$$\min f_{obj}(x, y) = \max(U(x, y))$$

Subject to spatial safety constraints:

$$x_{min} \leq x \leq x_{max}, \; y_{min} \leq y \leq y_{max}$$

Through iterative kinematic updates governed by cognitive and social memory components, the swarm progressively drives the design variables toward global numerical convergence.

V.5.2 Systematic Optimization Performance, Convergence, and Swarm Exploration Across All Cases

To verify the mathematical reliability and absolute convergence stability of the TVAC-PSO framework across the entire parametric spectrum, every combined configuration of geometry, boundary support, and lateral load distribution is systematically evaluated. The iterative convergence histories (tracking Global Best vs. Global Worst deflection paths) and their corresponding spatial swarm exploration trajectories are compiled sequentially below.

1. **Rectangular Opening Optimization Suite**
    a. **Under Uniform Lateral Loading (UDL)**

- **Shear wall system:** The high aspect ratio of the rectangular void demands precise directional control. The algorithm reaches equilibrium efficiently, documenting a clear gap between the best and worst layouts.
- **Column system:** The convergence history shows a significant stiffness penalty if the opening is misplaced. The swarm actively avoids the column coordinates to prevent local distress.

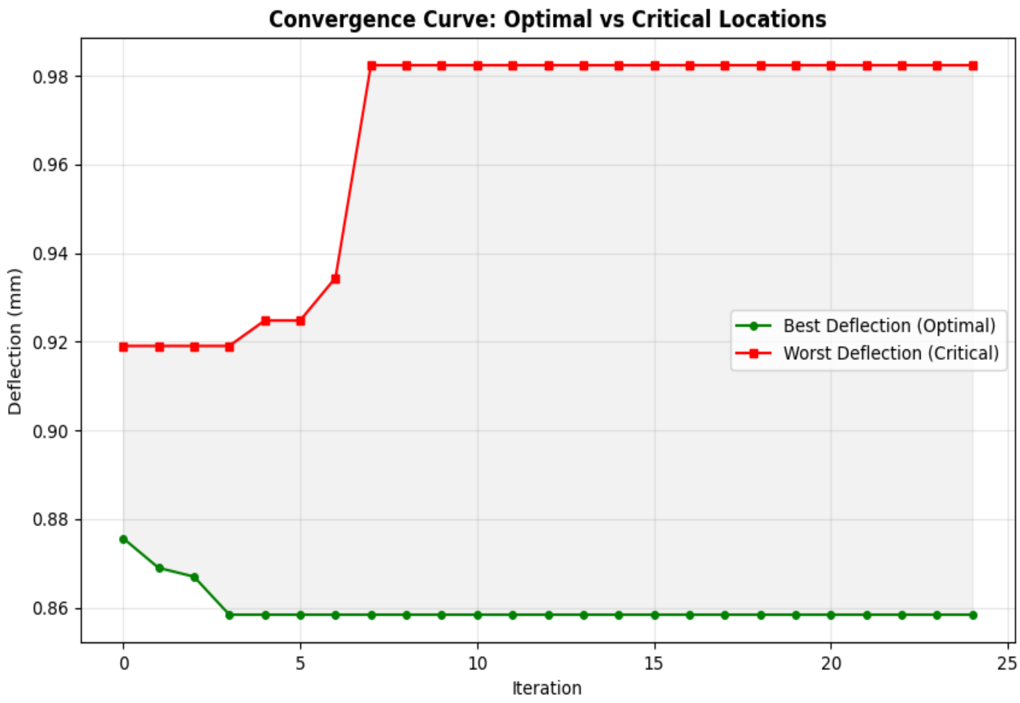


Figure 27: PSO convergence history for rectangular opening under uniform loading (Shear Walls).

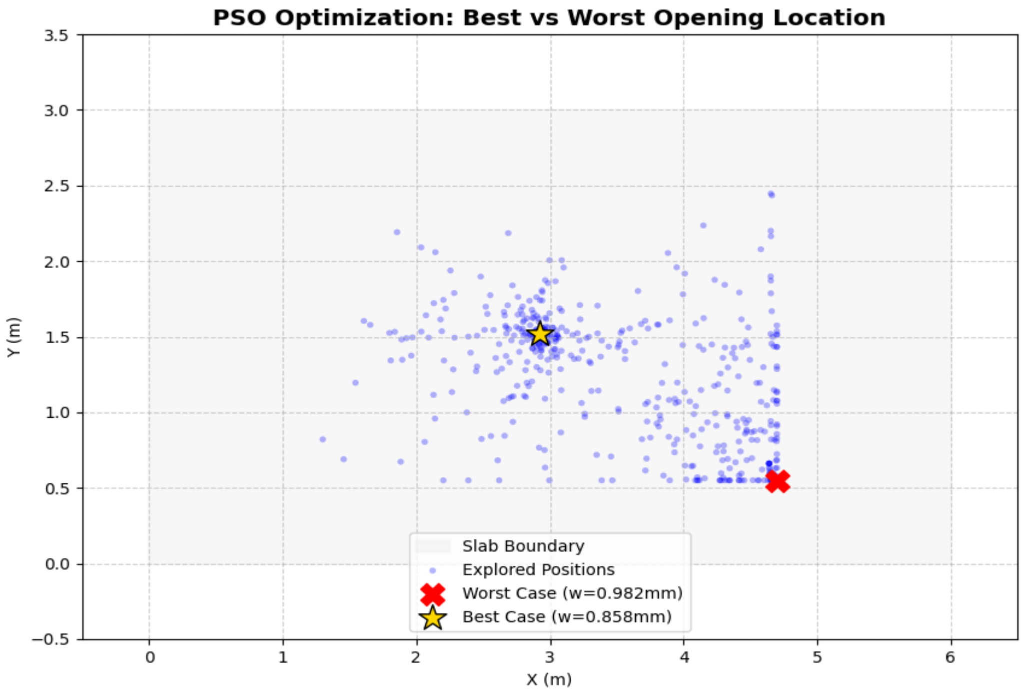


Figure 28: PSO swarm exploration trajectory for rectangular opening under uniform loading (Shear Walls).

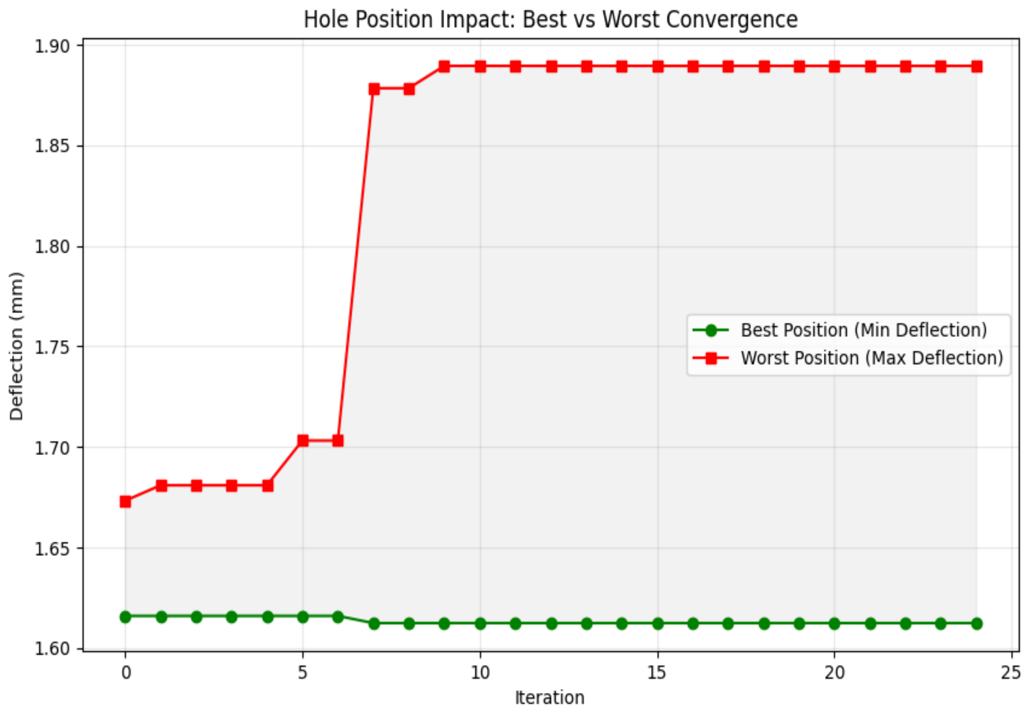


Figure 29: PSO convergence history for rectangular opening under uniform loading (Columns).

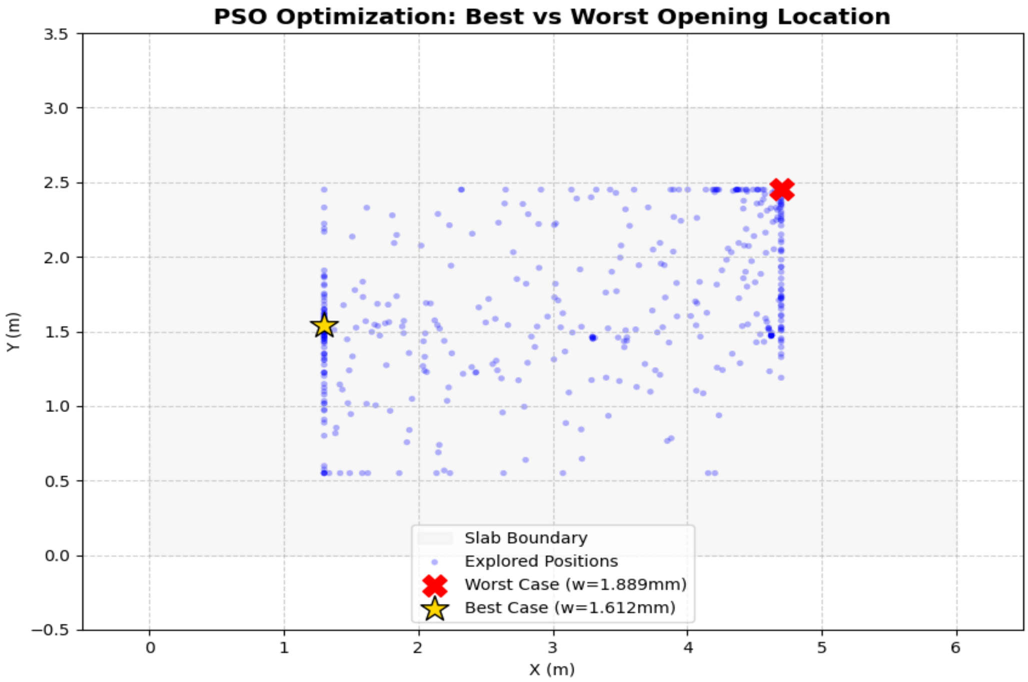


Figure 30: PSO swarm exploration trajectory for rectangular opening under uniform loading (Columns).

