

UTILIZATION OF OBJECT-ORIENTED PROGRAMMING, DESIGN PATTERNS AND JAVA FOR SIMULATING EARTHQUAKE-INDUCED POUNDINGS OF BASE-ISOLATED BUILDINGS

P. KOMODROMOS & P. POLYCARPOU

Department of Civil and Environmental Engineering, University of Cyprus, Cyprus.

ABSTRACT

Base-isolated buildings experience large horizontal relative displacements during strong earthquakes due to the excessive flexibility that is purposely incorporated, through seismic bearings, at their bases. When the available clearance around a base-isolated building is limited, there is a possibility of the building pounding against the surrounding moat wall or adjacent structures. Considering the nonlinearities involved in this structural impact problem, it is evident that the effects of potential pounding on the overall seismic response of base-isolated buildings during earthquake excitations should be investigated numerically through appropriate simulations. Object-oriented programming (OOP), design patterns (DPs), and the Java programming language have been utilized in order to design and implement a flexible and extendable software application that can be effectively used to perform the necessary numerical simulations and parametric studies of base-isolated buildings that may experience structural poundings during earthquake excitations. The aim of this paper is twofold: (i) to explain the significant advantages of utilizing OOP, DPs, and Java in structural analysis software and (ii) to use the developed software to study earthquake-induced poundings of base-isolated buildings.

Keywords: Base isolation, design patterns, impact, Java, object-oriented programming, pounding.

1 INTRODUCTION

Base isolation is usually implemented with the incorporation of flexible elastomeric bearings at the base of a relatively stiff building, to shift its fundamental eigenfrequency outside the danger of resonance range of frequencies and avoid resonance with an earthquake excitation. However, strong seismic excitations may cause large horizontal relative displacements at the isolation level of a base-isolated structure due to the excessive flexibility that is provided through the seismic bearings. Thus, a wide clearance, known as ‘seismic gap’, must be provided around a base-isolated building to accommodate the expected large horizontal displacements during a strong earthquake. However, the width of the provided seismic gap cannot be infinite due to practical constraints, especially in cases of seismically upgrading existing buildings in densely built areas. Furthermore, there are many uncertainties related to the characteristics of the expected earthquakes and the ways of estimating the induced relative displacements of a building. Hence, there is a possibility of a base-isolated building pounding against either the surrounding moat wall or an adjacent building during a stronger than expected earthquake excitation.

Pounding incidences between fixed-supported (FS) buildings during earthquakes had motivated relevant research [1–5]. Yet, very limited research work has been carried out for poundings of multi-storey base-isolated buildings [6–11], which exhibit very different dynamic characteristics from FS buildings. In particular, poundings of a base-isolated building occur primarily due to large relative displacements at the isolation level (Fig. 1), while poundings of a conventionally FS building occur due to the deformations of the superstructure, typically at the building tops. In addition, the performance requirements for buildings

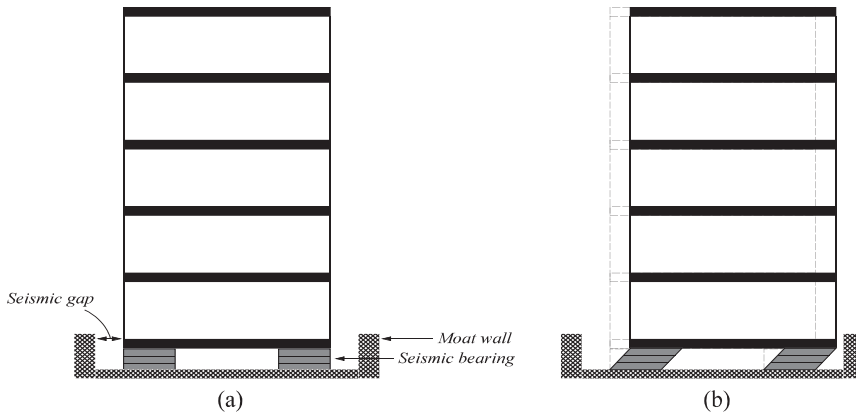


Figure 1: (a) Configuration of a base-isolated building; (b) deformation during an earthquake.

that utilize an innovative earthquake-resistant design, such as base isolation, are often much more stringent than those for conventionally FS buildings.

This paper presents a simple methodology for the numerical simulation of base-isolated buildings that undergo earthquake excitations with the possibility of impact occurrences. That methodology, which is described in the following section, is implemented utilizing advances in software engineering. In particular, object-oriented programming and design patterns are used in the design of the software to achieve the required extensibility, while the Java programming language is used for the implementation of the software. A practical example is also given to demonstrate the usage of the developed software, accompanied by some selected findings, while performing relevant numerical simulations and parametric studies.

2 METHODOLOGY FOR SIMULATING STRUCTURAL POUNDING

The modelling of the simulated base-isolated structures is performed in two dimensions (2D), while the multi-storey buildings are modeled as multi-degree of freedom (MDOF) systems, with shear-beam behavior, and the masses lumped at the floor levels, assuming linear elastic behavior during earthquake excitations. A bilinear model is used for the simulation of the behavior of the seismic isolation system, which corresponds to the usage of Lead Rubber Bearings (LRB). Specifically, prior to the yielding of the lead core, the isolation system has an initial stiffness K_1 , which is an order of magnitude higher than the post-yield stiffness K_2 that corresponds solely to the stiffness of the rubber.

2.1 Impact modelling

The numerical modelling of structural impact and the assessment of the impact forces acting on the colliding bodies is an interesting research topic not only for cases of structural poundings but also for other engineering problems involving numerical simulation of impact. Impact incidences during the dynamic response of structural systems, such as multi-storey buildings under earthquake excitations, are typically considered using force-based methods, also known as ‘penalty’ methods. These methods allow small interpenetrations between the colliding structures, which are justified by their deformability at the vicinity of the impacts. Contact springs are automatically formed as soon as an impact is detected, kept active as long as the colliding bodies remain in contact and removed as soon as the bodies are detached

from each other. The interpenetration depth is used together with the stiffness of the contact spring to estimate, according to the impact model that is adopted, the contact forces that are applied to the structures, pushing them apart.

In this research work, it is important to be able to use any of the various impact models, which, based on parallel impact springs and dashpots, exert impact forces to the colliding structures whenever their separation distances are exceeded. The impact models can be either linear or nonlinear, depending on whether the impact force is increasing linearly or exponentially with the indentation. Appropriate design and implementation of the software achieves the desired flexibility to be able to select at run-time the most suitable impact model to be used.