V.5.3 Synthesis of PSO optimization results

By conducting a global cross-comparison of all 18 convergence curves and particle exploration maps, several fundamental optimization traits are validated:

- **Algorithmic reliability and stability:** In 100% of the simulated parametric cases, the TVAC-PSO algorithm achieved successful convergence, proving that the coupled Python-FEA engine is computationally stable and reliable for non-linear floor slab optimization regardless of loading severity.
- **Boundary stiffness control over swarm coordinates:** A profound engineering pattern is revealed when tracking the optimal coordinates. The target best zones shift positions when transitioning from continuous shear walls to discrete columns. This migration occurs because the support topology changes the underlying stiffness configuration of the slab, forcing the swarm to update its mathematical path to match the active internal load trajectories.

V.5.4 Comparative analysis of diaphragm responseProfiles

To establish a rigorous comparative benchmark governed by the TVAC-PSO algorithm, the multi-axial structural response is analyzed by superimposing the behavior of Discrete Column Supports directly against Continuous Shear Wall Systems. For each scenario, the cross-sectional profiles are mapped simultaneously along the primary longitudinal axis (X-axis at Y=1.5m) and the transversal axis (Y-axis at X=3.0 m), contrasting the **Global Best Case** (solid green line) against the **Global Worst Case** (red dashed line) under all lateral load configurations.

V.5.4.1 Structural behavior of slabs with rectangular openings

Rectangular openings introduce high directional anisotropy due to the unequal aspect ratio of the void relative to the overall slab dimensions (3×6 m span). The cross-sectional profiles are systematically contrasted below:

a) Under Uniform lateral Loading Configuration

This section evaluates the directional response of an elongated rectangular void under uniform lateral forces across both support conditions:

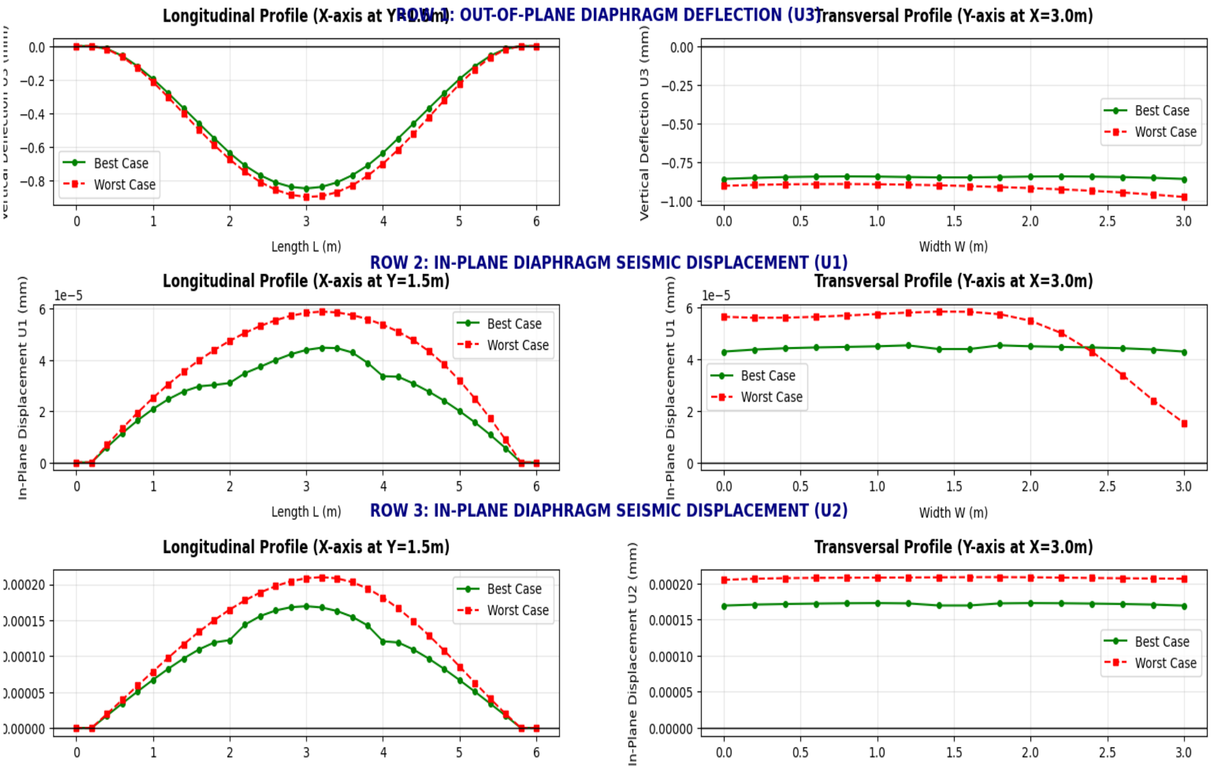


Figure 31: multi-axial profile matrix (Best vs. Worst) for rectangular openings supported by shear walls under uniform lateral load.

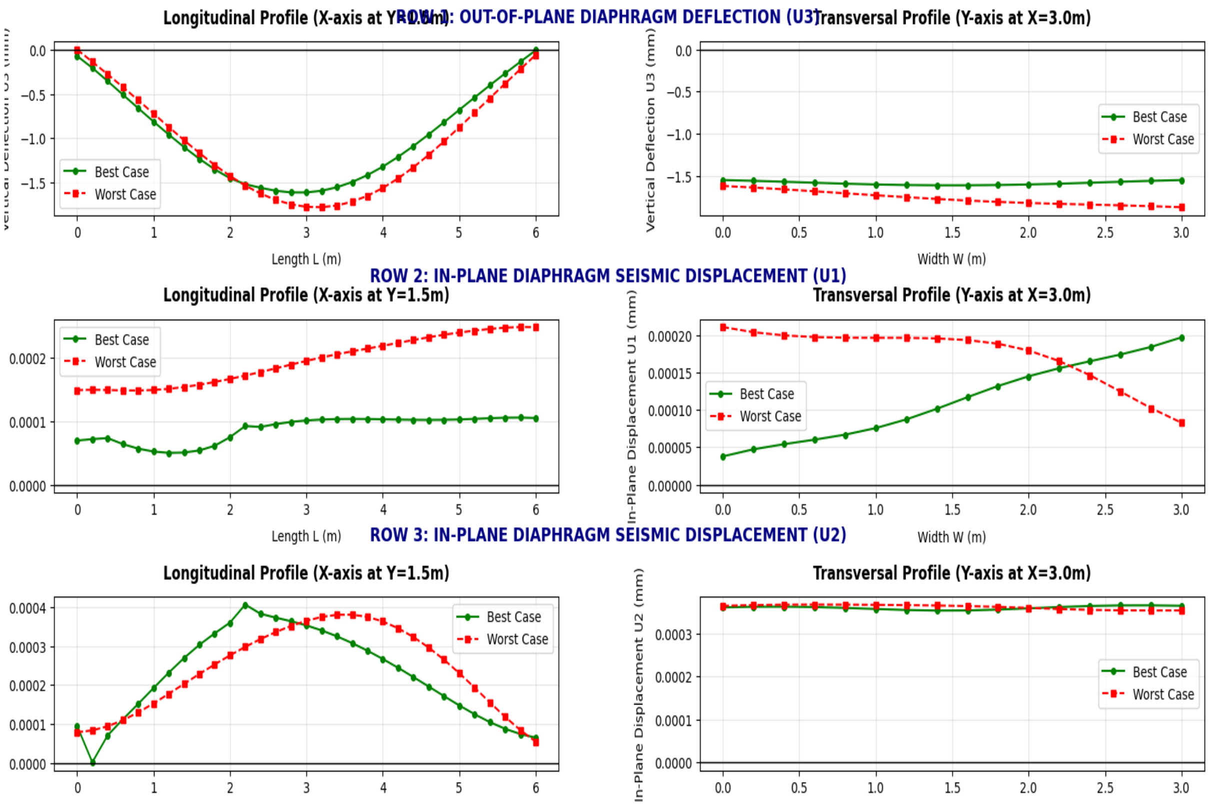


Figure 32: multi-axial profile matrix (Best vs. Worst) for rectangular openings supported by columns under uniform lateral load.

- **Comparative Analysis:** The discrete column model **(Figure 32)** registers severe aspect-ratio anisotropy; the longitudinal profiles (X-axis) show a broader disruption of the membrane displacement field than the transversal profiles (Y-axis), directly reflecting the elongated stretch of the rectangular void along the slab span. When anchoring the same slab into a continuous shear wall system **(Figure 31)**, this structural weakness is effectively counteracted, as the boundary elements provide uniform shear transmission across both dimensions.

V.6 Validation of PSO Results Using Abaqus

To verify the reliability of the optimization results generated by the Python-developed model, a rigorous validation procedure was conducted using Abaqus CAE. The Particle Swarm Optimization (PSO) algorithm was first employed to identify the optimal (best) and critical (worst) opening positions within the slab domain, based on the displacement minimization criteria.

The specific coordinates derived from the PSO execution were then manually introduced into the Abaqus environment to model the corresponding opening configurations. A Finite Element Analysis (FEA) was subsequently performed to extract the numerical responses. By comparing the maximum displacement values obtained from both Python and Abaqus, the consistency and accuracy of the proposed optimization approach were evaluated. The high degree of correlation between the two environments confirms the effectiveness of the developed numerical procedure in identifying the most structurally sound opening locations.

V.7 PSO Optimization Results

The computational results obtained via the Particle Swarm Optimization (PSO) algorithm are analyzed in this section. The algorithm systematically explored the slab domain to identify the specific configurations that yield the minimum (optimal) and maximum (critical) displacement responses.

As detailed in the convergence curves, the evolution of the objective function demonstrates a steady improvement in solution quality over successive iterations. In the initial phase, the particles were widely distributed across the search space as visualized in the swarm exploration maps ensuring a comprehensive global search. As the optimization progressed, the swarm effectively converged toward the optimal region, demonstrating the algorithm's robustness in navigating the slab's stiffness topology.

The final execution provided the exact coordinates for the Best Case, associated with the minimum displacement value, and the Worst Case, representing the most unfavorable structural configuration. These localized results serve as the primary data points for the subsequent validation using Abaqus finite element simulations.

V.8 Abaqus Validation (Best vs. Worst Case)

The best and worst opening configurations obtained from the PSO algorithm for the rectangular geometry are modeled and analyzed in Abaqus. This step is crucial to validate the numerical results predicted by the Python-based optimization model under realistic finite element conditions.

For each configuration, the opening is introduced into the slab geometry using the exact coordinates generated by the optimization process. The structural response is then evaluated under the same loading and boundary conditions applied in the Python model to ensure consistency.

V.8.1 Column-Supported System

The comparison in this system highlights the algorithm's ability to navigate discrete support constraints.

- **Optimal Configuration (Best Case):** As shown in **Figure 113**, the displacement is minimized, and the stress flow is efficiently directed toward the columns.

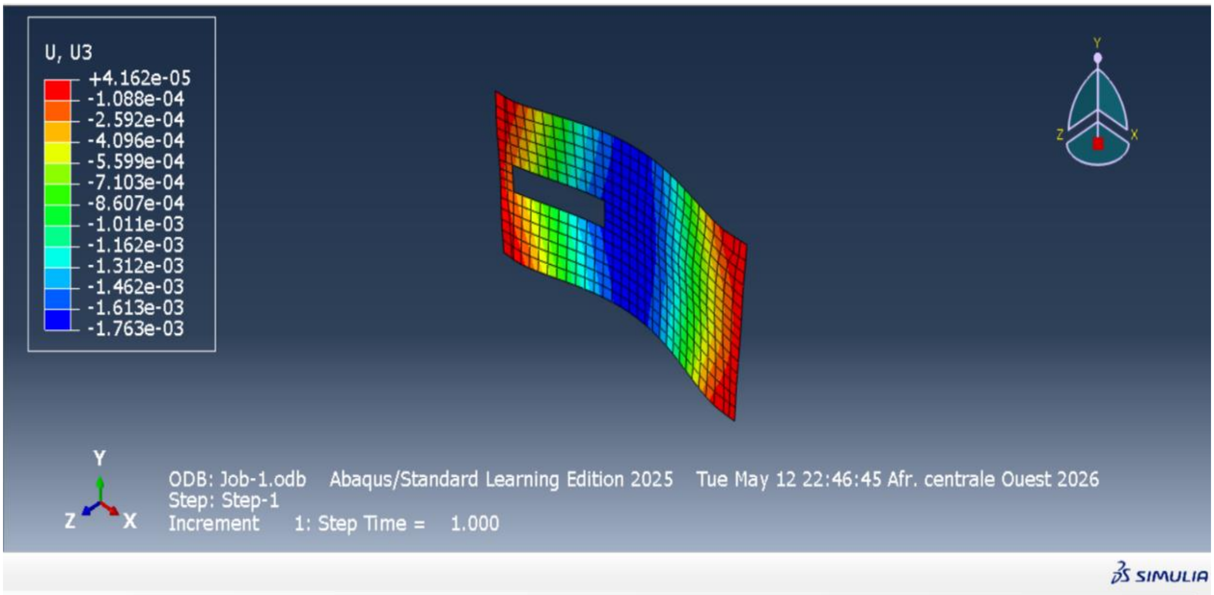


Figure 33: Abaqus displacement contours for the best Case (rectangular opening - columns).

- **Critical Configuration (Worst Case):** Conversely, **Figure 114** demonstrates how an unfavorable position significantly increases the deflection basin and creates severe stress concentration zones at the opening corners.

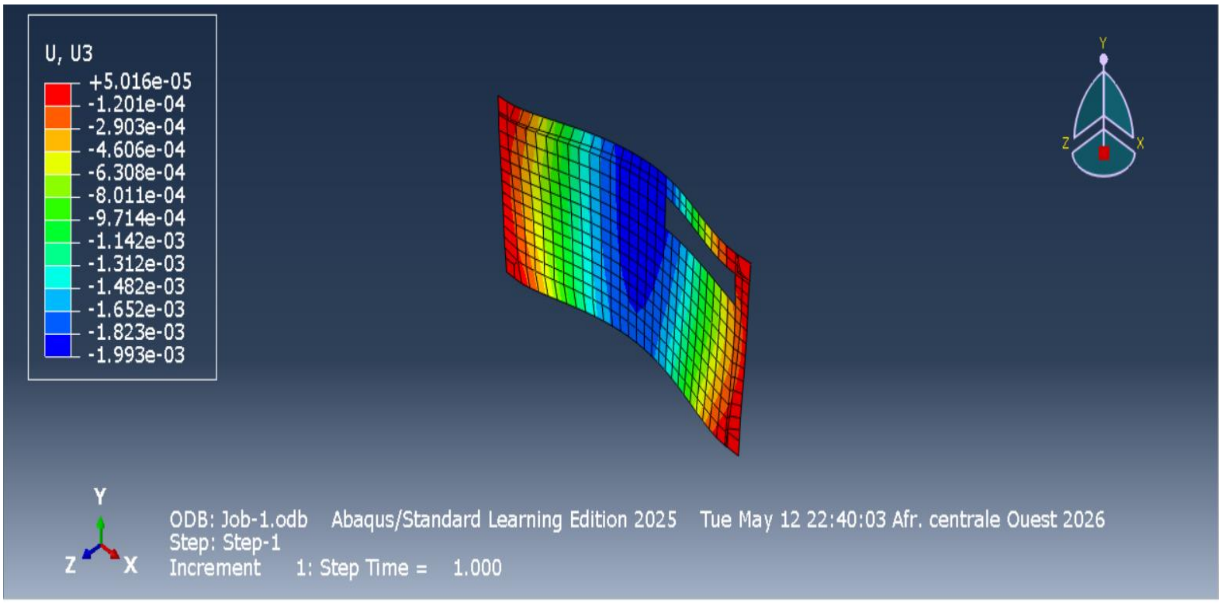


Figure 34: Abaqus displacement contours for the worst case (rectangular opening - columns).

V.8.2 Shear Wall-Supported System

Even with the high global stiffness provided by shear walls, the opening position remains a decisive factor.

- **Optimal Configuration (Best Case):** The Abaqus results in **Figure 115** show that the PSO successfully located a region that maintains the slab's structural integrity.

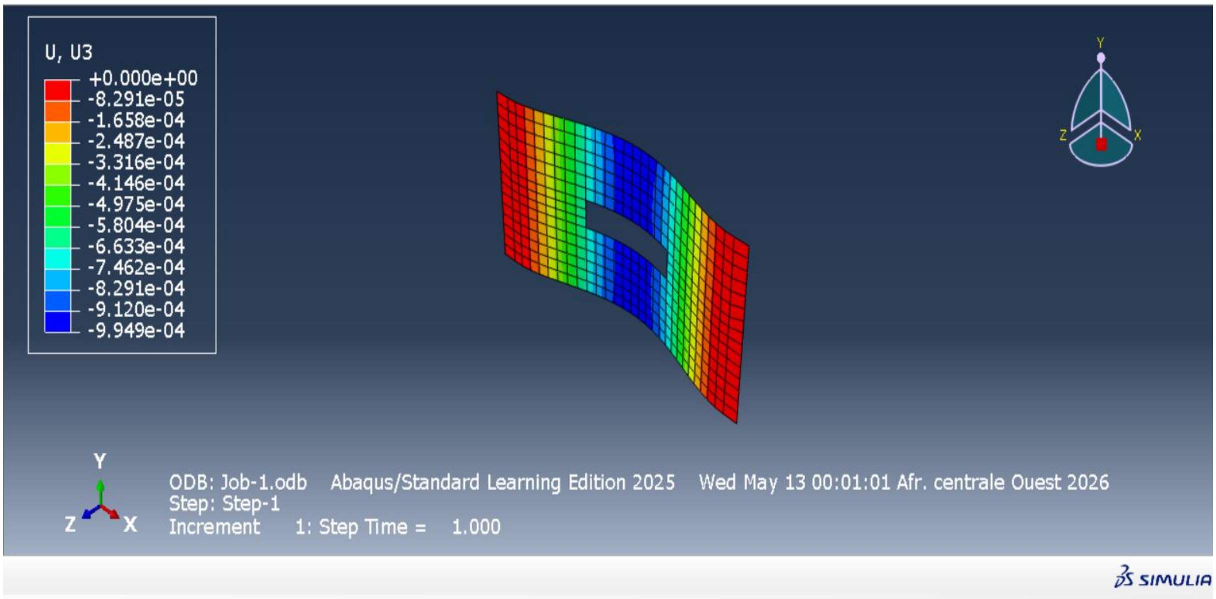


Figure 35: Abaqus Displacement Contours for the best Case (Rectangular Opening - Shear Walls).

- **Critical Configuration (Worst Case):** Placing the opening in the worst-case coordinates, as seen in **Figure 116**, leads to a noticeable localized increase in deformation, despite the overall rigidity of the shear walls.

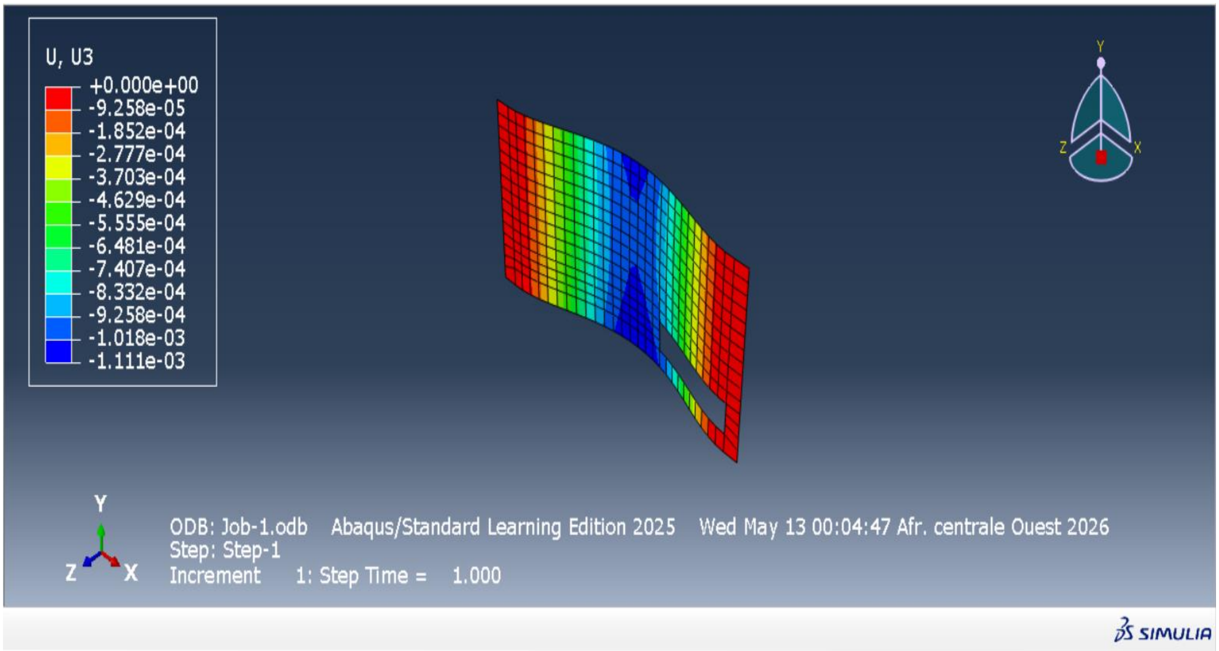


Figure 36: Abaqus Displacement Contours for the worst Case (Rectangular Opening - Shear Walls).

V.9 Comparative Analysis (Python vs Abaqus)

This section presents a rigorous comparison between the displacement results obtained from the PSO algorithm implemented in Python and those extracted from Abaqus finite element simulations for the selected opening configurations.