One such impact model is the modified linear viscoelastic impact model [9], which is a slight variation of the classical Kelvin–Voigt impact model. According to the modified linear viscoelastic impact model, which is used in the presented simulation results, the impact force at each time-step is computed by the following expressions:

$$F_{imp}(t + \Delta t) = \begin{cases} k_{imp} \cdot \delta(t) + c_{imp} \cdot \dot{\delta}(t) & \text{when } F_{imp}(t) > 0 \\ 0 & \text{when } F_{imp}(t) \leq 0 \end{cases} \quad (1)$$

where, $\delta(t)$, is the interpenetration depth, $\dot{\delta}(t)$ is the relative velocity between the colliding bodies, k_{imp} is the impact spring's stiffness, and c_{imp} is the impact damping coefficient. With this impact model, the tensile forces that arise at the end of the restitution period can be omitted, allowing small plastic deformations to be introduced, which increase the available clearance around the base-isolated building (Fig. 2). The impact damping coefficient can be computed by the following formulas, provided by Anagnostopoulos [1]:

$$c_{imp} = 2 \cdot \zeta_{imp} \sqrt{k_{imp} \frac{m_1 \cdot m_2}{m_1 + m_2}}, \quad \zeta_{imp} = -\frac{\ln(COR)}{\sqrt{\pi^2 + (\ln(COR))^2}} \quad (2)$$

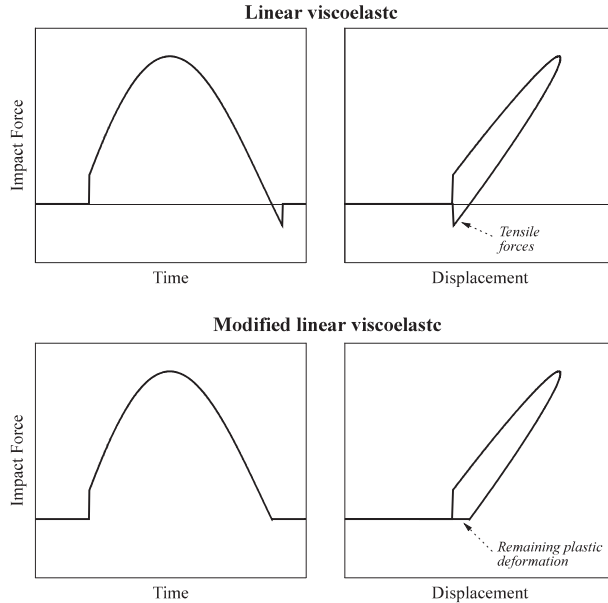


Figure 2: The classical and the modified linear viscoelastic (Kelvin–Voigt) impact model.

In the above formulas, m_1 and m_2 are the masses of the two colliding bodies and COR represents the coefficient of restitution, which is defined as the ratio of relative velocities after and before impact ($0 < COR \leq 1$). For the presented simulations, the value of 2500 kN/mm is chosen for the impact stiffness, while the COR is assumed to be equal to 0.6.

The effect of using different types of force-based impact models that are available in the scientific literature on the response of seismically isolated (SI) buildings during poundings has been investigated in a relevant study [12]. The results of that research work have revealed that, in general, the type of the impact model does not significantly affect the overall response of the colliding structures, assuming that the corresponding impact parameters are appropriately selected and used.

2.2 Equations of motion

The differential equations of motion for a base-isolated building, which is modeled as a MDOF system, are expressed in the following matrix form:

$$\underline{F}^I(t) + \underline{F}^D(t) + \underline{F}^E(t) = \underline{0} \quad (3)$$

where, \underline{F}^I , \underline{F}^D , and \underline{F}^E are the vectors of the inertia, damping, and elastic forces, respectively, acting on the structure at time t . For a seismic excitation with an acceleration time-history, $\ddot{U}_g(t)$, the inertia forces are expressed as follows:

$$\underline{F}^I(t) = \underline{M} \cdot \ddot{\underline{U}}(t) + \underline{M} \cdot \underline{1} \cdot \ddot{U}_g(t) \quad (4)$$

where $\underline{1} = [1 \ 1 \ \dots \ 1]^T$. The damping forces are expressed in terms of the floor relative velocities and the damping matrix of the MDOF system, through the following expression:

$$\underline{F}^D(t) = \underline{C} \cdot \dot{\underline{U}}(t). \quad (5)$$

The elastic forces (\underline{f}_s) of the superstructure are computed based on the stiffness matrix and the corresponding relative displacements at time t , while the shear forces for the base isolation system are computed according to the model that is used for the seismic isolation system. In the presented simulations, the bilinear model is used to compute the shear forces based on the relative displacement time-history $u(t)$ and the relative velocity sign at the base isolation level at time t :

$$\underline{F}^E(t) = \begin{cases} \underline{f}_s(u(t), \dot{u}(t)) & \text{no impact} \\ \underline{f}_s(u(t), \dot{u}(t)) + \underline{e} \cdot F_{imp}(t) & \text{during impact.} \end{cases} \quad (6)$$

The impact force F_{imp} during impact is computed at each time-step according to the corresponding impact model. Each of the terms e_i of vector \underline{e} , which has a dimension equal to the number of the degrees of freedom, is equal to 0 when no contact is detected in DOF, and 1 when an impact occurs at the corresponding floor level. The equations of motion are integrated using a numerical method, in this case the Central Difference Method (CDM), computing the displacements at time $t + \Delta t$.

2.3 Integration time-step

While simulating pounding use a ‘penalty method’, the selection of a small time-step is very important for both the accuracy of the computed impact force and the proper detection of all

potential impacts. To ensure the stability and the accuracy of the CDM, a sufficiently small time-step must be used for the time-integration, according to the following formula:

$$\Delta t < \Delta t_{cr} \frac{2}{\omega_{max}} = \frac{T_{min}}{\pi} \quad (7)$$

where ω_{max} is the maximum eigenfrequency of the system. However, during impacts, ω_{max} should be the equivalent maximum eigenfrequency of the force-based impact model:

$$\omega_{imp} = \sqrt{\frac{k_{imp}}{m_{eff}}} \quad (8)$$

where $m_{eff} = \frac{m_1 \cdot m_2}{m_1 + m_2}$.

For example, for an impact stiffness $k_{imp} = 2500$ kN/mm and two colliding masses of 250 tons each, the equivalent eigenfrequency of the impact model is equal to 141.42 rad/s. Therefore, according to the stability criterion of the CDM (eqn. 7), the critical time-step is about 1.4×10^{-2} s. In the current research study, the time-step is taken to be 2×10^{-5} s, which is three orders of magnitude smaller than the aforementioned critical time-step to ensure sufficiently accurate calculation of the contact and separation time instances.

3 UTILIZATION OF MODERN SOFTWARE DESIGN

To numerically investigate this research problem, it is very important to properly design and implement a software application that can be effectively and efficiently perform large number of parametric analyses of base-isolated buildings that may collide with adjacent structures. The developed software should not only facilitate specific needs of this research work but also maintain the extensibility that is required for considering different impact and structural models and approaches. As explained in the following section, modern object-oriented programming approaches have been employed to benefit from the significant advantages that they can offer, compared to conventional procedure-oriented programming approaches.

In particular, the numerical simulations that should be performed to address this research problem require the ability to simultaneously simulate multiple structures, using different impact and structural models, while certain characteristics should be varied, to parametrically study the problem. Moreover, the aim of this research effort was to initially study the problem in 2D and then extend it to three dimensions (3D) to study torsional and other spatial effects. Therefore, the extensibility of the software from 2D to 3D is a very crucial characteristic and has been maintained by utilizing advances in modern software design.

3.1 Disadvantages of procedure-oriented programming

Conventional structural analysis software often lacks the desired flexibility for modifications and extensions as it is habitually developed using Procedure-Oriented Programming (POP), which is inherently based on tightly coupled procedures. POP is based on a number of elementary procedures, which are invoked to perform certain tasks on data that are not associated with them and without any prevention mechanisms against accidental data corruption. In spite of the remarkable increase in computing power and resources, the design of structural analysis software, even for research purposes, is often still based on POP, resulting in a very sizable and complicated source code, in the effort to support different simulation models and analysis approaches.