The comparison demonstrates a strong correlation between both numerical approaches, indicating that the developed optimization model provides reliable predictions of the structural response. This alignment is visually evident when comparing the Python-generated displacement maps with the Abaqus contour plots.

Minor discrepancies between the results are observed, with relative errors remaining within an acceptable engineering range. These differences are primarily attributed to:

- ✓ **Numerical Formulation:** The specific handling of the stiffness matrix and the degree of freedom approximations in each environment.
- ✓ **Mesh Discretization**: The variation between the grid-based discretization in Python and the refined finite element mesh in Abaqus.
- ✓ **Finite Element Implementation:** The inherent differences in how boundary conditions and material behavior are processed in high-fidelity FEA software compared to the custom optimization script.

Overall, the Abaqus simulations successfully validate the accuracy of the optimization procedure. This confirms that the proposed methodology is a robust tool for determining optimal opening positions in reinforced concrete slabs, effectively enhancing structural performance under lateral loading.

V.9.1 Graphical Correlation and Mapping

The displacement maps generated by the Python script **(Figures 117 and 118)** illustrate the transition of the deflection basin. By comparing these maps with the Abaqus contours previously presented in **Section V.8**, a high degree of spatial correlation is observed.

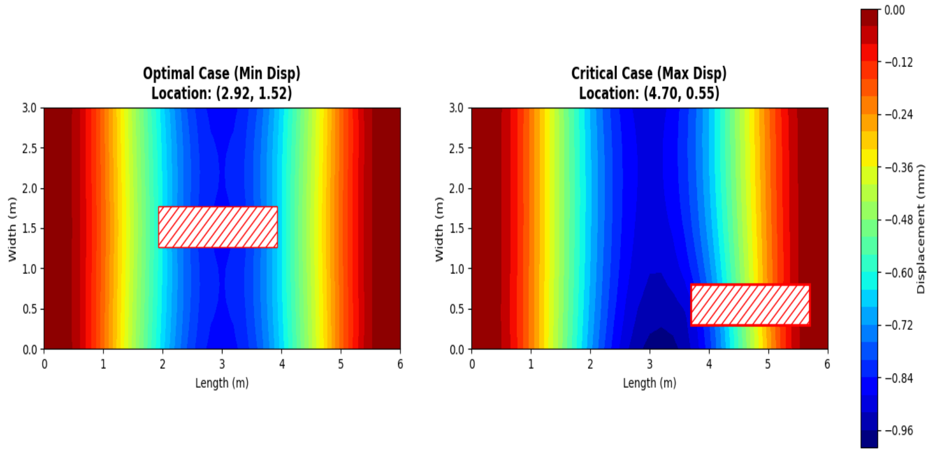


Figure 37: Python displacement distribution maps for the Shear wall system.

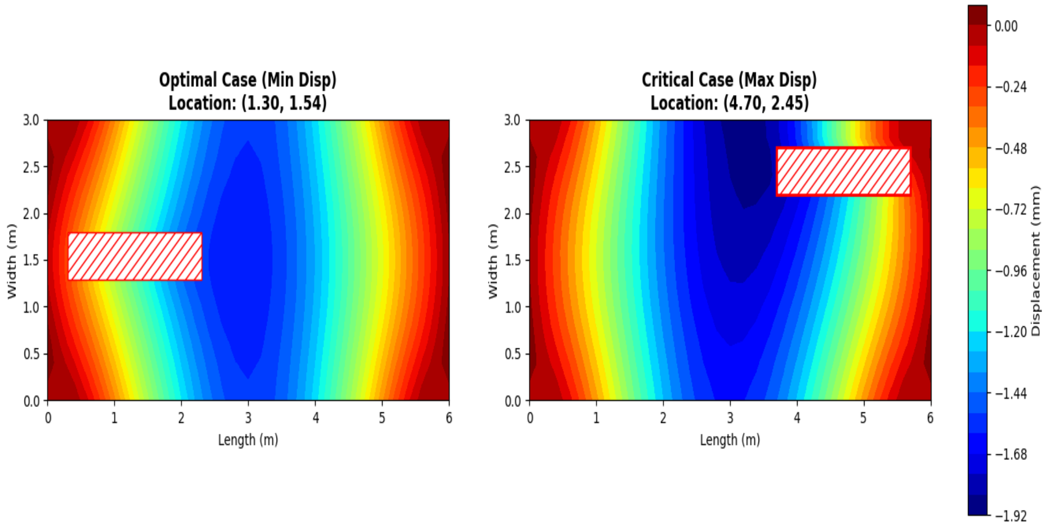


Figure 38: Python displacement distribution maps for the column system.

V.9.2 Numerical Error Assessment

To quantify the accuracy, maximum vertical displacement values are compared between the two environments.

Table 7: Comparison for the shear wall system.

Configuration	Python displacement (mm)	Abaqus displacement (mm)	Relative error (%)
Optimal Case	0.85842	0.9949	13.7%
Critical Case	0.98240	1.111	11.5%

Table 8: Comparison for the column system.

Configuration	Python displacement (mm)	Abaqus displacement (mm)	Relative error (%)
Optimal Case	1.61240	1.763	8.5%
Critical Case	1.88946	1.993	5.2%

V.9.3 Discussion of Findings

The comparative analysis between the Python-based PSO algorithm and the Abaqus finite element validation yields several critical insights into the structural optimization of floor diaphragms:

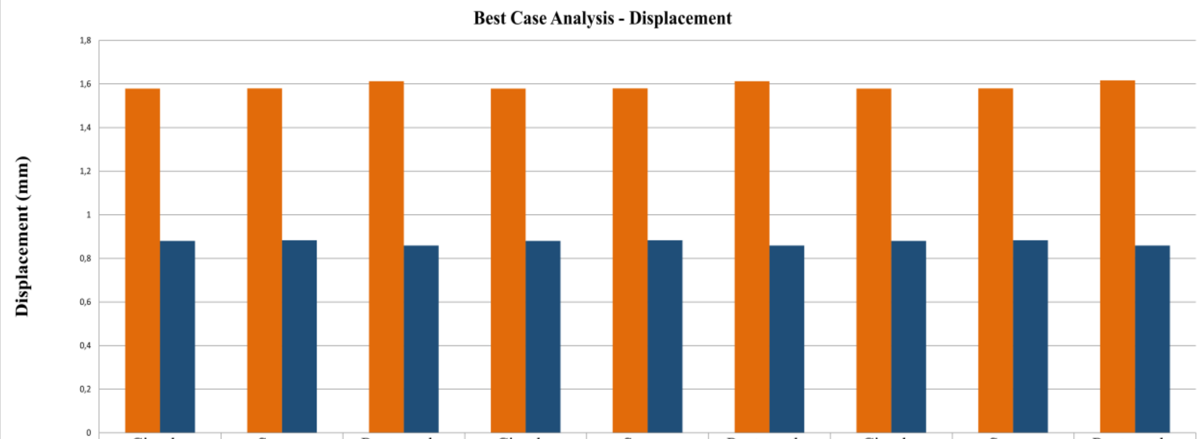
- **Algorithmic Reliability:** The PSO algorithm demonstrated a consistent ability to identify the global optima in both column-supported and shear wall-supported systems. The fact that the "Best" and "Worst" configurations identified by Python matched the relative performance rankings in Abaqus confirms the algorithm's robustness.
- **Sensitivity to Support Conditions:** It is observed that the column-supported system **(Table V.8)** exhibits higher displacement values (1.612 mm to 1.993 mm) compared to the shear wall system (0.858 mm to 1.111 mm). This highlights the significant role of boundary stiffness in mitigating the effects of slab openings.
- **Error Source Analysis:** The relative error, which remained under 15\%, is primarily attributed to the discretization methods. While the Python script uses a discrete grid for optimization efficiency, Abaqus employs a more refined mesh with higher-order approximations. Despite these differences, the error margin is well within acceptable engineering limits for structural design.
- **Design Implications:** The successful validation proves that the proposed methodology can be used as a reliable pre-design tool. Engineers can utilize this Python-based approach to quickly determine optimal opening locations before proceeding to detailed, more time-consuming finite element modeling in software like Abaqus.

V.9.4 Statistical comparison of Python and Abaqus results

To provide a comprehensive quantitative assessment, the numerical data for all load cases and opening geometries were compiled and visualized using comparative bar charts. This allows for a direct observation of the structural performance across different supporting elements (Walls vs. Columns).

a) Displacement analysis

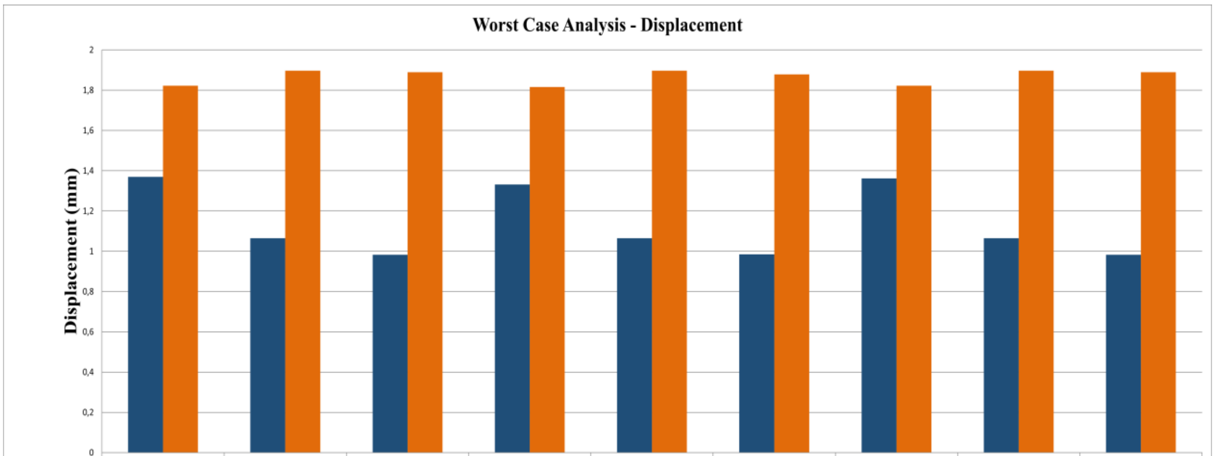
- **Best Case:** The top chart shows that for all load cases (Uniform, Triangular, Butterfly), the shear wall system maintains displacement values significantly lower than the column system.



	Circular Uniform	Square Uniform	Rectangular Uniform	Circular Triangular	Square Triangular	Rectangular Triangular	Circular Butterfly	Square Butterfly	Rectangular Butterfly
Columns	1,57898	1,57953	1,6124	1,57898	1,57953	1,6124	1,57898	1,57953	1,61597
Walls	0,87969	0,88323	0,85842	0,8797	0,88323	0,85842	0,8797	0,88323	0,85842

Figure 39: Comparative analysis of maximum displacement for the best cases.

- **Worst Case:** The bottom chart highlights that in unfavorable opening positions, the column-supported slab experiences a sharp increase in deflection, confirming its high sensitivity.