Research needs call for more functionalities and different characteristics of modern structural analysis software that cannot be achieved with classical software design approaches and POP languages. While frequent changes and extensions are routinely requested in structural analysis, especially if they are used for research purposes, POP software is inherently not easily reusable, modifiable, or extendable.

In spite of the significant advances in software engineering and computing hardware, structural analysis software still does not provide the desirable levels of maintainability and extensibility to facilitate the state-of-the-art knowledge and the constantly changing needs of the researchers. The increased demands and requirements of modern structural analysis software well exceed the adequacy of POP and justify the utilization of recent advances in software design and modern development methodologies.

3.2 Object-oriented programming

Complicated structural systems, such as those involving structural impact incidences, can be more effectively simulated with software that is designed according to the Object-Oriented programming (OOP) paradigm and utilizes the provided abstraction, encapsulation, information hiding, inheritance, and polymorphism mechanisms.

OOP has been introduced in structural engineering software in the 1990s, much after OOP languages had become available, and the OOP principles have been promoted by relevant research work [13–19]. However, the advantages of OOP have been appreciated only by a very small number of researchers. In addition, the code of many Object-Oriented (OO) structural engineering applications is still tightly coupled without fully utilizing the advances in OOP and software engineering to develop flexible and extendable applications.

Although the term OOP is used to collectively represent Object-Oriented Analysis (OOA), Object-Oriented Design (OOD), and Object-Oriented Programming (OOP), in structural engineering software, emphasis is given primarily to the OOP aspect and much less to the OOA and OOD. In particular, OOA is used to identify the objectives of a software system, while OOD the software is designed to fulfill these objectives. To fully utilize the advantages of OOP in structural analysis software, it is not sufficient to use an OOP language, such as C++, Java, and C#, but certain OOA and OOD principles should also be followed, from the initial software analysis and design phases, to achieve the desired flexibility.

Furthermore, the next logical step to code-reusability is the achievement of OOD reusability, which can offer the experience that is obtained through optimized solutions of recurring design problems of structural engineering software. That is exactly what is proposed through this paper, with the utilization of Design Patterns (DPs), to reuse optimized solutions of recurring problems in designing structural analysis software.

3.3 Design patterns

The DPs can be used to design and develop structural analysis software in ways that manage its size and complexity. Although the DPs had originated from architecture, thanks to the relevant work of Christopher Alexander [20], they have been utilized in the design of large OO software applications only after a set of 23 specific DPs was defined in the *Design Patterns: Elements of Reusable Object-Oriented Software* book [21]. However, DPs have not yet been exploited in the design and development of structural analysis software. The DPs take advantage of the experience gained by software developers to recurring problems [21, 22].

The aim is to design and develop flexible software that can be easily reused, modified, and extended to fulfill the continually changing needs of academic research or engineering practice. In particular, the DPs provide proven effective designs to organize classes and objects, regarding their responsibilities and relations, to achieve the desired maintainability and extensibility.

OOP and DPs can be used to substantially reduce the development time, considering future revisions and extensions, especially as the size and complexity of the code increases. More importantly, future modifications and extensions can be performed with the minimum possible impact on existing code, avoiding unintentional introduction of bugs. In addition, the DPs provide a concise vocabulary that facilitates the communication and collaboration among software designers, developers, and users of structural analysis software not only during development time but also during expected revision and extension phases.

Most DPs enable the isolation, through encapsulation, of parts of software applications from the rest of the code, by keeping the classes separated and uninformed about the others, so as to avoid the possibility of accidentally introducing bugs during modifications. Although OOP provides both composition and inheritance, it is preferable, whenever possible, to use composition over inheritance to enhance the flexibility of the software. This can be achieved by encapsulating the behavior in an object that can be easily changed with some other behavior that implements the proper behavior interface, essentially and dynamically ‘inheriting’ behavior through composition and delegation at run-time.

The DPs can be categorized [21] in three major categories: (i) *the creational DPs*, which provide different flexible ways to instantiate objects dynamically at run-time without providing any creation details, (ii) *the structural DPs*, which organize classes and objects into larger structures in ways that facilitate future modifications and extensions, and (iii) *the behavioral DPs*, which define the communications and interactions of objects and classes so as to facilitate encapsulation of the relevant processes and distribute responsibilities.

DPs have been utilized for the design of the developed software application to achieve the desired maintainability and extensibility of the code, which can be easily extended in the future to enable 3D dynamic simulations of multi-story buildings that may experience pounding with adjacent structures whenever the available clearance around them is exceeded.

For instance, Fig. 3 demonstrates the utilization of the abstract factory DP to instantiate a set of related objects, such as a structure and an excitation, without an explicit binding of the code to concrete subclasses, to facilitate future modifications and extensions of the concrete classes. In particular, the abstract factory is a Creational DP, as well as the Factory Method, the Builder, the Prototype and the Singleton patterns. The Creational DPs provide adaptable ways to create instances of classes so that the system that uses them is independent of how the objects are instantiated, composed, and represented. In essence, the Creational DPs encapsulate information regarding which concrete classes are instantiated and how they are configured, hiding this information from the system that uses them, to provide the loose-coupling that is essential for future extensions of the application. The abstract factory DP offers a flexible way to instantiate objects without specifying their concrete classes to avoid explicit binding of the code to concrete subclasses, which would limit the extensibility of the code.

Specifically, the abstract factory DP delegates, through composition, the instantiation to another object that creates and returns one of several families of related objects, without letting the client know about the actual objects that are created. The abstract factory superclass defines the interface that all concrete factories must implement to create the families of related objects. Each of the methods of the specified interface is responsible to create the corresponding concrete object. Therefore, the abstract factory DP can be used to instantiate one of the number of

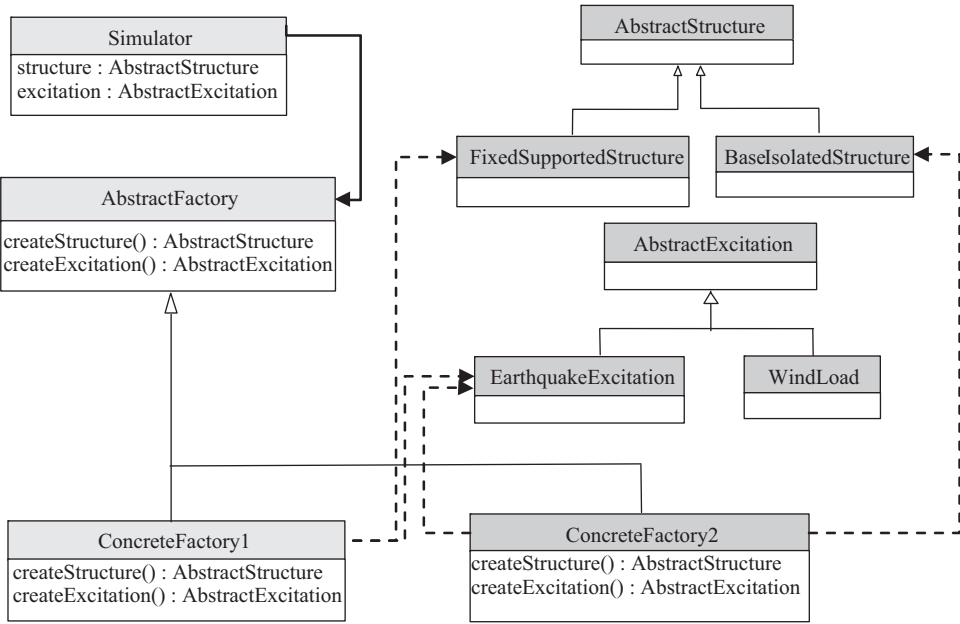


Figure 3: Utilization of the abstract factory DP to instantiate a set of related objects, such as structure and excitation, without an explicit binding of the code to concrete subclasses, to facilitate future modifications and extensions of the concrete classes.

concrete implementations of an abstract factory class, returning one of the several factories. Subsequently, each concrete factory returns a set of instances of several related classes by implementing the methods that are defined in the abstract factory for creating a family of objects. Consequently, the actual concrete classes that are instantiated and used in an application are isolated from the client, which facilitates straightforward modifications and transparent additions of new families of concrete classes. In addition, this DP allows the implementation and use of a variety of factories that can instantiate different objects, depending on the current and future needs of the engineering application, to obtain various behaviors.

3.4 Utilization of Java technologies

Although the C++ programming language has been used in most attempts to utilize OOP in developing structural analysis software, Java [23] offers several advantages compared to C++, as it combines the most important advantages of OOP languages, while providing portability and architectural neutrality among the various computing hardware and operating systems [24]. Java is a pure OOP language that combines the best ideas, concepts, and mechanisms of other OOP languages, such as virtual functions and polymorphism, inheritance, function overloading, garbage collection, and exception handling. It offers modularity, robustness, reusability, portability, architectural neutrality, superior memory management, and multi-threading capabilities.

In addition, the various Java APIs allow efficient development of applications with graphics, database connectivity, and graphical user interfaces without a need to use any other programming language or library. Furthermore, the availability of multiple processors

with shared memory can be easily utilized with the multithreading capabilities of Java to perform faster numerical simulations and parametric analyses, such as those needed in this research problem. Although multithreading is not really parallel computing, utilization of multiple processors can be achieved with minimal effort, without increasing the complexity of the program and avoiding the resulting overhead to actually parallelize it, which may not be feasible in some structural engineering problems.

The use of the Java programming language in engineering applications sometimes encounters concerns about the performance limitations and scalability of Java. The expected performance difference between Java and C/C++ is due to the fact that while C/C++ programs are compiled into executable machine instructions for the underlying architecture and operating system, Java code is compiled into bytecode, which is then translated into machine instructions during execution by the Java Virtual Machine (JVM). Although this two-phase compilation–interpretation has some overhead on the computational performance, it provides portability to the Java programs. Furthermore, the latest JVM allows just-in-time compilation that achieves performance comparable to that of a compiled language due to run-time optimization using profiling and recompilation during execution. In contrast, using C/C++, any optimization is done during compilation. The run-time optimizations are more powerful since they can take advantage of given conditions during execution of the program, while static compilation can take advantage of only what the compiler can predict based on certain predefined rules.

3.5 Developed software

A primary aim of this research work was the development of specialized software to efficiently and effectively perform numerical simulations and parametric analyses of base-isolated buildings with automatic impact detection and handling capabilities. OOP, DPs, and Java have been utilized to design and implement a flexible, robust, and extendable software application with effective visualization capabilities (Fig. 4) that can be used to efficiently and effectively perform the necessary numerical simulations and parametric analyses to

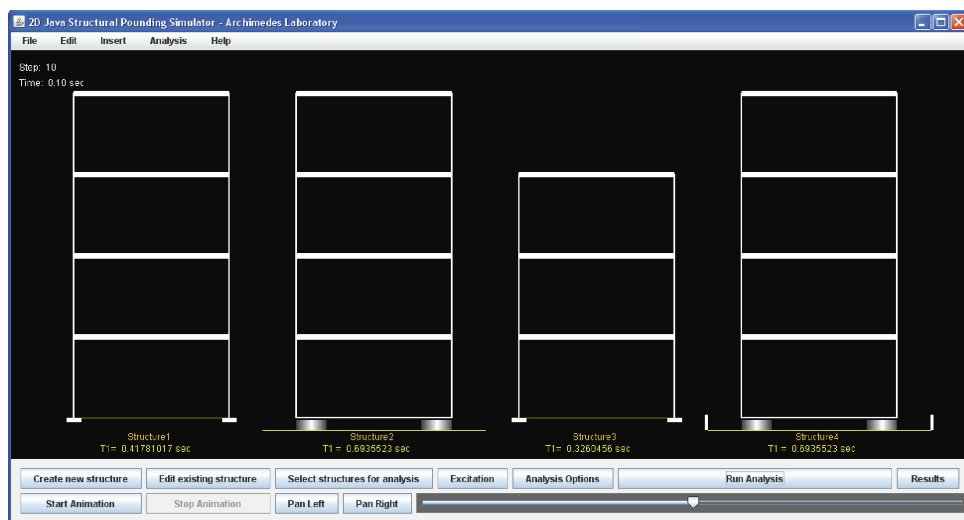


Figure 4: The main window in the GUI of the developed software.

thoroughly study this research problem. The software has been initially developed for performing 2D simulations of MDOF systems with shear-beam behavior under dynamic loadings, while its extension to 3D simulations is currently underway.

The developed software application allows the consideration of poundings of a base-isolated building, either with the surrounding moat wall or with one or more adjacent buildings. In the latter case, the equations of motion for two or more buildings, undergoing the same ground excitation, are coupled and integrated simultaneously. Moreover, the software allows both linear and bilinear models to be used for the simulation of the base isolation system. The ability to automatically perform parametric analyses is also provided, to investigate the effects of certain parameters, such as the structural characteristics, the size of the separation gap, and the earthquake characteristics, on the computed response.

A typical nonlinear dynamic analysis of a five-story base-isolated building, considering bilinear behavior for the seismic isolation system, under an earthquake excitation with a duration of 30 s, considering also poundings with the adjacent moat wall, needs a fraction of a second to be performed with a common personal computer (PC). The efficiency of the developed software can conduct of large numbers of dynamic analyses and parametric analyses. During the parametric analysis mode of the software, certain parameters are automatically varied through a user-specified range and step to parametrically study the problem under consideration.

Finally, a robust Graphical User Interface (GUI) with various capabilities has been designed and implemented, using the Java Swing API, to facilitate the effective performance of simulations and parametric analyses. The developed software allows the input data to be either imported from input files or specified using the GUI, while the computed results can be exported in output files or used to generate and store plots and animations in vector formats.

4 EXAMPLE

A typical base-isolated building is used in the simulations to examine the pounding effects on its seismic response. The superstructure is assumed to have five floors, each with a lumped mass of 320 t, except of the top floor where a mass of 250 t is considered. Each story has a horizontal stiffness of 600 MN/m. An additional mass of 320 t is assumed to be lumped at the isolation level, while the bilinear properties of the isolation system are taken as follows: $K_1 = 200$ MN/m, $K_2 = 25$ MN/m, $f_y = 0.1 \times W_{tot}$, where W_{tot} is the total weight of the building. A viscous damping ratio of 2% is assumed for the superstructure, while for the isolation system, in addition to the hysteretic energy dissipation, a 5% viscous damping ratio is considered. The fundamental period of the FS superstructure is equal to $T_{fixed} = 0.49$ s.