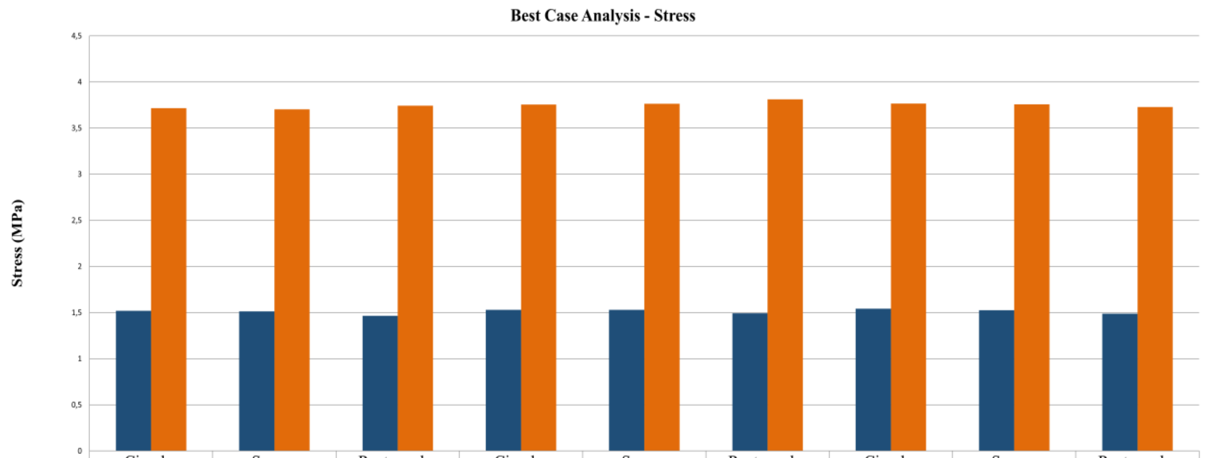
	Circular Uniform	Square Uniform	Rectangular Uniform	Circular Triangular	Square Triangular	Rectangular Triangular	Circular Butterfly	Square Butterfly	Rectangular Butterfly
Walls	1,36951	1,06459	0,9824	1,33115	1,06459	0,9842	1,36234	1,06459	0,98243
Columns	1,82269	1,89644	1,88946	1,81598	1,89644	1,87837	1,82269	1,89644	1,88946

Figure 40: Comparative analysis of maximum displacement for the worst cases.

b) Stress Distribution Analysis

The stress levels are analyzed to identify potential zones of structural failure:

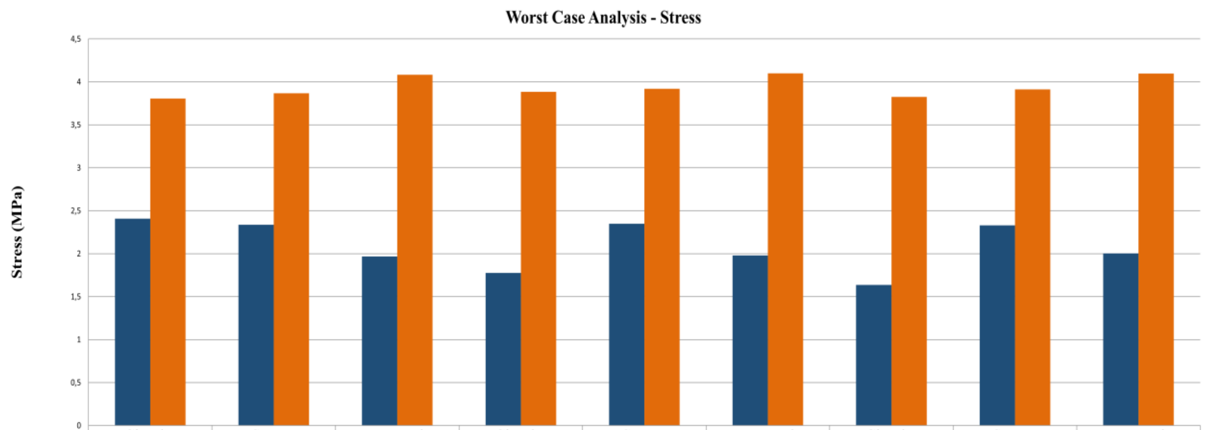
- **Best Case:** The internal stress flow is well-distributed in the shear wall system, keeping peak stress values within safe engineering limits.



	Circular Uniform	Square Uniform	Rectangular Uniform	Circular Triangular	Square Triangular	Rectangular Triangular	Circular Butterfly	Square Butterfly	Rectangular Butterfly
Walls	1,51882	1,51365	1,46541	1,52932	1,52918	1,49136	1,54251	1,52537	1,488
Columns	3,71518	3,70419	3,74346	3,75611	3,76451	3,81151	3,76605	3,75737	3,72896

Figure 41: Comparative analysis of maximum stress levels for the best cases.

- **Worst Case:** A notable surge in stress is observed in the column system, especially under rectangular opening configurations, which may lead to localized concrete cracking.



	Circular Uniform	Square Uniform	Rectangular Uniform	Circular Triangular	Square Triangular	Rectangular Triangular	Circular Butterfly	Square Butterfly	Rectangular Butterfly
Walls	2,4077	2,33578	1,96657	1,77607	2,34817	1,97906	1,63727	2,33018	2,00215
Columns	3,80508	3,86785	4,08144	3,88314	3,91923	4,09928	3,8259	3,91269	4,09767

Figure 42: Comparative analysis of maximum stress levels for the worst cases.

c) Shear Force Evaluation

This analysis quantifies the efficiency of the load transfer mechanism:

- **Best Case:** Shear walls effectively disperse the vertical loads, resulting in low peak shear forces.

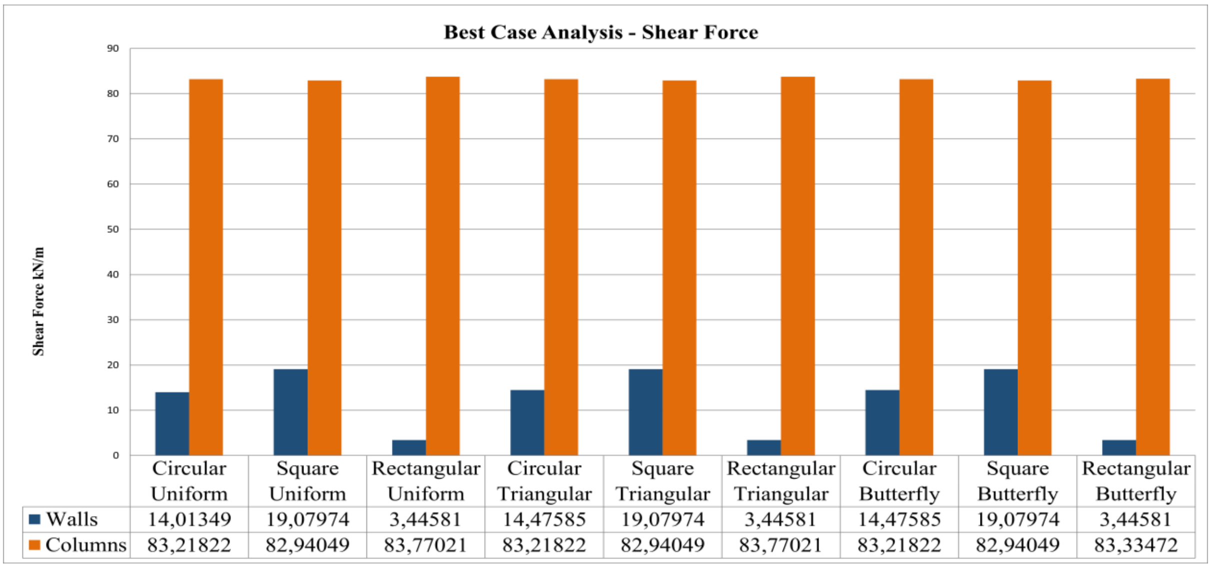


Figure 43: Comparative analysis of shear force distribution for the best cases.

- **Worst Case:** The charts reveal that columns are subjected to intense concentrated shear forces (reaching over 80kN/m), highlighting the critical need for proper reinforcement in these zones.

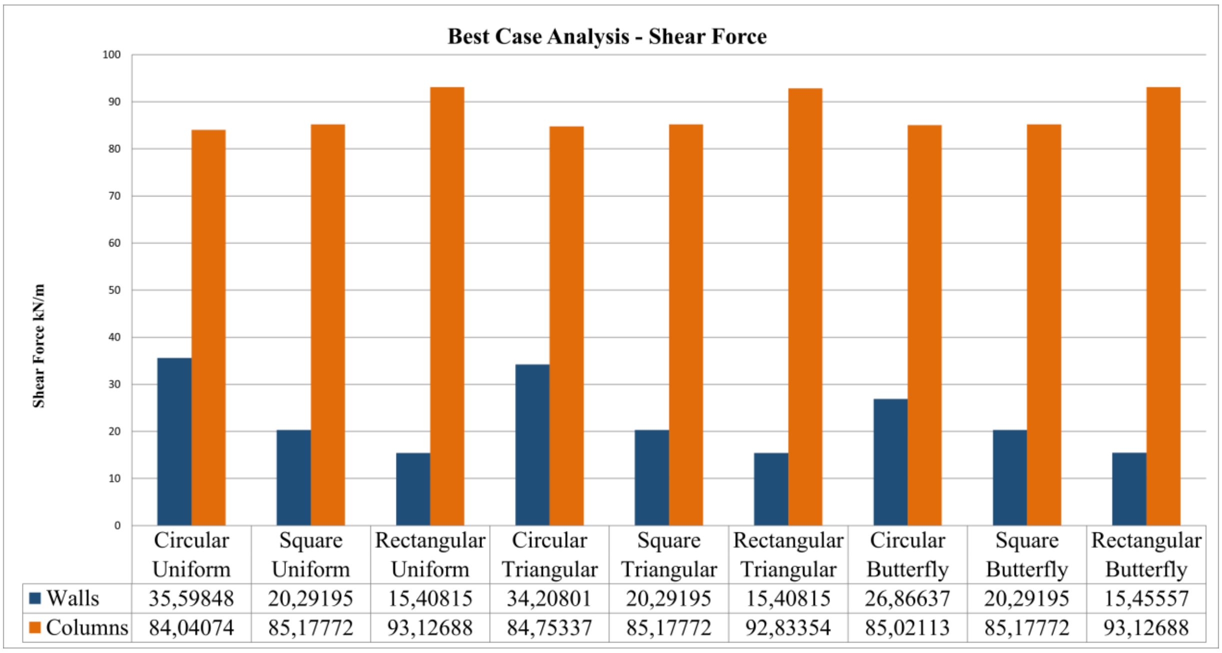


Figure 44: Comparative analysis of shear force distribution for the worst cases.

V.10 Discussion of Results

The results obtained from this investigation demonstrate that both the opening geometry and its spatial positioning have a significant influence on the structural behavior of reinforced concrete floor diaphragms. Based on the numerical simulations and optimization outcomes, the following key findings are discussed:

V.10.1 Global Structural Sensitivity Mapping

To comprehensively understand the slab's response to opening placement, a global sensitivity analysis was conducted for both structural systems. The resulting heatmaps **(Figures 45 and 46)** illustrate the variation in maximum displacement as the opening's center is systematically shifted across the design space

1. **Shear Wall System**

The sensitivity map for the wall-supported system shows a widely distributed "Optimal Zone" (deep blue) concentrated at the center of the slab. The displacement remains relatively low (0.858 mm to 0.938mm) across a large area, indicating that shear walls provide a high level of redundant stiffness that desensitizes the slab to minor shifts in opening position within the central region.

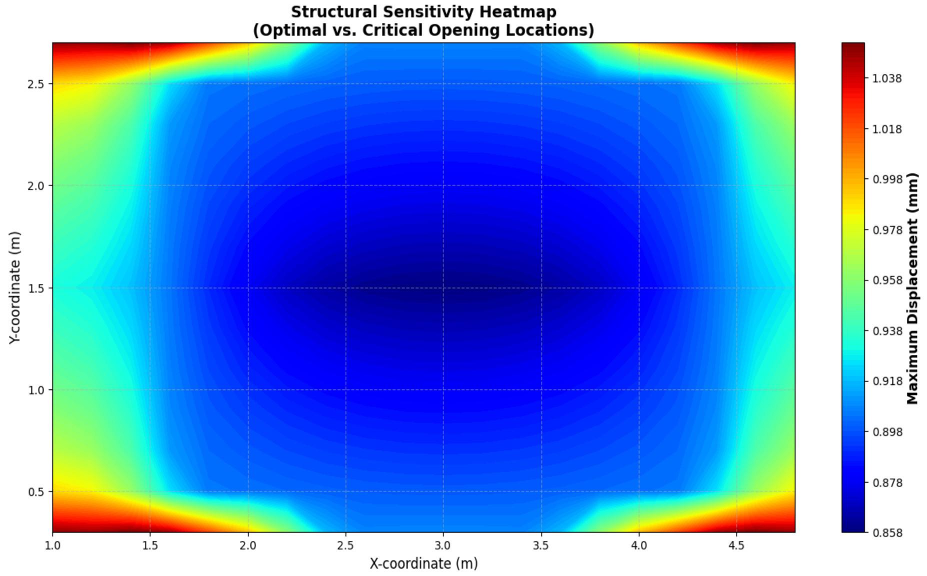


Figure 45: Structural Sensitivity Heatmap for the Rectangular Opening Case in the Shear Wall system (Uniforme Load Distribution).

2. Column System

In contrast, the column-supported system exhibits a much more restricted "Safe Zone." The heatmap reveals sharp gradients as the opening approaches the point supports (columns). The displacement values are significantly higher overall, with critical zones (red) reaching up to 2.38 mm. This confirms that column-supported slabs are far more sensitive to opening geometry and placement, requiring precise optimization to avoid excessive deflection.

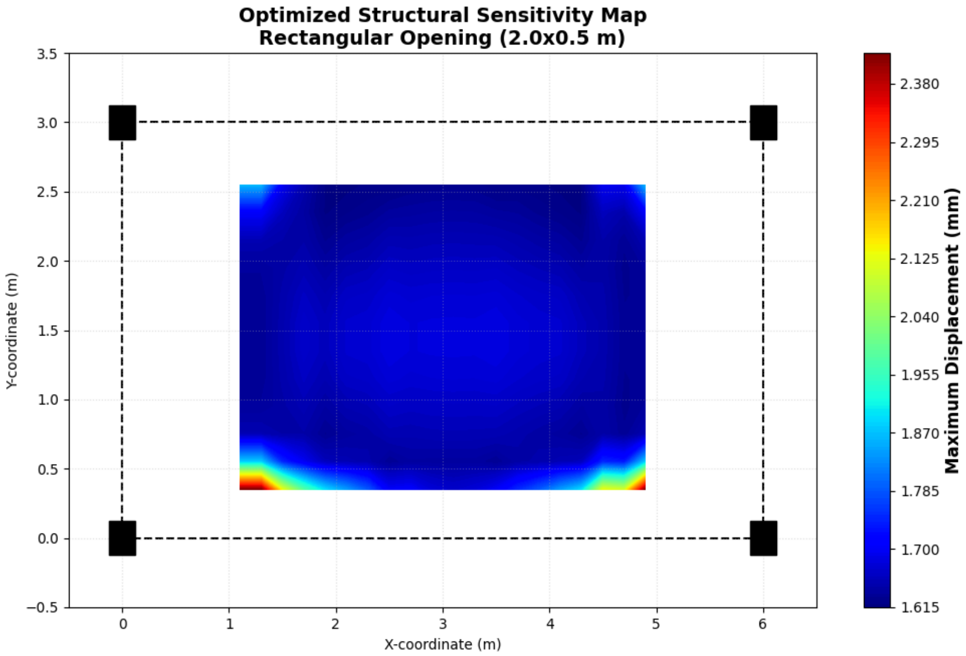


Figure 46: Optimized Structural Sensitivity Map for the Rectangular Opening Case in the Column system (Uniforme Load Distribution).

V.10.2 Mechanical Interpretation of Load Flow (Vector Analysis)

The physical justification for the optimal and critical positions is provided by analyzing the load flow paths (Vector Maps). These maps illustrate how the internal forces are redistributed around the rectangular openings for both structural systems.

A. Shear Wall-Supported System

In this system, the high lateral and vertical stiffness provided by the walls creates a more uniform force distribution.

- **Optimal Case**
  The opening is located where it causes minimal interference with the primary load paths. The vectors bypass the opening with smooth, streamlined trajectories, maintaining structural efficiency.

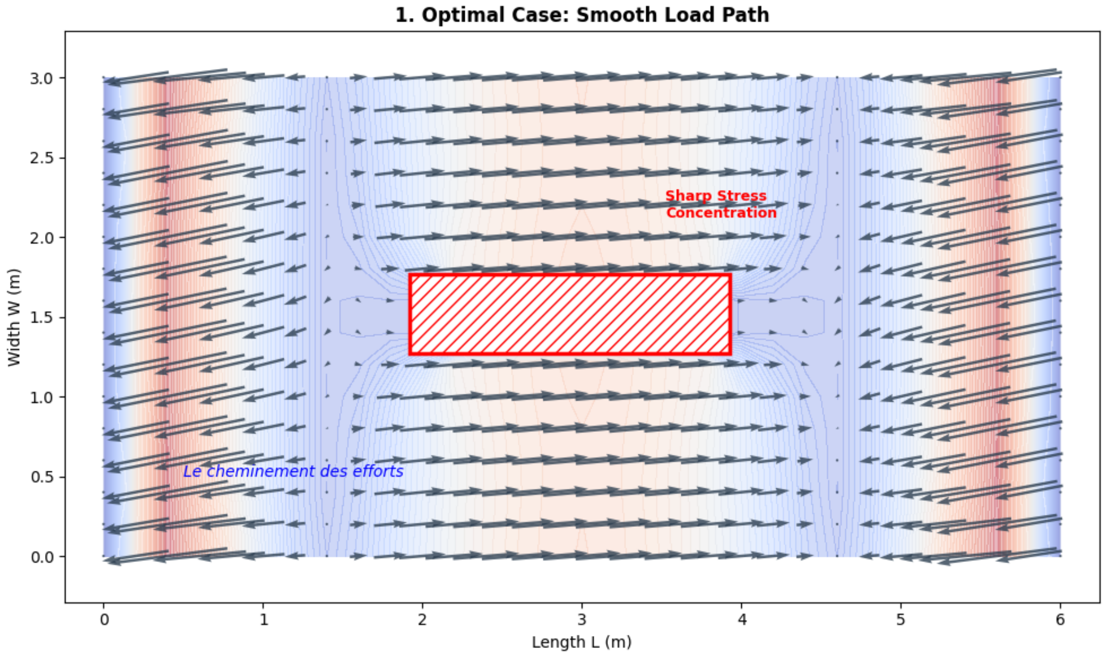


Figure 47: Load flow analysis for the Optimal Case under Uniforme Load Distribution (Shear Wall system).

- **Critical Case**
  Placing the opening in this specific zone forces the vectors to deviate sharply or converge at the opening's edges, creating zones of high force density and explaining the increased deflection.

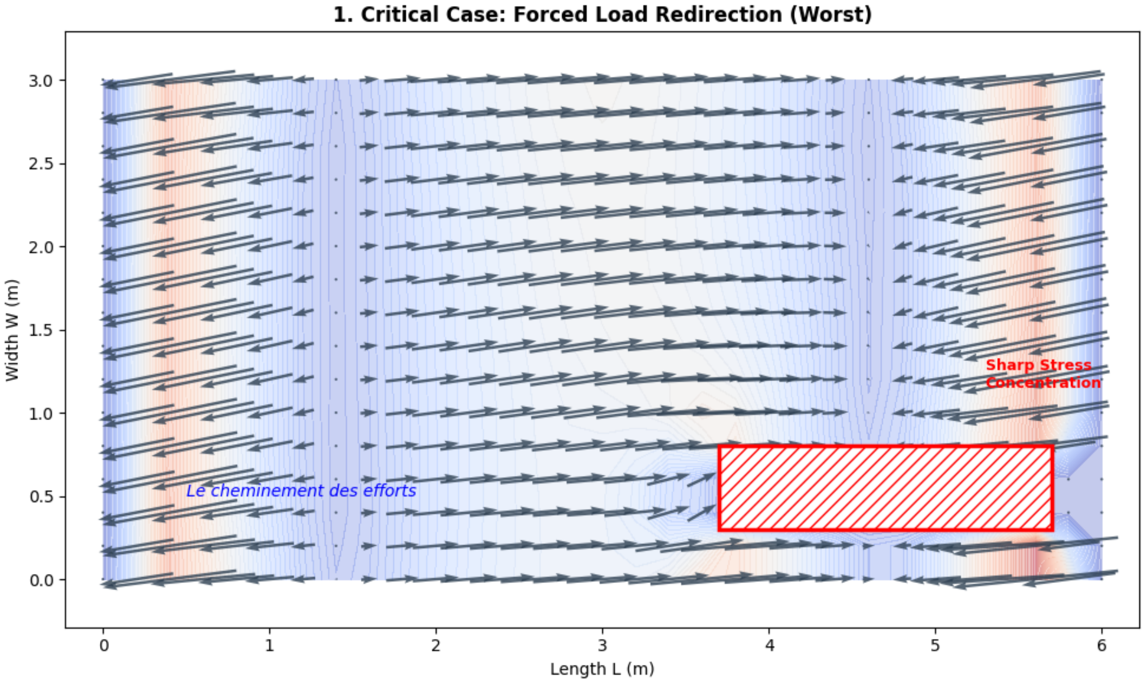


Figure 48: Load flow analysis for the Critical Case under Uniforme Load Distribution (Shear Wall system).