The structure is considered under three different configurations (Fig. 5). In the first case, the base-isolated building is considered without the possibility of impacts, assuming a sufficiently wide seismic gap. In the second case, a seismic gap width equal to 24 cm, which is about 15% smaller than the maximum unobstructed induced relative displacement (27.48 cm) of the building under the San Fernando earthquake, is considered at its base, leading to pounding with the surrounding moat wall. The latter is assumed to be completely rigid and move with the ground during an earthquake. In the third case, a conventional FS building is assumed to be located on the left side of the base-isolated building, leading to potential pounding incidences between the two buildings during a very strong earthquake. The seismic gap in the third case is also considered to be 24 cm, as well as the clear distance between the two buildings. The FS building has the same characteristics (stiffness of stories and masses) as the superstructure of the base-isolated building, while a damping ratio of 5% is considered. The San Fernando, California, USA, earthquake of 1971, which is a very strong earthquake (1.17 g), is used as the ground

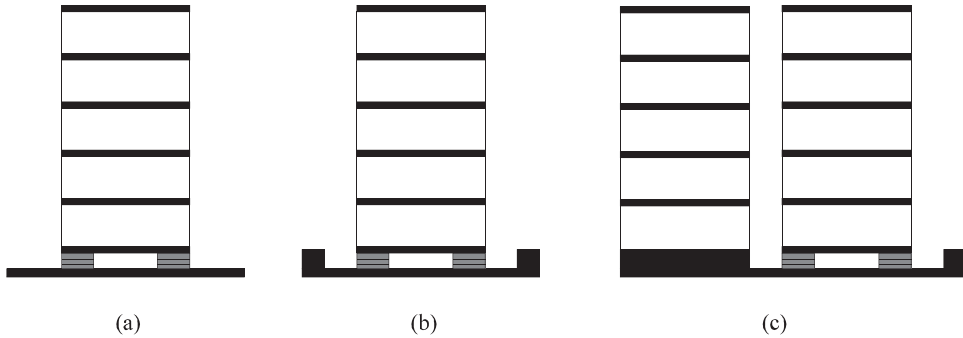


Figure 5: Three different configurations of the five-story base-isolated building taken into account in the example regarding the locations of potential impacts: (a) no impacts; (b) impacts only at the base; and (c) impacts at all floors.

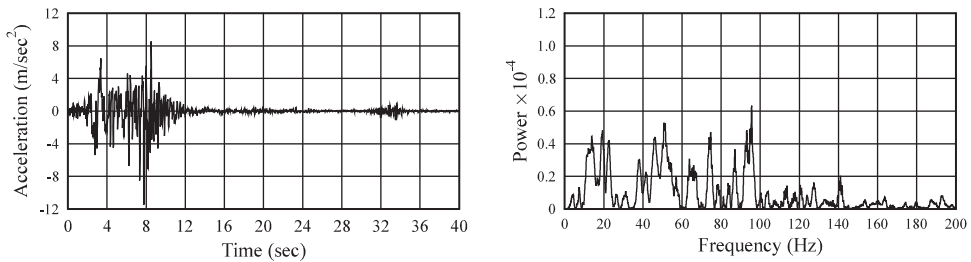


Figure 6: Ground acceleration time-history and its corresponding Fourier spectrum for the San Fernando earthquake record.

excitation in the performed simulations. Figure 6 shows the ground acceleration time-history and the frequency content of the aforementioned seismic record.

4.1 Pounding effect on the acceleration time-histories

The analyses revealed that in the second case, the base-isolated building hits only once against the left wall at time 3.66 s during the earthquake. In the third case, the first impact occurs at time 3.57 s, between the two buildings at the fifth floor, due to the deformations of the superstructures. The initial pounding is followed by another impact between the fourth floors at 3.58 s, then an impact between the third floors at 3.59 s, and, finally, an impact between the second floors at time 3.60 s. No impacts occurred either at the first floor or at the isolation level with the surrounding moat wall.

Figure 7 presents the acceleration time-histories of the SI building, at the base isolation and top floor levels, for the three different configuration cases. It is observed that floor accelerations are very sensitive to impact occurrences. In particular, spikes of high-amplitude occur at the times of impact and are more pronounced at the time-history response of the floor that experiences impact incidences. For example, in the case of poundings only with the moat wall at the isolation level, although the width of the clearance is only 3.5 cm smaller than the maximum unobstructed relative displacement at that level, very high accelerations are observed

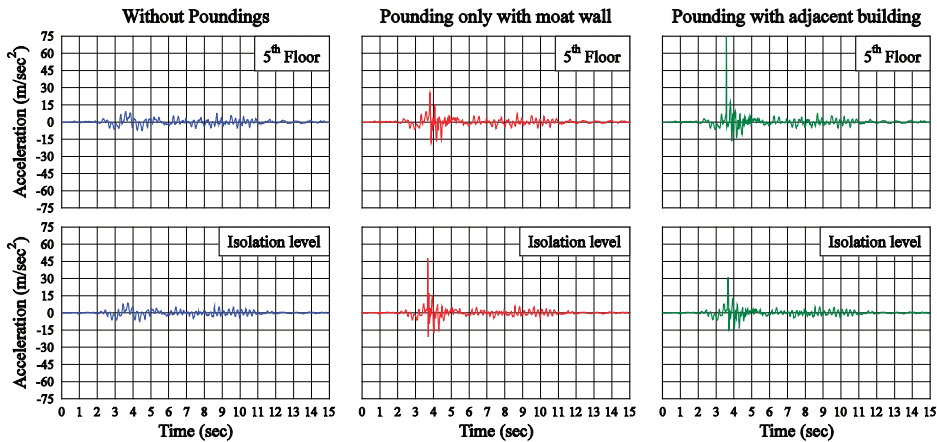


Figure 7: Acceleration time-histories at each floor level of the five-story base-isolated building, without and with poundings, respectively.

at the time of impact (3.66 s) that reach up to six times the corresponding peak floor accelerations without poundings, in which case the response, as shown by the plot, is much smoother. Nevertheless, due to the structure's damping, very shortly after the impact occurrences, the response tends to become identical to the corresponding response without poundings.

4.2 Pounding effect on the frequency content

To demonstrate the effect of pounding incidences on the response of the SI building in terms of frequency content, the corresponding Fourier spectra of the time-histories presented in Fig. 7, were created and shown in Fig. 8. Furthermore, Fig. 9 presents the outcome of the subtraction of the plots of the first column of Fig. 8 from the second and third column, respectively. Essentially, the plots in Fig. 9 provide the contribution of the poundings in response in terms of the frequency content.

To better understand the information that is provided in Figs 8 and 9, it is important to have the eigenmodes and eigenfrequencies of the SI building. However, to calculate the eigenvalues of the structure, it is necessary to consider an equivalent stiffness for the isolation system, due to its nonlinearity. Assuming that the base isolation system consists of lead-rubber bearings with bilinear behavior, the eigenvalues are calculated for two stages, the one before and the other after the yielding of the isolators' lead cores. Regarding the stiffness K_{iso} of the isolation system, the initial stiffness equals $K_1 = 200$ MN/m, while the post-yield stiffness equals $K_2 = 25$ MN/m. The eigenmodes and eigenvalues of the SI building for the two stages are presented in Fig. 10.