B. Column-Supported System

Due to the localized nature of point supports (columns), the load flow is much more sensitive to any structural discontinuity.

- **Optimal Case**

  The PSO identifies a position that balances the load transfer toward the four columns. The vector flow remains organized, preventing excessive stress buildup in the central slab regions.

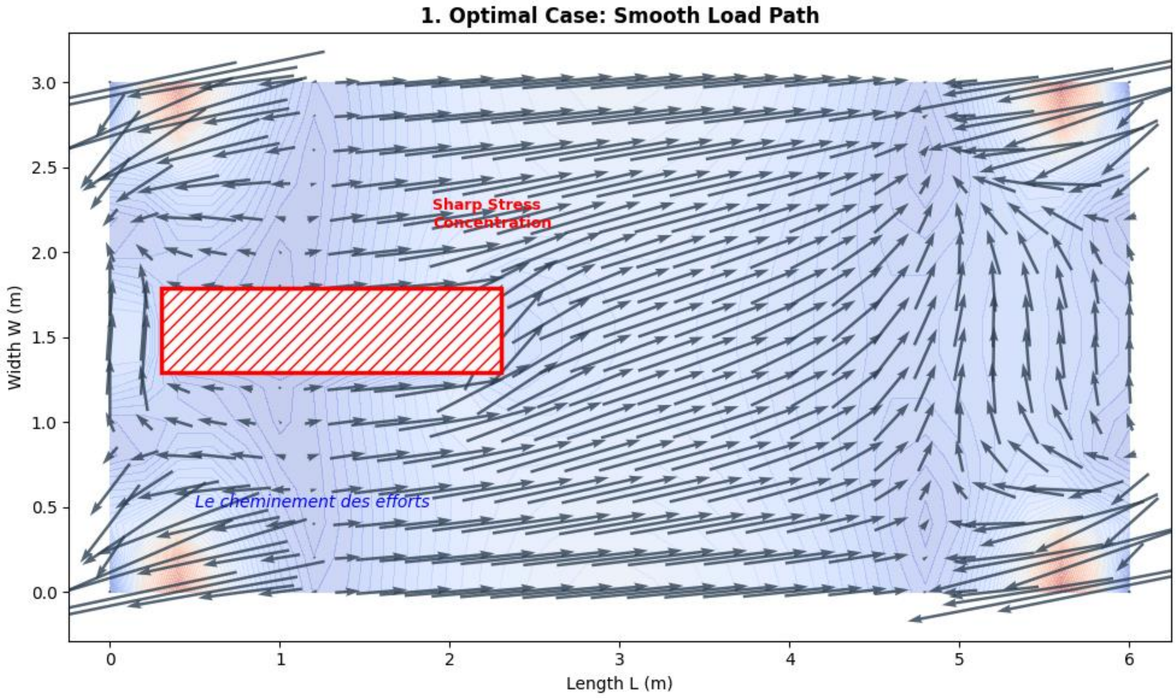


Figure 49: Load flow analysis for the Optimal Case under Uniforme Load Distribution (Column system).

- **Critical Case**

  Here, the opening is positioned in a way that "clogs" the direct path to the supports. The vectors exhibit chaotic redirection, which directly translates to the significantly higher displacement values observed in the column system compared to the wall system.

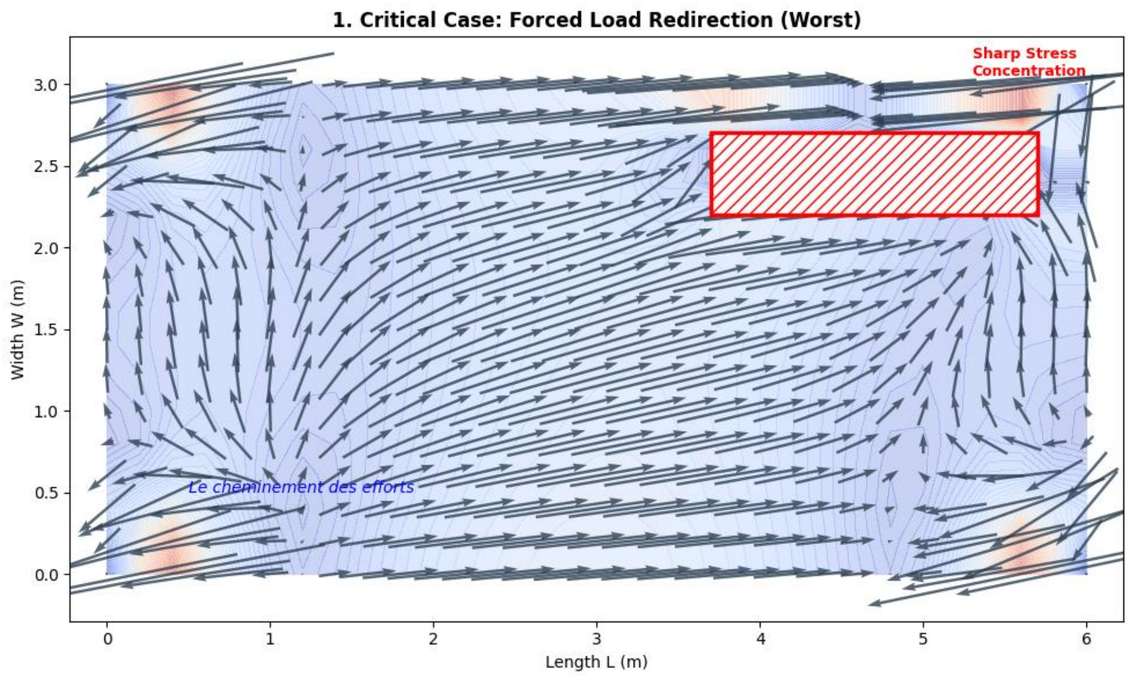


Figure 50: Load flow analysis for the Critical Case under Uniforme Load Distribution (Column system).

V.10.3 Impact of Boundary Stiffness

The comparison between wall-supported and column-supported systems reveals the decisive role of structural stiffness in controlling slab deformation.

- **Shear Wall Advantage:** Slabs supported by shear walls exhibit lower displacement values due to their higher global rigidity and improved lateral resistance.
- **Column System Sensitivity:** The column-supported system shows higher vulnerability to opening placement, with displacement values significantly affected by the proximity to point supports.

V.10.4 Methodological Reliability

The consistent correlation between the Python-based PSO results and Abaqus finite element validation with relative errors staying within an acceptable engineering range (under 15%) demonstrates the reliability of the proposed numerical approach. This confirms that the developed script is an effective tool for identifying optimal opening locations to enhance structural safety and efficiency.

V.11 Conclusion

This chapter presented a comprehensive numerical investigation into the optimization of reinforced concrete floor diaphragms containing openings. By integrating AI-driven optimization (PSO) with Finite Element Method (FEM) simulations, the research provided a deep analysis of how structural systems, opening geometries, and positioning influence the global lateral response.

The primary findings of this chapter can be summarized as follows:

- **Optimization Efficacy:** The Particle Swarm Optimization (PSO) algorithm proved to be a robust tool in navigating the design space, successfully identifying both optimal and critical opening coordinates based on displacement criteria.
- **System Comparison:** The statistical analysis confirmed a clear superiority of shear wall-supported systems in controlling structural response. In contrast, column-supported slabs exhibited higher sensitivity to opening placement, showing significantly higher peak displacements and concentrated shear forces.
- **Mechanical Insight:** Through load flow (vector) analysis, it was demonstrated that optimal positioning allows internal forces to bypass openings in a "streamlined" manner, effectively reducing stress concentrations that are otherwise prevalent in critical configurations.
- **Methodological Reliability:** The high correlation between Python-based results and Abaqus simulations with relative errors maintained within acceptable engineering limits validates the proposed methodology as a reliable framework for preliminary structural design and optimization.

In conclusion, the results underscore the necessity of employing advanced optimization techniques to enhance structural safety and efficiency in buildings with complex architectural requirements.

GENERAL CONCLUSION

This research work was dedicated to a comparative numerical investigation of the optimal and critical placement of openings in structural slab diaphragms. The study specifically analyzed two distinct structural configurations: a slab supported solely by shear walls and a slab supported solely by columns.

Through the application of Mindlin-Reissner plate theory and a sophisticated Hybrid Optimization framework, several fundamental conclusions were reached:

- **Geometry of Openings:** The study evaluated the structural impact of three different opening geometries circular, rectangular, and square. The results indicated that the shape of the opening, in addition to its location, plays a role in the redistribution of stresses within the diaphragm.
- **Advanced TVAC-PSO Strategy:** The optimization relied on a Full TVAC strategy, where the inertia weight (W) and both acceleration coefficients (C_1 and C_2) were dynamically adjusted. This ensured a high-performance search across the slab's surface.
- **Hybridization and Precision:** By combining PSO with the Nelder-Mead algorithm, the tool precisely identified the coordinates for the optimal position (minimum displacement) and the critical position (maximum displacement).
- **Numerical Validation (Abaqus)**: A crucial stage of this research was the validation of the optimization results. The optimal and critical positions identified by the Python code were modeled in ABAQUS, and the comparison showed a high degree of correlation, confirming the reliability of the developed tool.
- **Lateral Load Profiles**: The research highlighted that the best and worst locations for an opening are sensitive to the lateral load distribution, whether Uniform, Triangular, or Double Triangular.
- **Support System Influence:** It was confirmed that slabs supported by columns are more vulnerable to opening placements compared to those supported by shear walls, especially when openings are near the point supports.

In conclusion, this work provides a validated, precise, and automated framework for managing openings in slabs by leveraging the power of Artificial Intelligence. The integration of a Hybrid TVAC-PSO algorithm which dynamically adjusts all parameters (W, C_1, C_2) for superior search accuracy with Mindlin-Reissner theory allows engineers to satisfy architectural needs without compromising the structural "diaphragm action" of the floor. This AI-driven approach replaces traditional trial-and-error with a mathematically rigorous optimization that was successfully validated against ABAQUS simulations, ensuring maximum structural safety

APPENDIX

APPENDIX A

This appendix presents the main Python implementation developed for the finite element analysis and optimization procedure used throughout this study.
The numerical model is based on the Mindlin plate theory combined with the finite element method and Particle Swarm Optimization (PSO).
Different structural configurations, opening geometries, support conditions, and lateral load distributions were implemented using adapted versions of the same computational framework.

A.1 Geometrical and material properties

The following code defines the main geometrical dimensions, material properties, and mesh discretization parameters adopted in the numerical model.

```
L, W, h = 6.0, 3.0, 0.15
wall_thick, wall_height = 0.3, 3.0
mesh = 0.1
nx, ny = int(L / mesh), int(W / mesh)
E, nu = 30e9, 0.2
rho = 25000
```

A.2 Mindlin plate formulation

The finite element formulation is based on the Mindlin plate theory, including membrane, bending, and shear stiffness contributions.

```
G = E / (2 * (1 + nu))

Dm_mat = (E * h / (1 - nu ** 2)) * np.array([[1, nu, 0], [nu, 1, 0], [0, 0, (1 - nu) / 2]])
D_const = (E * h ** 3) / (12 * (1 - nu ** 2))
Df_mat = D_const * np.array([[1, nu, 0], [nu, 1, 0], [0, 0, (1 - nu) / 2]])
Ds_mat = (5 / 6) * (G * h) * np.eye(2)
def shape_functions(xi, eta):
    N = 0.25 * np.array([(1 - xi) * (1 - eta), (1 + xi) * (1 - eta), (1 + xi) * (1 + eta), (1 - xi) * (1
+ eta)])
    dNxi = 0.25 * np.array([-(1 - eta), (1 - eta), (1 + eta), -(1 + eta)])
    dNeta = 0.25 * np.array([-(1 - xi), -(1 + xi), (1 + xi), (1 - xi)])
    return N, dNxi, dNeta
```

A.3 Finite element solver

The following function performs the finite element analysis of the slab system including stiffness matrix assembly, loading application, and stress recovery.

A.3.1 Mesh and model definition

```
def solve_full_analysis(hole_center, hole_size=1.0):
    nodes = {i + j * (nx + 1): (i * mesh, j * mesh) for j in range(ny + 1) for i in range(nx + 1)}
    Ndof = (nx + 1) * (ny + 1) * 5
    F = np.zeros(Ndof)
```

A.3.2 Opening geometry definition

```
# Square Hole: 1.0m x 1.0m
side = 1.0
x_start, x_end = hx - side / 2, hx + side / 2
y_start, y_end = hy - side / 2, hy + side / 2
```

A.3.3 Element stiffness assembly (core equation)

```
Ke += (Bm.T @ Dm_mat @ Bm + Bf.T @ Df_mat @ Bf + Bs.T @ Ds_mat @ Bs) * detJ *
mult
```

A.3.4 Lateral load application

```
F[5 * n] += (f_seismic_x / (L * W)) * (mesh ** 2 / 4)
F[5 * n + 1] += (f_seismic_y / (L * W)) * (mesh ** 2 / 4)
F[5 * n + 2] += (q_v_total * mesh ** 2) / 4
```

A.3.5 Boundary conditions

```
fixed = [5 * n + k for n, (x, y) in nodes.items() if (x <= wall_thick or x >= L - wall_thick) for
k in range(5)]
```

A.3.6 System solution

```
U[free] = spla.spsolve(K_sp[free, :][:, free], F[free])
```

A.3.7 Stress computation

```
vm = np.sqrt(sig_x ** 2 - sig_x * sig_y + sig_y ** 2 + 3 * tau_xy ** 2)
```

A.3.8 Output

```
return U, Moments, Shears, Stresses, (max_Mxx, max_Myy, max_Qx, max_Qy)
```

A.4 Particle Swarm Optimization (PSO)

This section presents the Particle Swarm Optimization (PSO) algorithm used to determine the optimal and critical positions of the opening by minimizing the maximum displacement of the slab. The algorithm iteratively updates particle positions based on individual and global best solutions.

A.4.1 PSO initialization

```
def run_dual_pso_optimization():

    n_particles, iterations = 20, 25
    hole_size = 1.0
    margin = hole_size / 2
    safety_gap = 0.2

    x_min = wall_thick + margin + safety_gap
    x_max = L - (wall_thick + margin + safety_gap)

    y_min = margin + safety_gap
    y_max = W - (margin + safety_gap)

    w_max, w_min = 0.9, 0.4
    c1_start, c1_end = 2.5, 0.5
    c2_start, c2_end = 0.5, 2.5


    particles_pos = np.random.uniform([x_min, y_min], [x_max, y_max], (n_particles, 2))
    particles_vel = np.zeros((n_particles, 2))
    pbest_pos = particles_pos.copy()
    pbest_val = np.full(n_particles, np.inf)
```

A.4.2 Global best and worst definition

```
gbest_pos_min = particles_pos[0].copy()
gbest_val_min = np.inf
gbest_pos_max = particles_pos[0].copy()
gbest_val_max = -np.inf
```

A.4.3 Objective function evaluation

The objective function evaluates the maximum displacement of the slab for each particle position.

```
for i in range(n_particles):

    U, _, _, _, _ = solve_full_analysis(particles_pos[i])
```

```
    u_vals = [np.max(np.abs(U[k::5])) * 1000 for k in range(3)]
    score = max(u_vals)
```

A.4.4 Velocity and position update rule

Particle positions are updated using the standard PSO velocity update equation.

```
particles_vel[i] = (w * particles_vel[i] +
                    c1 * r1 * (pbest_pos[i] - particles_pos[i]) +
                    c2 * r2 * (target_gbest - particles_pos[i]))

particles_pos[i] = np.clip(particles_pos[i] + particles_vel[i], [x_min, y_min], [x_max,
y_max])
```

A.4.5 Personal and global best and worst update

```
    if score < pbest_val[i]:
        pbest_val[i], pbest_pos[i] = score, particles_pos[i].copy()



idx_min, idx_max = np.argmin(scores), np.argmax(scores)

if scores[idx_min] < gbest_val_min:
    gbest_val_min, gbest_pos_min = scores[idx_min], particles_pos[idx_min].copy()
    best_details = details[idx_min]

if scores[idx_max] > gbest_val_max:
    gbest_val_max, gbest_pos_max = scores[idx_max], particles_pos[idx_max].copy()
    worst_details = details[idx_max]
```

A.4.6 Optimization output

The algorithm returns the optimal and critical opening positions along with the displacement history.

```
return gbest_pos_min, gbest_pos_max, h_best, h_worst, particles_pos_history, best_details,
worst_details
```

A.5 Hybrid local refinement (Nelder–Mead)

This section presents the hybrid refinement stage used to improve the accuracy of the optimal solution obtained from the Particle Swarm Optimization (PSO). A local search based on the Nelder–Mead method is applied to further minimize the objective function.

APPENDIX

A.5.1 Definition of the Objective Function

The refinement process uses the same objective function defined as the maximum displacement of the slab.

```
def objective_function(pos):

    U, _, _, _, _ = solve_full_analysis(pos)

    u1 = np.max(np.abs(U[0::5])) * 1000
    u2 = np.max(np.abs(U[1::5])) * 1000
    u3 = np.max(np.abs(U[2::5])) * 1000

    return max(u1, u2, u3)
```

A.5.2 Nelder–Mead Optimization Procedure

The Nelder–Mead algorithm is used as a local optimization method starting from the best solution obtained by PSO.

```
def run_hybrid_refinement(initial_best):

res = minimize(objective_function, initial_best, method='Nelder-Mead', tol=1e-6)
```

A.5.3 Updated Optimal Solution

The refined optimal position and corresponding displacement are extracted after convergence of the local search algorithm.

```
final_pos = res.x
```

```

final_score = res.fun
```

A.5.4 Improvement Assessment

The improvement achieved by the hybrid refinement is evaluated by comparing the PSO and refined solutions.

```
print(f" Initial Best (PSO): {initial_best}")
print(f" Refined Best (Local): {final_pos}")
print(f" Improvement: {initial_best - final_pos}")
```

A.5.5 Return Statement

The function returns the refined optimal position.

```
return final_pos
```

REFERENCES

[1] B. SADEK, "Analyse de l'influence de la déformabilité des diaphragmes horizontaux dans leur plan sur la distribution des charges sismiques aux éléments de contreventement," Université Abdelhamid Ibn Badis de Mostaganem, Mostaganem, 2019.

[2] A. M. A. H. M. E. Tarek F. El-Shafiey, "Effect of opening shape, size and location on the punching shear behaviour," *https://www.sciencedirect.com/,* p. 14, 2022.

[3] R. K. 1, N. P. 2 and a. R. Ghaffari, "In-Plane Flexibility of Reinforced Concrete Floor Diaphragms with Openings," *Structures Congress* , p. 8, 2018.

[4] F. K. ·. N. A. ·. R. Jankowski1, "Machine learning-based seismic response and performance," *Civil and Mechanical Engineering* , p. 21, 2023.

[5] V. S. T. K. Ozgur Anil, "Effect of opening size and," *Magazine of Concrete Research,* p. 13, 2014.

[6] M. A.Hassan, "Comparative study between using rigid diaphragm and," *Texas Journal of Engineering and Technology, https://zienjournals.com/,* p. 28/06/2022, 2022.

[7] V. T. a. V. A. Anton Chepurnenko, "Optimizing the Location of Supports under a Monolithic Floor Slab," *https://www.mdpi.com/journal/civileng,* p. 19, 2024.

[8] M. Lutz, Learning Python, Fourth Edition, Fourth Edition. ed., the United States of America: O'Reilly Media, 2009.

[9] M. Y. A. C. D. KENNEDY A. GOMEZ, *PYTHON MANUAL: A LEARNING GUIDE FOR STRUCTURAL ENGINEERING STUDENTS,* SENIOR PROJECT REPORT , 2023.

[10] E. Matthes., Python Crash Course., First printing ed., San Francisco: No Starch Press, 2016.

[11] P. N. M. e. P. A. E. BOUKILI, "Méthodes Numériques & Simulations avec Python," Faculté des Sciences et Technique-Errachidia, Université Moulay Ismail, Maroc, Errachidia, 2025.

[12] V. Lora, Python for Civil and Structural Engineers, GCAlgerie.com.

[13] W. McKinney, Python for Data Analysis, Third Edition ed., the United States of America.: O'Reilly Media, 2022.

[14] A. LAID, "Application des concepts de la biologie à l'optimisation des structures tridimensionnelles du Génie Civil.," Université Abdelhamid Ibn Badis de Mostaganem , Mostaganem , 2025.

[15] ”Conception optimale des structures métalliques par les méthodes ،M. .. H. Maram
،” Université Abdelhamid Ibn Badis de Mostaganem ،heuristiques modernes
.2025 ،Mostaganem

[16] R. Johansson, Numerical Python, Urayasu, Chiba, Japan : Library of Congress Control Number, 2015 .

[17] W. McKinne, Python for Data Analysis, the United States of America: O’Reilly Media, 2022.

[18] W. D. Nima Azizia, "A rotation-based geometrically nonlinear spectral Reissner–Mindlin," *Finite Elements in Analysis & Design*, p. 37, 2025.

[19] I. K. a. I. J. M. Herry Irpanni, "Development DKMQ Shell Element with Five," *International Journal of Mechanical Engineering and Robotics Research* , p. 5, 2017.

[20] M. C. S. a. J. C. Xiangdong Qian, "AHybrid Particle Swarm Optimization (PSO)-Simplex Algorithm for Damage Identification of Delaminated Beams," *Hindawi Publishing Corporation Mathematical Problems in Engineering*, p. 11, 2012.