The plots in Fig. 8 demonstrate that the response of the SI building without any pounding contains mainly the first two eigenfrequencies which correspond mostly to the second stage of vibration, where the stiffness of the seismic isolation system equals the post-yield stiffness of the LRBs. During poundings, however, the high-frequency content increases, as shown in Fig. 8, while the amplitudes for lower frequencies remain almost the same as in the case of no poundings. This observation is more apparent in Fig. 9, where the contribution of pounding in the Fourier spectrum is presented. Specifically, the plots show that, during poundings the participation of higher modes of deformation is increased in relation to the case without poundings.

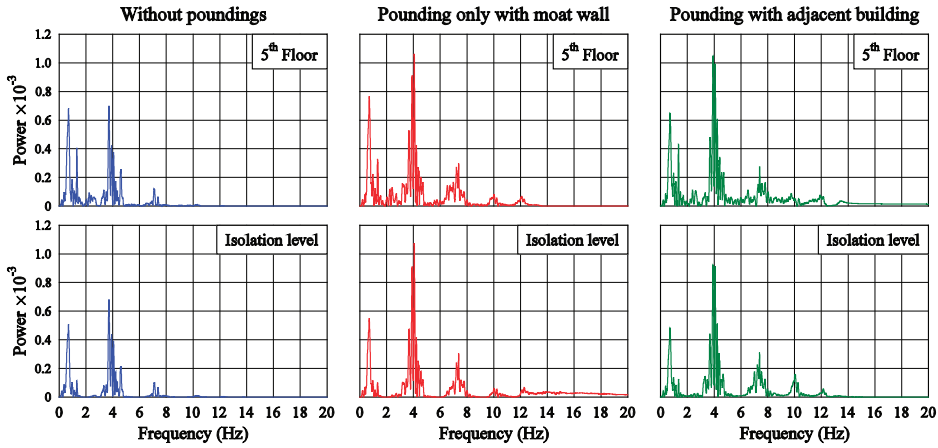


Figure 8: Fourier spectra of the response at the base and top of the SI building for the three cases under consideration.

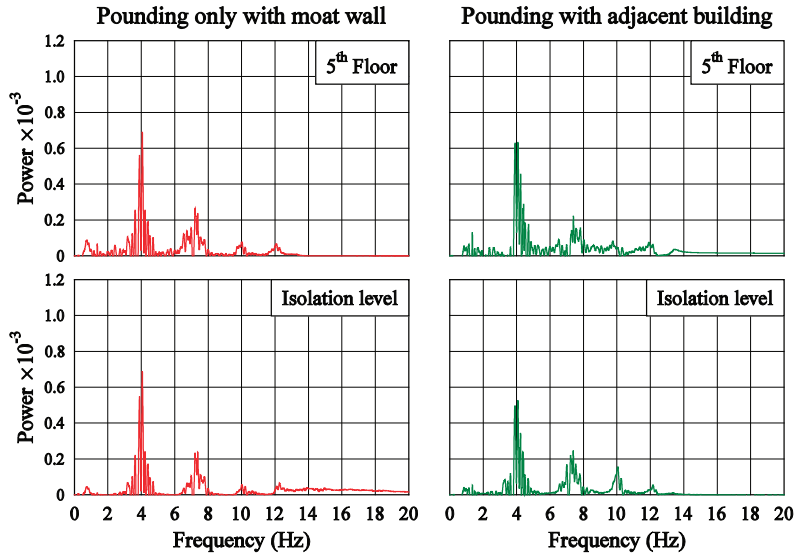


Figure 9: Contribution of pounding in terms of frequency content in the response of the SI building during impacts.

4.3 Pounding effect on maximum response values

The peak values of the inter-story deflections and absolute floor accelerations of the SI building for the three different configurations, as well as the corresponding maximum responses of the five-story FS building of the third case, are plotted together and compared in Fig. 11. It is observed that inter-story deflections are also significantly amplified (up to three times in this case) due to poundings, compared to the case without poundings. This amplification is more pronounced in the case of poundings of the base-isolated building

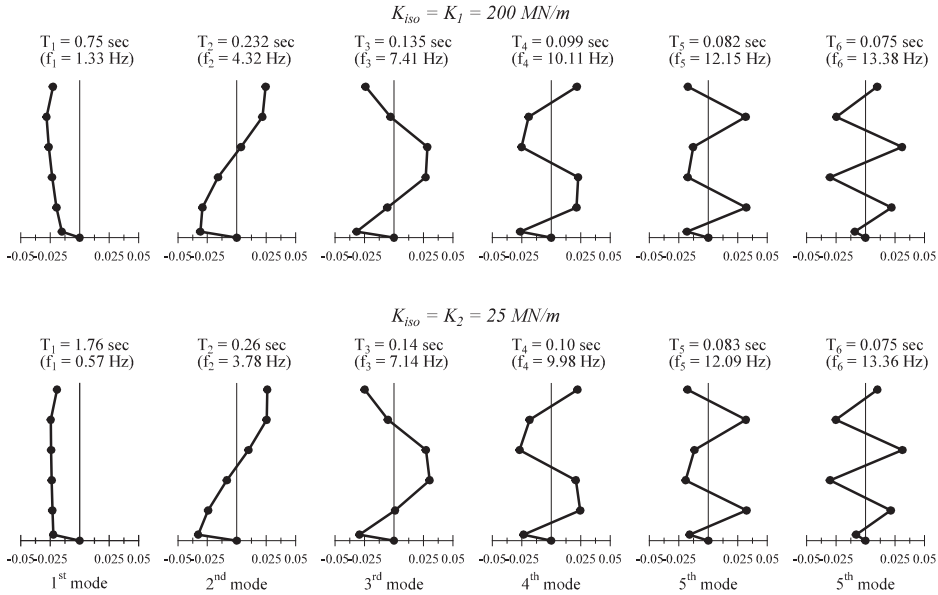


Figure 10: Eigenmodes of the five-story SI building, assuming an equivalent linear isolation system, for two different values of the seismic isolation stiffness, representing the initial stiffness K_I and the post-yield stiffness K_2 of the considered bilinear model, respectively.

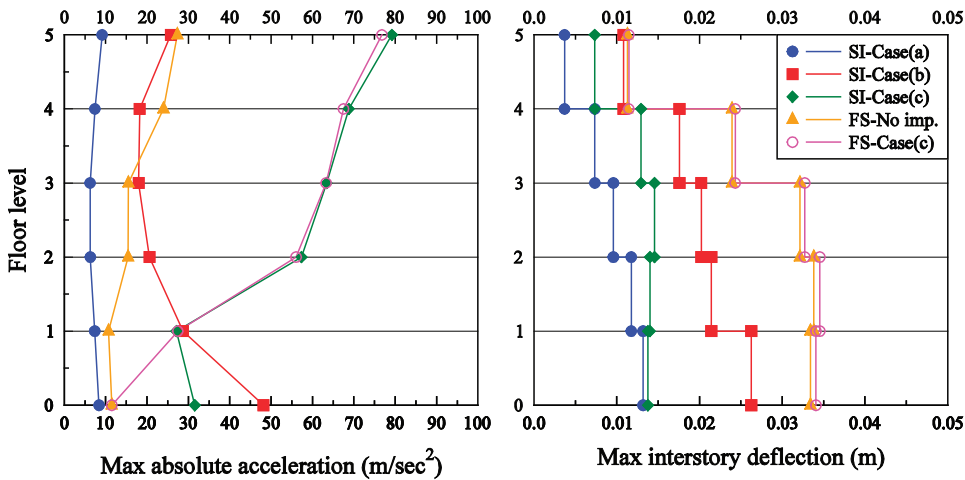


Figure 11: Peak floor accelerations and peak inter-story deflections of the SI building and the adjacent FS building, for the three different configurations of Fig. 5.

only with the moat wall, where the superstructure can deform unobstructed, in contrast to the case of having the five-story FS building on its left side, preventing the large horizontal deflections of the superstructure. Consequently, the same amplification of the shear forces that act on the particular stories of the superstructure are expected during such potential

impact incidences. If no sufficient strength is provided to the structural elements, taking into account these effects of potential poundings during the design of the superstructure of the base-isolated building, it can be severely damaged from potential poundings. It is also observed that the effect of pounding on the maximum inter-story deflections of the adjacent FS building is negligible.

Comparing the computed peak absolute floor accelerations of the building, the influence of poundings in the response is much more pronounced at the impacting floors in each case, that is, at the isolation level for the second configuration of the base-isolated building (Fig. 5b) and at the upper floors for the third configuration (Fig. 5c). It is well known from previous studies [1, 3] that the acceleration response is highly affected by impacts. It is also known that these extremely high values of floor accelerations that are caused by poundings can damage sensitive equipment that may be housed in the building. Another observation is that the maximum absolute floor accelerations of the SI building during poundings become larger than the corresponding responses of the FS building without poundings, while in the case where the two buildings pound against each other, they have almost the same maximum floor accelerations. Moreover, in the latter case, the peak floor acceleration value increases while moving to upper floors due to the increase in the impact velocity.

4.4 Effect of the seismic gap size

To examine the effect of the seismic gap size on the response of the base-isolated building during poundings with the moat wall (Fig. 5b), a parametric analysis has been conducted. In particular, the width of the seismic gap is varied from 10 to 45 cm with a step of 0.5 cm, considering equal gap sizes on both sides of the building. The plots in Fig. 12 present the peak floor accelerations and peak inter-story deflections of the five-story base-isolated building under the San Fernando earthquake record, in terms of the width of the seismic gap. In general, as the seismic gap increases, both floor accelerations and inter-story deflections of the superstructure decrease. However, the simulation results indicate that for relatively narrow gap sizes the response increases with the width of the available clearance, and after a certain value the response of the base-isolated building begins to decrease. This observation is more pronounced for the lower floors, which are closer to the impact location. This is probably due to the fact that for very narrow seismic gap widths (< 15 cm), in comparison with the

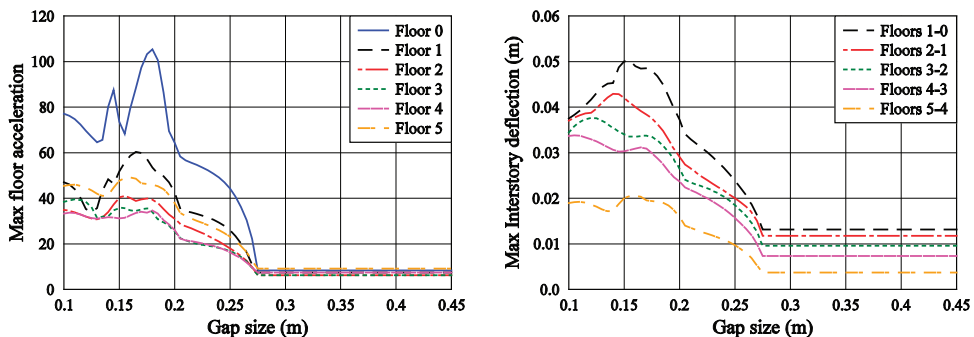


Figure 12: Peak floor responses of the five-story base-isolated building in terms of the width of the seismic gap, considering poundings only at its base (case b in Fig. 5).

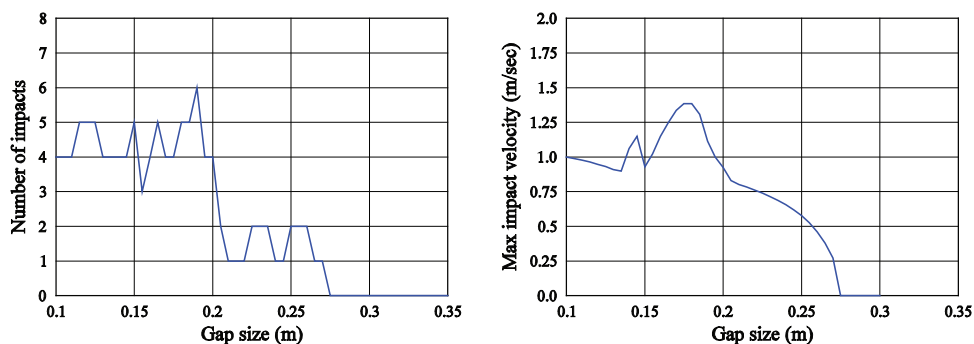


Figure 13: Total number of impacts and maximum impact velocity at the isolation level, in terms of the size of the seismic gap, considering poundings only at its base (case b in Fig. 5).

maximum unconstrained induced displacement, the base-isolated building retains relatively low impact velocities.

Figure 13 presents the total number of impacts and the maximum impact velocity at the isolation level of the five-storey base-isolated building, in terms of the seismic gap width. It is observed that the trend of the maximum impact velocity is very similar to the corresponding peak floor acceleration response of the base-isolated building at the isolation level, indicating that the amplification of the response due to impact is proportional to the impact velocity. It is also observed that the total number of impacts is not always decreasing with the width of the seismic gap as it was expected, and seems to depend from the earthquake characteristics and the structural properties. Nevertheless, for gap sizes larger than 17 cm, the peak floor responses and the impact velocity decreases with the width of the seismic gap.

5 CONCLUSIONS

A methodology that can be used for simulating earthquake-induced pounding of base-isolated buildings has been implemented in a software application that has been designed utilizing OOP and DPs and programmed using Java. In addition to the usage of OOP, this paper suggests the utilization of DPs to accomplish reusability of optimized solutions of recurring design problems in structural analysis software. The developed software has achieved the required extensibility, while ensuring the necessary efficiency and effectiveness in performing numerical simulations and parametric studies. A practical example with a typical five-story base-isolated building has been given to demonstrate the usage of the developed software, accompanied by some selected findings while performing relevant numerical simulations and parametric studies. The simulations show that poundings are particularly unfavorable for a base-isolated building and its contents, since they significantly increase the absolute floor accelerations and inter-story deflections. Specifically, very high accelerations are observed during structural impacts due to the sudden changes of the relative velocities, especially at the levels where impacts occur. Such high values of induced floor accelerations can damage sensitive equipment that may be housed in the building. Furthermore, significant amplifications of the story shear-forces have been observed especially at the upper floors, which may cause considerable damage to the corresponding structural and non-structural elements at that level.

REFERENCES

- [1] Anagnostopoulos, S.A., Pounding of buildings in series during earthquakes. *Earthquake Engineering and Structural Dynamics*, **16**, pp. 443–456, 1988. doi: <http://dx.doi.org/10.1002/eqe.4290160311>
- [2] Papadrakakis, M., Apostolopoulou, C., Zacharopoulos, A. & Bitzarakis, S., Three-dimensional simulation of structural pounding during earthquakes. *Journal of Engineering Mechanics*, **122**, pp. 423–431, 1996. doi: [http://dx.doi.org/10.1061/\(ASCE\)0733-9399\(1996\)122:5\(423\)](http://dx.doi.org/10.1061/(ASCE)0733-9399(1996)122:5(423))
- [3] Maison, B.F. & Kasai, K., Analysis for type of structural pounding. *Journal of Structural Engineering*, **116**, pp. 957–977, 1990. doi: [http://dx.doi.org/10.1061/\(ASCE\)0733-9445\(1990\)116:4\(957\)](http://dx.doi.org/10.1061/(ASCE)0733-9445(1990)116:4(957))
- [4] Anagnostopoulos, S.A. & Spiliopoulos, K.V., An investigation of earthquake induced pounding between adjacent buildings. *Earthquake Engineering and Structural Dynamics*, **21**, pp. 289–302, 1992. doi: <http://dx.doi.org/10.1002/eqe.4290210402>
- [5] Chau, K.T. & Wei, X.X., Poundings of structures modeled as non-linear impacts of two oscillators. *Earthquake Engineering and Structural Dynamics*, **30**, pp. 633–651, 2001. doi: <http://dx.doi.org/10.1002/eqe.27>
- [6] Tsai, H.C., Dynamic analysis of base-isolated shear beams bumping against stops. *Earthquake Engineering and Structural Dynamics*, **26**, pp. 515–528, 1997. doi: [http://dx.doi.org/10.1002/\(SICI\)1096-9845\(199705\)26:5<515::AID-EQE654>3.0.CO;2-C](http://dx.doi.org/10.1002/(SICI)1096-9845(199705)26:5<515::AID-EQE654>3.0.CO;2-C)
- [7] Malhotra, P.K., Dynamics of seismic impacts in base-isolated buildings. *Earthquake Engineering and Structural Dynamics*, **26**, pp. 797–813, 1997. doi: [http://dx.doi.org/10.1002/\(SICI\)1096-9845\(199708\)26:8<797::AID-EQE677>3.0.CO;2-6](http://dx.doi.org/10.1002/(SICI)1096-9845(199708)26:8<797::AID-EQE677>3.0.CO;2-6)
- [8] Matsagar, V.A. & Jangid, R.S., Seismic response of base-isolated structures during impact with adjacent structures. *Engineering Structures*, **25**, pp. 1311–1323, 2003. doi: [http://dx.doi.org/10.1016/S0141-0296\(03\)00081-6](http://dx.doi.org/10.1016/S0141-0296(03)00081-6)
- [9] Komodromos, P., Polycarpou, P.C., Papaloizou, L. & Phocas, M.C., Response of seismically isolated buildings considering poundings. *Earthquake Engineering and Structural Dynamics*, **36**, pp. 1605–1622, 2007. doi: <http://dx.doi.org/10.1002/eqe.692>
- [10] Polycarpou, P.C. & Komodromos, P. On poundings of a seismically isolated building with adjacent structures during strong earthquakes. *Earthquake Engineering and Structural Dynamics*, **39**, pp. 933–940, 2010.
- [11] Polycarpou, P.C. & Komodromos, P., Simulating seismically isolated buildings under earthquake-induced pounding incidences. *Proc. of the Eleventh International Conference on Structures Under Shock and Impact (SUSI2010)*, eds N. Jones, C.A. Brebbia & U. Mander, WIT Press: London, pp. 245–256, 2010.
- [12] Polycarpou, P.C. & Komodromos, P., On the numerical simulation of impacts for the investigation of earthquake-induced pounding of buildings, *The Tenth International Conference on Computational Structures Technology (CST2010)*, 14–7 Sep., Valencia, Spain, 2010.
- [13] Forde, B.W.R., Russell, A.D. & Stierner, S.F., Object-oriented knowledge frameworks. *Engineering with Computers*, **5(2)**, pp. 79–89, 1989. doi: <http://dx.doi.org/10.1007/BF01199071>
- [14] Mackie, R.I., Object-oriented programming of the finite element method. *International Journal of Numerical Methods in Engineering*, **35**, pp. 425–436, 1992. doi: <http://dx.doi.org/10.1002/nme.1620350212>
- [15] Zimmermann, T., Dubois-Pelerin, Y. & Bomme, P., Object-oriented finite element programming: I: Governing principles. *Computer Methods in Applied Mechanics*

- and Engineering*, **98(2)**, pp. 291–303, 1992. doi: [http://dx.doi.org/10.1016/0045-7825\(92\)90180-R](http://dx.doi.org/10.1016/0045-7825(92)90180-R)
- [16] Adeli, H. & Yu, G., An integrated computing environment for solution of complex engineering problems using the object-oriented programming paradigm and a black-board architecture. *Computers and Structures*, **54(2)**, pp. 255–265, 1995. doi: [http://dx.doi.org/10.1016/0045-7949\(94\)00323-U](http://dx.doi.org/10.1016/0045-7949(94)00323-U)
 - [17] Archer, G.C., Fenves, G. & Thewalt, C., A new object-oriented finite element analysis program architecture. *Computers and Structures*, **70(1)**, pp. 63–75, 1999. doi: [http://dx.doi.org/10.1016/S0045-7949\(98\)00194-1](http://dx.doi.org/10.1016/S0045-7949(98)00194-1)
 - [18] Takahashi, Y. & Fenves, G.L., Software framework for distributed experimental-computational simulation of structural systems. *Earthquake Engineering and Structural Dynamics*, **35(3)**, pp. 267–291, 2005. doi: <http://dx.doi.org/10.1002/eqe.518>
 - [19] Murthy, A.R.C., Palani, G.S. & Iyer, N.R., Object-oriented programming paradigm for damage tolerant evaluation of engineering structural components. *Advances in Engineering Software*, **42(1–2)**, pp. 12–24, 2011. doi: <http://dx.doi.org/10.1016/j.advengsoft.2010.10.012>
 - [20] Alexander, C., Ishikawa, S. & Silverstein, M., *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, 1977.
 - [21] Gamma, E., Helm, R., Johnson, R. & Vlissides J., *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley Pub Co, 1995.
 - [22] Cooper, J.W., *Java Design Patterns*. Addison Wesley Pub Co, 2000.
 - [23] Arnold, K., Gosling, J. & Holmes, D., *The Java Programming Language*, 3rd edn., Addison-Wesley Pub Co, 2000.
 - [24] Komodromos, P., A simplified updated Lagrangian approach for combining discrete and finite element methods. *Computational Mechanics*, **35(4)**, pp. 305–313, 2005. doi: <http://dx.doi.org/10.1007/s00466-004-0609-6>